A CIP-PSP funded pilot action
Grant agreement n°325188

Deliverable	**D1.7.2 – Data Format Specification**
Work package Due date Submission date Revision Status of revision	WP1, Requirements and Specifications 30/07/2015 28/07/2015 1.0 Final
Responsible partner Contributors	DFN-CERT Services GmbH KU Leuven TU Delft CARNet FKIE Various project partners for the questionnaires and data schemata
Project Number Project Acronym Project Title Start Date of Project	CIP-ICT PSP-2012-6 / 325188 ACDC Advanced Cyber Defence Centre 01/02/2013

Dissemination Level	
PU: Public	X
PP: Restricted to other programme participants (including the Commission)	
RE: Restricted to a group specified by the consortium (including the Commission)	
CO: Confidential, only for members of the consortium (including the Commission)	

Version history

Rev.	Date	Author	Notes
0.1	21/07/2014	DFN-CERT	First draft based on D1.7.1
0.2	29/07/2014	DFN-CERT	Including data schemata
0.3	31/07/2014	DFN-CERT	Including data workflows
0.4	07/08/2014	DFN-CERT	Schema adjustments on bot, fast_flux, and malicious_uri Adding dependencies in schema descriptions Specification of confidence_level Finalising schemata version 1
0.5	27/08/2014	DFN-CERT	Streamlining text Referencing malware prevalence Draft of end customer format Extending workflows
0.6	24/03/2015	DFN-CERT	Adding data submission and retrieval workflows Detailing end customer notification format
0.7	12/06/2015	DFN-CERT	Adding data format for aggregation Adding data submission and notification feedback from WP3 Adding requirement analysis for custom data formats
0.8	10/07/2015	DFN-CERT	Incorporating QA feedback Workflow clarification and status update Updating submission and notification reports Incorporating improvements from WP3, WP4, and pilot
0.9	24/07/2015	DFN-CERT	Incorporating QA feedback Extending conclusion Adding appendix for aggregated schemata
1.0	28/07/2015	DFN-CERT	Final QA Adding aggregated schemata

Glossary

ACDC	Advanced Cyber Defence Centre
ASN	Autonomous System Number
C2	Command and Control Server
CCH	Centralised Data Clearing House
CERT	Computer emergency response teams
CPE	Common Platform Enumeration
CSIRT	Computer Security Incident Response Team
ISP	Internet Service Provider
LIR	Local Internet Registry
MTA	Message Transfer Agent or Mail Transfer Agent

Table of contents

Table of figures

1. Executive summary

The aim of the ACDC project is to set up a European Advanced Cyber Defence Centre (ACDC) to fight botnets. To reach this goal, the project introduces components and workflows to gather and analyse data originating from technical sensors such as honeypots and IDSs as well as user reports. The Centralised Data Clearing House (CCH) plays a central role in the ACDC project. It provides a platform for storage and analysis of data gathered by diverse sensors. The data does not solely vary by technical sources, but also by different user groups involved. Since each user group and technical sensor has different requirements, the choice of applicable data transport formats is a crucial task. In this document the relevant data formats are assessed, and a common representation of data in the context of ACDC together with data submission and retrieval workflows is defined.

We collect a list of technical, organisational, and legal requirements from an analysis of publicly available specifications of data formats as well as the projects' evaluation tasks and the legal framework.

All ACDC partners contributing data sources or sinks filled in a questionnaire regarding the data formats used by their software, describing their use cases, properties, and planned extensions. The answers to this questionnaire reveal that a total of 15 different formats are in use by the partners. Almost all of these formats rely on a plain text representation of the actual data. Ten of these formats are structured and can hence be processed automatically. Seven of those implement a publicly available specification of their syntax and semantics. The responses include well-known formats such as IODEF, X-ARF, and STIX. Other formats are devoted to a specific use case like the sFlow format for transferring NetFlow data and hpfeeds, which is specific for honeypot data.

Developing a set of use cases relevant for the ACDC project from the collected requirements suggests that the data formats already meet the demands of the project. Experiences with the pilot of the CCH as well as discussions inside the project, however, support the concern that the multiplicity of formats used raises the bar for submitting data to as well as consuming data from the CCH. Hence, we developed a lightweight representation of data in the context of ACDC. This format provides a guideline on what information must or should be provided for partners who submit data. On the other end, the effort required for partners that consume data from the CCH is reduced significantly. The new data format is complemented by a set of workflows for data submission and retrieval based on the type of information transferred as well as the role of the submitting or retrieving party. An analysis of the experiences from the pilot deployment identifies future areas of improvement.

2. Introduction

The intention of the ACDC project is to set up a European Advanced Cyber Defence Centre (ACDC) fighting botnets. ACDC's approach is to

- foster extensive information sharing across borders to improve the early detection of botnets,
- provide an extensive set of ACDC Tools and Services accessible online for mitigating ongoing attacks,
- use the pool of knowledge to create best practices that support organisations in raising their protection level, and
- create a network of cyber defence centres all across Europe.

ACDC will deploy a comprehensive set of national support centres throughout eight Member States interconnected to ACDC's Centralised Data Clearing House (CCH). Through this networked approach, ACDC will also pave the way for a consolidated approach to protect organisations from cyberthreats and support mitigation of ongoing attacks through easy access to an increasing pool of ACDC Tools and Services.

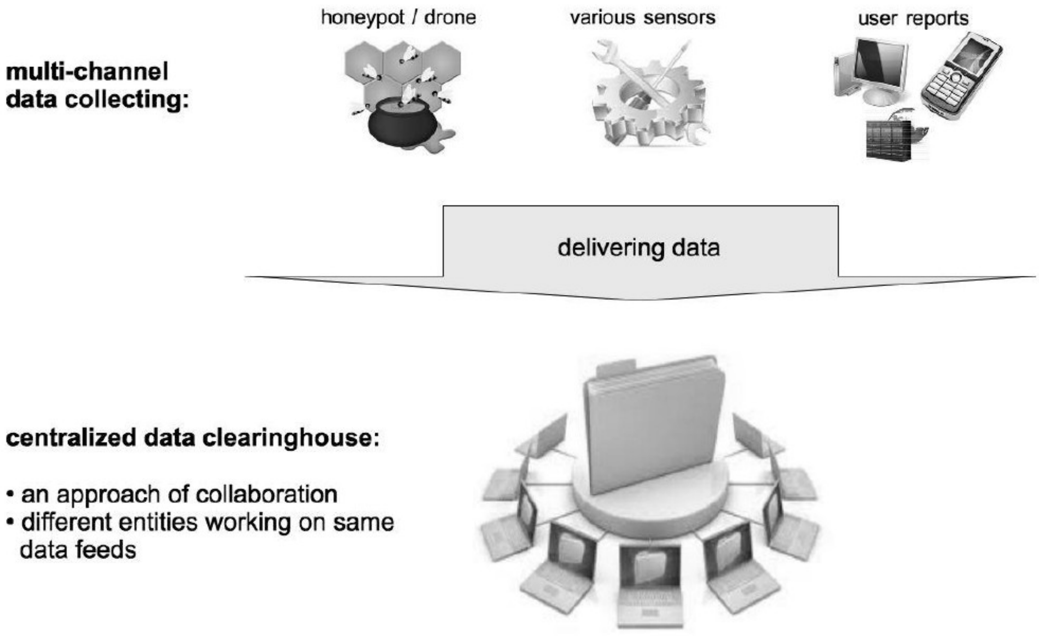

multi-channel data collecting:
honeypot / drone
various sensors
user reports
delivering data
centralized data clearinghouse:
• an approach of collaboration
• different entities working on same data feeds

Figure 1: Sources for data distribution into the Centralised Data Clearing House

Figure 1 shows that ACDC's Centralised Data Clearing House collects all data gathered by technical sensors such as honeypots and IDSs as well as user reports. As a fundamental advantage, the CCH provides a central platform to analyse and process the data allowing to completely reveal botnets and other global incidents by attack data correlation and to distribute the resulting enriched data. It is important to note, that the data does not solely vary by technical sources, but also by different user groups that are involved.

All this results in different requirements regarding the data exchange and processing. For example, some technical sensors produce large amounts of attack data requiring an efficient way for their submission. Some user groups might contribute data intended for reporting security incidents and supporting research. While reporting incidents could not be done without exact information, legal restrictions might require an anonymization for any other usage of the data.

These differences in requirements can be addressed in two ways. Either with one format flexible enough to accommodate them depending on the context or with a bunch of formats contributing the different properties as required. This document aims to collect a list of formats that are already in use in the ACDC community and to investigate whether these formats meet the demands of identified use cases. This analysis leads to the introduction of a new set of lightweight formats that provide a low barrier to interface with the ACDC CCH while still being flexible enough to convey specialised information depending on context. The set of formats is complemented by a definition of multiple workflows, formalising the interaction of different stakeholders with the CCH.

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119].

2.1. Structure of the Document

The document is structured as follows: Section 3 gives an overview of identified technical, organisational and legal requirements to be used for evaluating data exchange formats. In the following Section 4 we describe the properties of data formats resulting from a survey wherein the partners contributed information about the data formats in use. Therefore, these formats can be expected to be relevant for the ACDC project. Section 5 gives an overview of the conducted survey. To decide which data formats are applicable we summarise relevant use cases in this section and enumerate their fundamental requirements that must be met by a data format. Considering these requirements applicable data formats are listed for each use case. The results of the survey are shown in Appendix A.

The ensuing Sections 6, 7, and 8 define a lightweight data exchange format for submitting and retrieving data from the ACDC CCH. This format includes schemata for plain incident reports as well as for aggregated and statistical data. Section 9 defines a translation of the incident reports to the established X-ARF format, targetting abuse notifications for end customers.

Section 10 introduces data submission and retrieval workflows to formalise the interactions with the CCH based on the Cyber Positioning of the interacting party and the data transferred.

The document closes with the conclusion in Section 11.

3. Data Formats and their Requirements

In this section common requirements for data exchange formats are specified and explained. We split the content of this section into five different categories of requirements: user groups, technical requirements, operational requirements, content requirements and legal requirements. The presentation of criteria is concluded in a final section.

The requirements in general apply to the data exchange as a whole, comprising the data exchange format as well as the used communication protocol. Consequently, requirements can be satisfied by using a data format with suitable properties or by employing an appropriate communication protocol.

The requirements are listed as presented. There is no particular order of relevance, neither of nor within the categories themselves.

3.1. User Groups

This section lists the user groups participating in the ACDC data exchange. Each participating user or system may have a different set of requirements depending on its involvement, data exchange scope and communication amount with the CCH.

This list is by no means complete and not all listed user groups will actually transmit or receive data from the CCH. For a more detailed description of each user group see ACDC WP6.

3.1.1. End Users

For all users their privacy must be respected and protected. Therefore, legal requirements as stated in Section 3.5 must be observed. Anonymization is as a best practice implemented by the reporting ACDC Tool itself as stated in Section 1.1.6.1 in ACDC's Description of Work [DoW].

Whereas in other cases users want their personal data to be handled for involvement or notification purposes. In this case the appropriation of the provided personal data must be observed. Most likely users wants an immediate classification or result of the malware's analysis they reported. If this is not feasible, they might want to be informed when a manual analysis is completed.

Also, for information purposes, the end user is to be considered a receiver of data from the CCH, especially the results of analyses. This also includes the public domain unless access to this information is to be restricted.

3.1.2. Internet Service Providers (ISPs)

ISPs can submit data on malicious traffic patterns to the CCH. Furthermore, the ISP should contact the customer responsible for the traffic to remove malware, for example using ACDC's support centres.

ISPs might also be informed of malicious hosts within their network to contact the customer to sanitize their infected hosts. Since all ISPs provide abuse contact details, this data can be used to report detected malicious hosts. Therefore, ISPs contacted in this matter do not have to be a contributing partner to this project.

3.1.3. Intrusion Detection Systems (IDSs)

The information gained from analysis can be used to define new intrusion and attack schemes for intrusion detection and help in discovering weaknesses used for an attack. Therefore, intrusion related data could be shared with the CCH, but will most likely have to be cleaned from identifying data. Depending on the amount of data this additional analysis might not be manageable and therefore not uploaded to the CCH. But this data could be extraordinarily helpful.

IDS/IPS can also be updated using recently discovered malware (or malware behaviour). Therefore, IDSs/IPSs should also be considered receivers of analysed and classified data from the clearing house. This update process, however, will most likely not be automated, since system administrators will not simply trust traffic considered illegitimate by someone else. The update procedure might involve the vendor of the IDSs/IPSs.

Drones

Information security (ISEC) specialists also use drones, infected hosts receiving commands from the botnet's Command and Control (C2) server but not executing them. With these drones they are able to record the commands sent out to the botnet's bots. The captured traffic pattern may then be used to identify botnets in legal network traffic. The result may be shared with the CCH.

Honeypots

High-interaction honeypots are vulnerable hosts placed to be infected by malware. Most often the operating system is running on a virtual machine, so the operator is able to trace the operations used by the malware during infection and execution. Using a virtual machine she is also able to take memory snapshots and have a special environment (called sandbox) set up from which the malware cannot infect other hosts but at the same time execute the commands of the C2 server without any effect. At that stage the infected host in the honeypot works as a drone. Additionally, low-interaction honeypots are used to detect attacks and to capture malware. In contrast to a high-interaction honeypot, this type only simulates a vulnerable service.

3.1.4. Analysts

The ACDC architecture incorporates analysts working with the data stored in the CCH. They perform—mostly automated—analyses of the provided malware or reports and classify them. But they will most likely also perform manual analyses of malware or reports that cannot be classified by software. On the one hand these analysts take data samples from the CCH, on the other hand they also provide further data or information to be stored in the CCH or they enhance reports with results of their analyses.

The group of analysts also includes ISEC specialists as well as academic and business researchers. Depending on the type of research performed, the latter ones may be able to work with anonymized or pseudonymized data.

3.1.5. Computer Security Incident Response Teams (CSIRTs)

CSIRTs (or CERTs) are teams of IT security specialists dealing with incident handling and Information Security Management. They are highly interested in up to date data on malware spreading and recently upcoming malware.

While CSIRTs that are larger, or more successful in networking already have their channels receiving information about malware infections shortly after discovery, especially smaller or more isolated CSIRTs might be interested in data exchange.

Depending on their contractual situation CSIRTs might not be able to provide data on malware infections but are interested in receiving data of active malware.

3.1.6. CCH Operator

The operator of the Centralized Clearing House will be able to access all incoming and outgoing data. So organizational controls must be implemented to establish access control.

As the operator of the CCH service does not provide any transmitting or receiving of malware reports other than those required for the operation of the service the operator is included in this enumeration for completeness only.

3.1.7. Vendors

Vendors of Anti-Virus (AV) or Firewall (FW) software or appliances are interested in malware. They are interested in malware samples as well as the resulting analysis to close vulnerabilities in their products. Vendors of Operating Systems (OS) are interested in exploitation data too.

These vendors might, however, not be willing to share vulnerabilities or weaknesses of their products.

3.1.8. Law Enforcement Agencies

Law enforcement agencies (LEA) might want to access reported data in case of ongoing investigations or by court order. Whether there are legal requirements or even an obligation to share data with LEAs is covered in more detail in [D1.8.1].

3.1.9. Anti Botnet Initiatives

Anti Botnet Initiatives like the ACDC consortium might be interested in data exchange as well. These include national anti botnet initiatives as well as national anti botnet advisory centres which are not part of the ACDC consortium.

3.1.10. Private Hosting Companies

This includes Website operators, hosting companies, data centres and domain providers.

Website operators must be notified when their Content Management System (CMS) is infected to clean the system and stop it from spreading malware any further. To contact the website operator, there must be abuse contact details available. If not, the hosting company must have established an abuse contact team to handle the take down notice or cleaning request. This abuse team has to contact the website owner or take down the website themselves depending on the severity and the contractual situation.

Hosting companies might often be a faster replying and more reliable point of contact when trying to take down an infected Internet service. While they are scanning the network traffic data centre operators do want to prevent damage and illegal usage of its infrastructure. Therefore, also data centres might be able to share their findings of illegitimate traffic with the CCH. They might, however, not be willing to share their customers' traffic with the public. Data centre operators as well as hosting companies are most likely operating IDSs and IPSs to protect their infrastructure and their customers' data.

Domain providers might be willing to take down fast-flux domains, for which the domain serving IP address is changed frequently. These domains are most likely used for malware distribution because it's more complicated to predict the next IP address the domain points to in the future. Therefore, the easiest way to take down the malware distribution centre is by shutting down the domain and its Domain Name System (DNS).

3.1.11. Industrial Users

System administrators might be interested in receiving data about recent malware spreads to update their heuristics and IDSs/IPSs but are not likely to share malware infections with someone outside of the company unless required by law.

This is especially valid for enterprises operating critical infrastructure or banking companies since these might be even less willing to share security related infections resulting in negative publicity and therefore loss of customer trust.

Small and medium enterprises (SMEs) might be outsourcing their IT infrastructure as well as security related services. Therefore, they might not have access to relevant data for sharing.

So we assume this category might only be interested in receiving data.

3.1.12. Press and Media

Press and Media related services play an important part in today's society. They uncover incidents and investigate cases brought up by notifications and information provided by whistle-blowers. Therefore, press and media might be a provider of data for the CCH and on the other hand a receiver of statistical data or exemplary cases.

In other cases the press must be used by companies dealing with personal data to inform the public on issues of loss of data. This holds for example when other means are not appropriate or the number of people affected by a loss of data is so great that it would not be efficient or possible to notify each person individually.

3.2. Technical Requirements

Technical requirements are covering details needed by machines to work with the provided data. We split the criteria into a set of recommended criteria and additional or optional criteria.

3.2.1. Set of Recommended Criteria

Machine Readable Data Format

As with all data exchanged by computers, the data transmitted must be structured. For this, structural elements must added to the content so machines can separate the data fields. Those structural elements must be used according to a defining standard for the data format's language. See Criterion “Text-Based Data Format” for more details on this matter.

This criterion is obvious and must be met, since reports written in human language might not be parse-able for machines and therefore not understandable. And as the amount of data to be transmitted cannot be handled manually the format should be machine readable.

The data format should be validated whether it is the “right thing”. For this, a formal validation is to be carried out to decide whether the data format and its data fields are sufficient for the intended purpose.

Text-Based Data Format

In general, we recommend a text-based data format for reporting attacks and incidents since these ease the encoding and data format issues handled later on in this report. During development text-based data formats are more suited for bug tracking and for creating erroneous situations, falsified messages that are not compliant to the standard.

Also, the focus of exchanged data is on text-based components, but as the text elements have different data field lengths the advantage of binary formats (directly accessible data fields due to offsets) diminishes. Since encoding standards for binary data in text-based formats exists (e.g. Base64 encoding) it is possible to transmit binary data like malware samples or screenshots within the data exchange format. It is important to select a data format that supports attachments (even though this is a more content related requirement).

A text-based data format is not the best format for storage of the data since for iteration over the reports a database-based solution is much more efficient. But as the data format in this case is only for exchange of data the less efficient storage format is not important.

Text-based data formats are much easier to extend as existing parsers can ignore the additional parts. But changes to the data format become backwards incompatible if mandatory parts in a previous version are not available in a more recent one. A text-based data format is usually defined by a Document Type Definition (DTD, for XML), defining the legitimate usage of blocks and elements. These can be used by parsers and serializers to verify their input and output, respectively. Those definitions can also be used to generate parsers automatically without any implementation issues.

Even though a text-based data format is slower to process (parsers for binary data can use fixed byte positions for faster access to selected data fields) this is likely only done once upon receiving the message. Despite the complexity developing a parser for text-based formats messages in this case are rather simple and the data formats in use are already well known, so robust parsers most likely already exist.

Text-based formats are platform-independent, whereas binary encoded messages have to define the range of numbers or how data is to be interpreted by the parser (endianness of word/integers).

Internationalization

The data format should support internationalization (i18n) and localization (l10n) since this is a European and therefore multi-language project and not all end users will be able to state their report using the English language. So the data format must be able to transmit characters of all European languages (specified by the encoding of the data file, e.g. UTF-8). The receiver of the file must be notified of the file's encoding upon transmission, to use the right encoding for decoding. Otherwise, characters might be missing or displayed wrong in the text possibly leading to false data.

Next to the encoding issue is the criterion that reports might be multilingual whereas some parts might be in one and other parts in another or several other languages.

Ensuring Security and Message Safety

The message's security and safety could be handled by the underlying communication protocol but if that protocol may not be able to guarantee these factors they can be handled by the data format itself.

The data message must have some data fields to verify the message's integrity. This is usually done with (cryptographic) checksums. The message's integrity is saved if the message was not altered since checksum creation.

For some message contents protecting the confidentiality during transmission might be valuable. Therefore, the content of the message is encrypted, leaving the problem of encryption key distribution. Usually a public key infrastructure is used but in terms of criteria for the data exchange format it is merely important whether the data format supports encryption if the underlying communication protocol does not.

It would be wise if the sending and receiving peers were able to verify the other peer's identity. This could be done with a public key infrastructure and encryption. Usually the authentication is handled on communication protocol layer.

Another important task is to prevent message duplication since this could lead to denial of the service. As this is usually established on protocol layer, other means must be used to prevent a message being send several times. This could be a unique message id or a limited timeslot for which the message is valid.

Documentation Of Data Format

To prevent misunderstandings due to different interpretations of the provided data the documentation of the data format must be up to date and the specification of the data format must be unambiguous.

This is important for each data type and each data field. E.g. whether the string value of 'true' in the transmitted text-based message is interpreted as the boolean value of true, the string value of 'true' or as the integer value 1.

Especially important is the format of timestamps. These data fields should incorporate the time zone of the host. Usually this is not a problem when timestamps are formatted as strings but it turns into a problem when timestamps are transmitted in seconds since a specific date.

Supporting Bulk Messages

If possible, the data format should support the transmission of several messages in one bulk message. Therefore, the data format may act as a container comprising messages.

The receiver must decide whether to separate the messages into several reports or one large report. There could be a connection between several findings on one machine, which is a piece of information that would get lost if the message is split (See criterion in Section 3.2.2 under “Link between reports”).

Supporting IPv4 and IPv6

If data formats use data fields for IP addresses the data format specification should be able to handle IP addresses of version 4 as well as IP addresses of version 6 for future use and long usability.

As an IP address date must be considered as possibly human related there must be means to anonymize the field and indicate the anonymization to analysts.

Vendor Independence

The data format specification should be free of charge and defined in a free and open format. The reason is to not become dependent of the good will of the vendor or the software solution.

Using an open format allows exchanging the used software solution if the software is abandoned, or not sufficient, or not compliant to the projects' goals any more. An open format also allows extending and adopting the data format for the project's requirements as well as implementing parsers for all platforms, especially for new, emerging platforms possibly in competition with a vendor.

Version Tag

A data format needs a version tag to support extendibility, whereas a parser for a later version should be able to read a report formatted in a prior version of the data format. The version tag can also be used to identify backward and forward incompatible changes in the data format specification.

A version information should be standard in every data format specification as it is required for further improvement and development.

3.2.2. Set of Additional Criteria

There are two criteria which are merely optional: Using an object-oriented data format and support of compression.

Object-Oriented Data Format

As the object-oriented notation is the natural way of describing items, it might be wise to select a data format supporting this notation. The object-oriented notation includes attributes for objects and also nesting of objects. The applicability of object-orientation depends on the structure of the reports to transmit. For simple reports or data structures an object-oriented orientation is overload.

Support for Compression of Content

The content of a report could be compressed to save bandwidth during transmission. While executing compression on mobile devices might reduce the battery by a larger amount than saved by less data transfer. So compression should be optional in terms of using it if the data format provides it.

It should be considered, that compressed data must be encoded text-based when transmitted in a text-based format. Therefore, the compression might not save that many bytes since the encoding adds bytes (e.g. see Base64-Encoding).

3.3. Organizational Requirements

Another set of criteria to evaluate diverse data formats are organizational requirements. These deal with individual requirements raised by users on usage of their provided data, support of the ACDC workflow and licensing issues. As done with the technical requirements these are split into recommended and optional criteria.

3.3.1. Set of Recommended Criteria

Anonymization

As required by law in some countries (e.g. Germany), all data relating to a person must be anonymized. Exceptions for required anonymization of data are the user's consent whereas the consent must be free of choice, or if there is a law requiring the data. The last exception occurs if there is a legal binding between the user and the storage unit.

While this is easy to handle for the submitting user, it's much harder to achieve for the attacking or third party. Please see the legal requirements section in Section 3.5 for further details and Deliverable [D1.8.1].

Well-Defined Syntax and Semantics

As already stated in Section 3.2.1 under “Documentation of the data format” the data format must be well-defined for unambiguity. That includes well defined syntax and well defined semantics. Whereas syntax defines the construction of valid documents, semantics define the meaning of elements and how to interpret them.

Having well defined syntax and semantics allows validation of the data format (Whether the data format allows all required data to be transmitted) and automated verification of reports based on the data format (Whether the reports are legitimate based on the data format specification).

Individual Requirements

As individual requirements on the usage of provided data might arise it would be a wise choice if the data format supports the specification of individual requirements concerning the data usage. Therefore, the data fields or alternatively the protocol to submit data should provide attributes allowing to state requirements or restrictions using some formal language.

Please see ACDC's Deliverable [D6.1.1] for an analyst's point of view on this topic. Each partner who requests to analyse data has to define what he intends to do with the data to receive. Those intended purposes could then be selected by the submitting user.

There should also be a possibility of applying timing constraints, which are executed after a defined period. Those might include anonymization or deletion of provided data. There might also be a legal requirement to anonymize or delete data after a period defined by law or court order (e.g. seven days in Germany for logging data used to detect malicious behaviour or fraud). These timing constraints could be set by user or by law.

Confidentiality

In addition to technical requirements of message safety and content security, there exists an organization requirement to ensure that only authorized users are able to view the provided data. This means to employ access restrictions to the message's content on all levels either by encryption of the communication protocol or in the software environment at the CCH.

Data fields are needed in the data format to support specification of confidentiality restriction.

Appropriation of Human Related Data

The data format shall ensure that provided data is only used for the intended purposes stated unambiguously and well-defined during survey. So called appropriation must be ensured at all levels at the transmission and storage at the CCH and during evaluation. This is also a legal requirement introduced by law.

All additional purposes are generally forbidden by law. But narrow exceptions do exist.

Support of the ACDC Workflow

Where applicable the data format shall support the ACDC workflow, meaning that data fields should match required data fields for analysis or even finer but not more coarsely so the analysing software would have to split the data manually.

Providing the data in the right data fields supports automated analysis.

Extendable Data Message

The data format shall be extendable. That includes data fields (e.g. for additional free text) defined by a data format extension. These might also include vendor specific extensions to a data message (e.g. name and version of the software creating the report).

Therefore, the specification should not be too tightly tied down, but to allow individual improvements or adjustments. Additionally, the parsers must be adopted to the new format. Depending on the specification actually used, the parsers are generated automatically.

Licensing Issues

As already stated above in Section 3.2.1 under “Vendor Independence” the data format should be in a free and open format to be free of charge. Especially after the initial project phase as a pilot project and funding by the EU costs should be kept down.

But not only the costs are dependent on the license, also the usage of the data format might be limited by the publisher/owner of the data format. Therefore, a free specification is the best option, especially when the data format is to be extended. This might not be allowed in all licenses especially when the license is proprietary.

3.3.2. Set of Additional Criteria

Restrictions on Subsets of Data

If possible, the data format should allow to define restrictions only to subsets of data in a report. Therefore, the data format should be flexible to add restrictions to any subset of data.

These may include timing constraints as stated in Section 3.2.1 under "Individual Requirements".

This criterion is marked as additionally since most formats will not support it.

Stating Confidence in Report

To prioritize manual analysis of important reports the data format might be able to allow the user to state the confidence into the finding and the severity of the incident. On one hand this would result in an enhanced user involvement but could on the other hand lead to inexperienced users stating the problem as severe whereupon the problem is merely an annoyance.

Therefore, the user's profile could be equipped with a credibility score determining the experience of the user. But this would require users to register with the service and lead to problems as how to deal with first time submitting users being ITSEC professionals.

Linking Related Reports

The data format could allow links to related reports happened before or at the same time (see also Section "Supporting Bulk Messages") to establish relationships between reports. The connection between related reports could be established manually or by automated analysis.

The intention of establishing connections between reports is to create a data-warehouse dealing with 'big data' to gain even more information from reports.

3.4. Content Requirements

As discussed in Section 3.3.1 and in [D1.2.2], input data formats supported by CCH ought to guarantee anonymity and privacy. However, we should not neglect the impact of these security requirements in the quality of the analysis in WP4 and in other work packages. For example, consider anonymizing IP addresses using a certain mathematical function (e.g., hash function). While we will be able to tell how IP addresses' measure of evilness changes over time, we will not be able to tell *what* addresses they are and, consequently, will not be able to use the output in real-time IDS.

In this sense, we should evaluate each data type on a *case-by-case basis*, i.e., what fields must be anonymized to keep both user's anonymity and privacy while, at the same time, keeping the highest quality of analysis possible.

For example, consider a spam message. In this case, one could consider removing both sender and destination email addresses from the contents, or the entire content of the message, keeping only the metadata (e.g., source/destination IP addresses, timestamp, etc.). However, to keep the user's privacy in the metadata, one could remove the last octet of the IP addresses (e.g., 192.168.0.x instead of 192.168.0.53). Another example is the case of DDoS attacks reported using IPFIX/NetFlow [RFC 7011]. For this case, the report will only list the metadata, and not the message contents associated with the attack.

Taking into account the heterogeneity among data sources and exploited applications, we recommend:

1. Define unique anonymization functions for fields (or a single function). The functions *must* be consistently used across all datasets to enable correlation between various data sets.
2. For each type of data and format, evaluate which fields compromise both anonymity and privacy (or fields that the contributor requires to be anonymized).
3. Then, determine if they must be anonymized. If yes, then employ functions defined in 1.
4. Evaluate the results to ensure privacy and anonymity.

It is important to emphasize that this task should be executed interdisciplinarily, considering both technical and legal requirements.

3.5. Legal Requirements

This section is a short summary of the D 1.8.1/2 – Legal Requirements, which will clarify the rules applicable to the project and in particular to the ACDC tools for mitigation and detection to be deployed. The processing of personal data (any information relating to an identified or identifiable natural person, e.g. IP address, email address, etc.), requires observation of stringent protection rules. As a result, partners involved in this processing must comply with the principles of legitimacy, data accuracy and finality, proportionality, confidentiality and security, and transparency.

3.5.1. Legitimacy of Processing

The legitimacy of processing lies on the unambiguous, specific, freely given and informed consent of the data subject (person to whom the data relates). In principle, the partner with whom the end user has a direct contractual relationship (or is subject to in the case of a public mandate) is best placed to register user's consent as far as the collection or release of her/his personal information is concerned.

3.5.2. Data accuracy and Finality

The data accuracy requirement entails the data controller obligation of putting in place mechanisms and procedures that ensure the reliability of the personal data he/she processes. Furthermore, the principle of finality or purpose limitation dictates that the authorized usage of personal is restricted to the specified, explicit and legitimate purposes for which it was first collected.

3.5.3. Proportionality

Data processing also needs to be proportional, implying that:

1. There must be a sufficiently narrow correlation between the (legitimate) purpose articulated by the controller(s) and the data being collected;
2. Personal data should only be disclosed or otherwise made available to the extent that it is necessary to achieve the purposes of the processing;
3. Personal data should not be maintained longer than is necessary for the purposes for which the data were collected and/or further processed;
4. Controllers should seek to minimize the number of copies of personal data being processed;
5. If the purposes of the processing can also be realized by less intrusive means, i.e. by means which are less likely to have an adverse impact on the privacy or other fundamental freedoms of the data subject, such means should be used;
6. Even if legitimate, the processing may not prejudice the data subject in a way that is disproportionate in relation to the interests pursued by the controller.

3.5.4. Confidentiality and Security

The data controller is obliged to implement appropriate technical and organizational measures to ensure the confidentiality and security to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access.

3.5.5. Transparency

The transparency principle gives data subjects the right of being notified of the processing of their personal data (notice), having means to obtain further information (right of access) and immediate tools of recourse towards the controller in case the data subjects feel their data are being processed improperly (right to rectification, erasure or blocking).

3.5.6. Conclusion of Legal Requirements

Fewer legal restrictions apply to data that cannot be related to an identified or identifiable person, as they are out of the scope of data protection regulation. Overall, it means that non-personal or anonymized data require less legal consideration according to their usage and can be processed in a simpler way. Whenever the relation to a person is not required, partners are asked to convert the processed personal data into anonymized data, a form which does not identify individuals and does not allow re-identification through data matching.

Anonymized data shall be preferred whenever it does not significantly harm the outcome of the ACDC Tools for mitigation and detection.

3.6. Conclusion of Requirements

The technical and organizational requirements are merely sets of criteria that should be met by the data formats already in use. If a data format is to be selected from scratch, these criteria can be seen as an evaluation catalogue on how to find the most suitable data format. In this not all criteria have to be met, for some there are workarounds by implementing the requirement at the communication protocol layer.

While technical and organizational requirements are basically defined by technicians implementing the data exchange, content and legal requirements are defined by analysts and the legal situation, respectively. Whereas analysts try to get as much data as possible for the analysis, the legal department determines what data can be obtained and which processing (e.g., anonymization or pseudonymization) can be done with the data depending on the processing's legal background. These are two very diverse vantage points and the result—if and to which extend each data field could be used for analysis—will be somewhere in between. With anonymized data analysis is hardly feasible (or the results lack quality or significance), but breaking the law is no option either. So it has to be defined—as expressed in Section 3.4 in the second paragraph and Section 3.5.6, respectively—to which extend data must be anonymized to be legitimately used for processing at all. This results in a trade-off between what is allowed and what is required, obeying the risk mitigation strategy in Section 2.4.3 in [DoW].

4. Stocktaking of Relevant Data Formats

This section gives an introduction to the relevant data formats. Because of the massive number of formats, we focus on a selection resulting from a survey by which a questionnaire was distributed among the ACDC partners. The data formats can be structured using different choices of criteria. In this section we classify the formats according to the encoding which can be either binary or textual. A binary encoded data format can, for example, be related to a structure in the programming language C. Thus, the data records are structured according to a C variable structure. Textual formats use a character encoding such as ASCII or UTF-8. They can be either formally structured by using XML or lack a structure at all.

4.1. Binary Data Formats

As previously mentioned binary encoded data formats are typically related to a structure in the programming language C or a similar language. Their most important advantage is the compactness of the messages, because there is no need to use textual metadata separating data fields, as used, for example, in XML. However, they require adequate computer programs for their processing. In general, binary data formats are advantageous to transfer large amounts of bulk data.

4.1.1. IPFIX/NetFlow and sFlow

IPFIX/NetFlow [RFC 7011] and sFlow are part of a family of protocols to transfer metadata related to the fundamental information of network connections (NetFlow) which include the following information:

- IP addresses
- Port numbers
- TCP/IP Flags
- Number of packets in the flow
- Size of the transferred data.

The IPFIX/NetFlow protocols are initially supported by network equipment such as routers to export data of monitored network connections. Therefore, the primary intention of this protocol family is to gather and transfer NetFlow data on routers that may include productive traffic of a large network.

NetFlow data support the detection as well as forensic analysis of computer security incidents. This data is very valuable to react to distributed denial of service (DDoS) attacks. In addition, numerous security tools analyse NetFlow data for anomalies that may be caused by large scale attacks and technical problems. Another important use case is to reveal the full extent of an incident in a forensic investigation. NetFlow data allows tracking connections that are originated by a compromised system. This is especially important to track botnets because this allows monitoring network connections that either target or originate from a control and command server. Botnets are either controlled by a central server or a peer-to-peer network structure. In the first case, NetFlow data allows tracking down other systems that connect to this server. These systems are likely also compromised and controlled by the server. In the second case, NetFlows could help to track the peer-to-peer structure of the botnet. Therefore, NetFlows play an important role in the tracking and investigation of botnets.

4.1.2. Hpfeeds

Hpfeeds is a data exchange format especially dedicated to the exchange of honeypot data. Currently, some honeypots including Dionaea and Glastopf natively support the protocol. The idea behind hpfeeds is to supply and receive honeypot data on “data feeds”. Feeds are implemented as a bus-like architecture and are isolated by different channels. Thus, honeypots supply data to a central bus where receivers can subscribe to data they are interested in. Optionally, the channel can be secured using SSL/TLS. A receiver has to previously authenticate to the server to be able to access the data.

Each message starts with a message header consisting of its length and an opcode corresponding to a specific message type. The message is comprised of an identification number, a channel name, and the payload.

The primary advantage of the hpfeeds protocol is its bus-like structure that makes it very easy to connect new honeypots and receivers to the architecture. Integration only requires to register the honeypot to the appropriate channel. Thus, there is no need to negotiate the data exchange with all or selected sites that receive the data. In addition, a site willing to receive data has only to register and subscribe to the channel.

4.2. Textual Data Formats

Many data exchange formats represent data in textual form. Therefore, the messages can be processed and displayed with all programs that are able to process textual data whereas no knowledge of the structure is required. This is an advantage compared to the previously introduced binary data formats. The structure of the data is given by metadata such as XML-tags that are embedded in the message. Because the structuring data is part of the massage, the structure can be understood without an external specification.

XML as well as JSON introduce schemas to validate its validity. A schema is an external document that provides a formal specification of the structure of all related documents. The schema specifies the structure as well as the data type of each data entity. Thus, the validity can be verified by testing if the schema meets the specification of the schema.

Because most common data exchange formats are based on either XML or JSON we here divide all formats into these categories. XML (Extensible Markup Language) divides characters into “markup” which structures the message and content. Take for example the following line of an IODEF message:

```
<IncidentID name="csirt.example.com">908711</IncidentID>
```

It comprises the markup construct “`IncidentID`” and “`name`” and the content “`csirt.example.com`” and “`908711`”. A formal specification of the structure is given by a Document Type Definition (DTD) or an XML schema.

JSON (JavaScript Object Notation) is derived from the representation of simple data structures and associative arrays in the scripting language JavaScript. In short a JSON message consists of key and value pairs such as “`firstName`”: “`John`” whereas the first part is the name of the data field and the second parts its value. The structure of a message is defined by a JSON schema.

4.2.1. XML

IDMEF

The *Intrusion Detection Message Exchange Format* (IDMEF) as specified in [RFC 4765] is a very versatile data format especially devoted to exchange and transmit data produced by Intrusion Detection Systems such as Snort. The primary use case is to transmit alerts from a sensor to a central management system on which the messages are further processed and analysed. For example, the Prelude SIEM uses IDMEF to enable a distributed network of sensors in that all report to one or more managers which can be hierarchically organised. Sensors are IDSs including Snort or other components that enable to aggregate and correlate multiple alerts. For example, this can be used to detect coordinated port-scans that originate from more than one source.

While IDMEF messages are in principle human readable they are because of their complexity better suited to be processed by programs. This includes, for example, the import of data into a database where the data could be displayed by a web-application. In addition, the format is ideal to exchange alerts between CSIRTs and other security-aware teams such as ISPs.

In short, IDMEF contains the following parts:

- Identification and name of the analyser, e.g. Snort NIDS
- Time of detection and time of message generation

- Information about the source and target of an attack. This includes IP addresses, DNS names, process IDs, and file names.
- A classification of the alert. This comprises the signature that triggered the alert.
- An assessment of the severity of the threat
- Additional data, e.g. logs or other data related to the attack
- Information about correlated alerts

IODEF

The *Incident Object Description Exchange Format* (IODEF) is an adoption of IDMEF that is devoted to the exchange of computer security incidents among CSIRTs. The most important building blocks of an IODEF message are:

- **The temporal extent of the incident:** This includes the time the incident has been detected
- **An assessment of the incident:** A characterization of the impact of the incident.
- **Method:** A description of the method the attacker has been used, for example, to attack the system under analysis.
- **Contact information:** This is, for example, the postal address and telephone number of the reporting CSIRT.
- **The data related to the source and target:** This contains the data concerning all sources and targets related to the incident. For example, relevant data are the IP addresses or DNS information of the attacking and targeted system.
- **Other data related to the attack:** Usually, the reporting site adds data serving as evidence to the report. This includes log-excerpts, NetFlow data, and other information that are related to the incident. Additionally, a complete IDMEF reports can be included.

As previously mentioned, IODEF is devoted to the exchange of security incidents by CSIRTs. This is considered by some data entities that are specific for the requirements of this user group. First, the incident data can include an expectation that conveys to the recipient of the IODEF document the actions the sender is requesting. For example, this can be a request to block a host or to prevent any further abuse. To respect privacy concerns, the disclosure of information can be controlled by the attribute "*restriction".*

TAXII and STIX

These two formats are part of a very comprehensive and versatile framework that have been proposed by the MITRE Corporation. In short TAXII (*Trusted Automated eXchange of Indicator Information*) defines a set of protocols and services to exchange cyberthreat information. The language that provides a representation of these informations is given by STIX (*Structured Threat Information Expression*). While IODEF focuses on the exchange of incidents, the TAXII/STIX framework has a broader view on security incidents. A threat is modelled by STIX comprising the following entities:

- Indicators and Observables: A specific attack typically involves patterns that allow to characterise it. These pattern are, for example artefacts and/or behaviours of interest within a cybersecurity context and are specified in STIX by Observables acting as Indicators for an attack.
- **Incidents:** These are successful attacks detailing the information about the source and target. The related Indicators and Observables of the threat give information how that attack could be detected.
- **Exploit Targets:** Vulnerabilities or weaknesses that enable the attacker to successfully attack a system.
- **Tactics Techniques and Procedures (TTP):** These give an overview on the overall aim of the attack. For example, this can be to use malware to steal credentials.
- **Threat Actors:** A characterisation of the identity, suspected motivation, and suspected intended effects of the attackers.
- **Campaigns:** Usually, attackers do not attack a single target. Instead, they target a specific community or a set of computers or applications. This can, for example, be a set of SSH servers that share a specific user group such as the high energy physicist.

The TAXII framework is used to share the threat information that is specified by STIX. The framework support multiple organisational models to exchange this data. These are:

- **Source-Subscriber:** There is a central instance that provides the data to all consumers.
- **Hub-and-Spoke:** There is a central instance that provides the data to all consumers. However, the consumers can send data to the central instance that retransmits the data.
- **Peer-to-peer:** There is no central instance. Instead, the consumers exchange data in arbitrarily connected networks.

Overall, the strength of TAXII and STIX is the modelling of complex attack behaviour consisting of multiple related steps. It can be expected that this framework is well-suited for modelling threats concerning botnets. Under this aspect, this framework is advantageous to IODEF that serves solely as a data exchange format.

4.2.2. JSON

X-ARF

X-ARF is a light-weight but structured format for the exchange of data related to computer security incidents. In contrast to other formats like IODEF the format is kept as simple as possible. Thus, the aim of X-ARF is to introduce a light-weight and structured format which focuses on the most relevant information and can easily be used and extended. The format is not limited to incident data and can additionally be used to exchange malware, honeypot, or IDS data. An X-ARF message contains human as well as machine readable containers. Therefore, the same message can be used to inform the administrator of an abusive system about the incident and it can automatically be processed by an incident management system without changes. Currently, the format is supported by a growing list of CERTs and a broad acceptance in the academic community can be expected.

X-ARF documents are structured in multiple parts denoted as container. Each container is structurally independent of the other and may contain completely different content. However, the idea is to combine human readable and machine readable parts which contain the same or at least similar information. Therefore, an X-ARF document simultaneously addresses humans, for example, system administrators and allows an automated processing. All specified use-cases consist of three containers, although this number may vary for future specifications. Since X-ARF documents are transferred by email, each container has a specific MIME-type (Content-Type) and a specification of the character encoding (charset) which is usually UTF-8. Currently, three different use cases exist for brute-force, malware and phishing attacks that share the following containers:

- **The first container** is human readable and can contain arbitrary text. Its MIME type is typically “text/plain” encoded in an UTF-8 charset.
- **The second container** contains data that uses the YAML markup language for structuring the content. Thus, technically X-ARF is based on YAML and not JSON. But as these are equivalent with regard to the aspects investigated here—see also the YAML specification on this[1]—, X-ARF is put under the JSON heading to simplify the presentation. A JSON schema exists defining the structure and syntax of the data. This allows testing if an X-ARF document is well-structured and valid in respect of its specification.
- **The third container** is intended for transferring various additional data regarding to the abuse type that depends on the previous specification in the second container. It can include log-data as a kind of evidence for the abuse handler or it may be used for malicious files that are, for example, captured by a honeypot.

The first container is intended to contain a summary of the message in textual form. The second part contains the details of the attack data. It is structured and can be automatically processed by an incident handling system. The fields contained in the second container depend on the abuse type of the X-ARF document. However, all abuse types share a common set of fields that, for example, contain data about the abusive system.

The strength of X-ARF are its simplicity and versatility. The documents contain a machine as well as human readable part to support multiple groups of recipients. In addition, the

[1] http://yaml.org/spec/1.2/spec.html#id2759572

format is kept as simple as necessary to ease its application and assesses a lot of different use cases.

Proprietary formats based on JSON

As previously mentioned, JSON can be used to structure a message and to specify the data types. In the ACDC community data formats exist to submit data, which, for example, includes data gathered by honeypots. Other schema address information about hosts serving malware URLs, location of C2 server controlling a botnet, passive DNS information, Spambots, and IDS alerts.

The advantage of JSON is the easy and efficient definition of ad hoc or highly specialised data formats. JSON enables the quick design of a specialised data format that is, for example, applicable to submit data to a central repository. This is especially important if an appropriate data format for a specific data set such as the data mentioned above is missing.

5. Evaluation of the Data Formats

The aim of this document is to identify applicable data formats that can be used in the context of the ACDC project. Furthermore, the document should propose extensions or improvements if the available data formats do not satisfy all requirements. These requirements have been proposed in the second section. To identify applicable data formats a survey has been initiated to gather information about which formats are already deployed by the ACDC partners and what use cases are addressed. In this section these data formats are evaluated in respect to criteria that are derived from the requirements in Section 3.

The data format survey includes a questionnaire that is partitioned into three blocks (for the complete questionnaire we refer to the Appendix). After the name of the data format, the second block comprises questions concerning use cases the data format is related to. This includes the role of involved sites as well as information about incorporated workflows. Furthermore, experiences are questioned and whether there are demands for extensions or improvements. The third block comprises specific questions about the data format details. This includes properties of the data format as well as any bindings to a specific transport protocol. For example, some formats such as IODEF are designed to be submitted by email. This is important to consider because some aspects such as the data security are in some cases left to the protocol. For example, using S/MIME standard ensures confidentiality, integrity and authentication of IODEF messages. A fundamental information is whether the data is represented in binary or textual form and if there is a formal specification of syntax and semantic. It is important to note, that such a formal specification is crucial for an automated processing of the data. Ideally, the specification is publicly available, e.g. as an RFC document. To ease the deployment, the availability of programs or libraries to create and process messages in the specific data format is required. Ideally, these programs are released under an open license such as the Gnu GPL.

5.1. Evaluation of the questionnaires

Overall, 15 responses were analysed originating from 12 different sites. Nearly all questionnaires refer to different data formats. Only two responses refer to the same data format (IODEF). For the complete set of anonymized responses we refer to the Appendix. The most important result is that most of the used data formats are based on a textual representation and are structured by XML or JSON. Among them are the publicly specified formats X-ARF, IODEF, and STIX/TAXII that are used to exchange data with external sites. Additionally, other proprietary formats based on JSON and XML are internally used. Other specialised formats such as IDMEF, sFlow, and hpfeeds formats have been proposed that are devoted to transfer data of network connections and data gathered by honeypots and IDS such as Snort. The protocols are designed to cope with large amounts of data, for example produced by monitoring large networks. These formats are often internally deployed to transfer the data from a sensor such as an IDS, honeypot, or NetFlow collector to components that aggregate and correlate the data. This process is used to combine multiple events (e.g. IDS alerts) to the full extend of an incident.

The results of the evaluation of the questionnaires are summarised below. The questionnaires are labelled from “A” to “Q” and grouped according to different criteria. For a complete listing of all questionnaires we refer to Appendix. The first part summarises results grouped by the referred data format. The next part classifies results considering a list of major properties. In the following two parts we group the data formats according to supported user groups and data sources. The section concludes with an enumeration of use cases that are expected to be relevant for the project. For each use case the data formats are listed that are applicable.

5.1.1. Overview of the received questionnaires and referred data formats

- Text based proprietary formats (Whois output, FluxDetect tool, Skanna tool, EvidenceSeeker tool): **A, B, C, D**

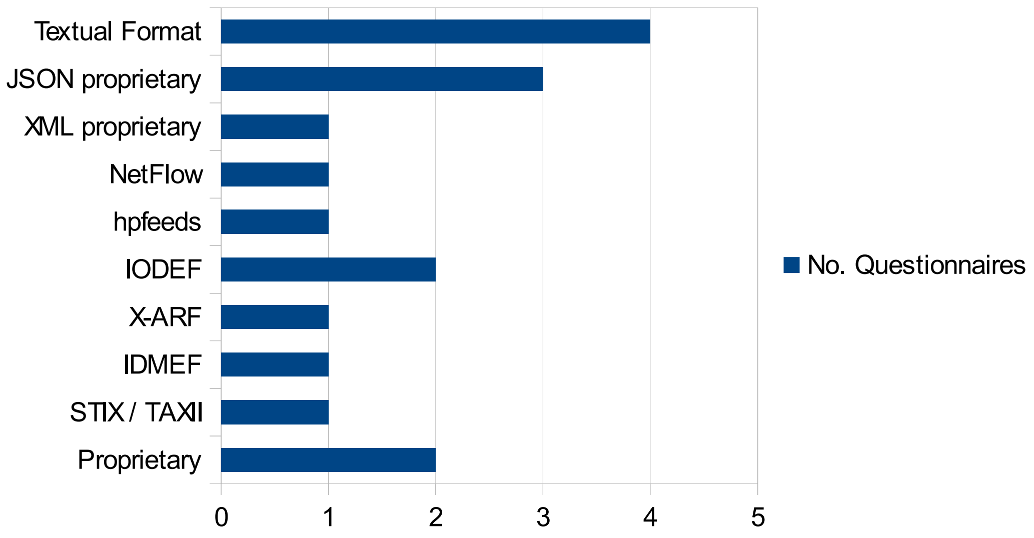


Figure 2: Overview of the received questionnaires

- Proprietary data formats based on JSON: **E, F, G**
- Proprietary data formats based on XML: **H**
- sFlow 5.0: **I**
- hpfeeds: **J**
- IODEF: **K, L**
- X-ARF: **M**
- IDMEF: **N**
- STIX / TAXII: **O**
- Proprietary formats: **P, Q**

5.1.2. Overview of the technical properties of the data formats

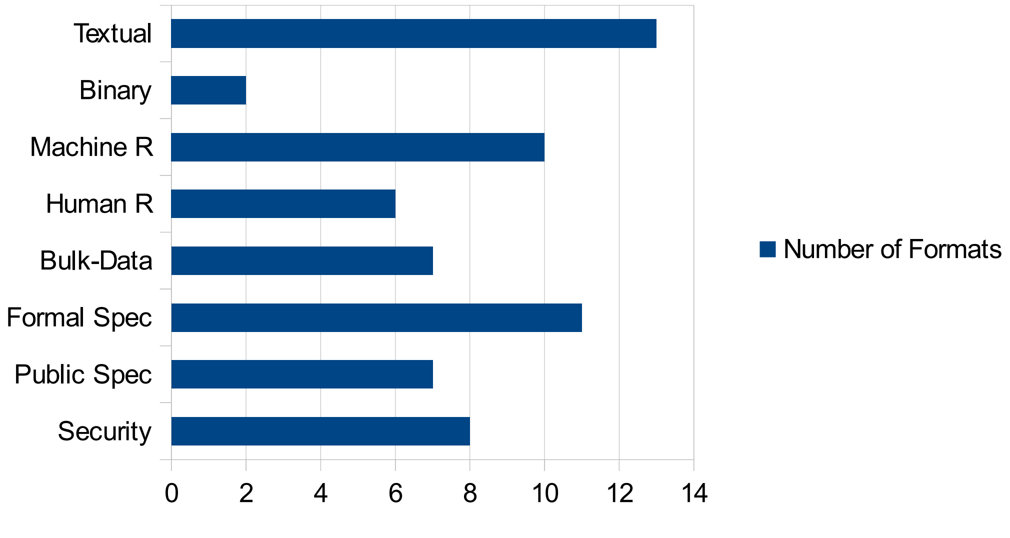


Figure 3: Properties of data formats

- Textual representation: **A, B, C, D, E, F, G, H, K, L, M, O, P**
- Binary representation: **I, J**
- Machine readable: **F, G, H, I, J, K, L, M, N, O**
- Human readable: **A, B, C, D, M, P**

- Capability for Bulk-data / aggregation: **G, H, I, J, M, O, P**
- Formal specification of structure: **E, F, G, H, I, J, K, L, M, N, O**
- Public specification available: **I, J, K, L, M, N, O**
- Explicit support for security aspects: **H, J, K, L, M, N, O, Q**

5.1.3. User groups and their requirements

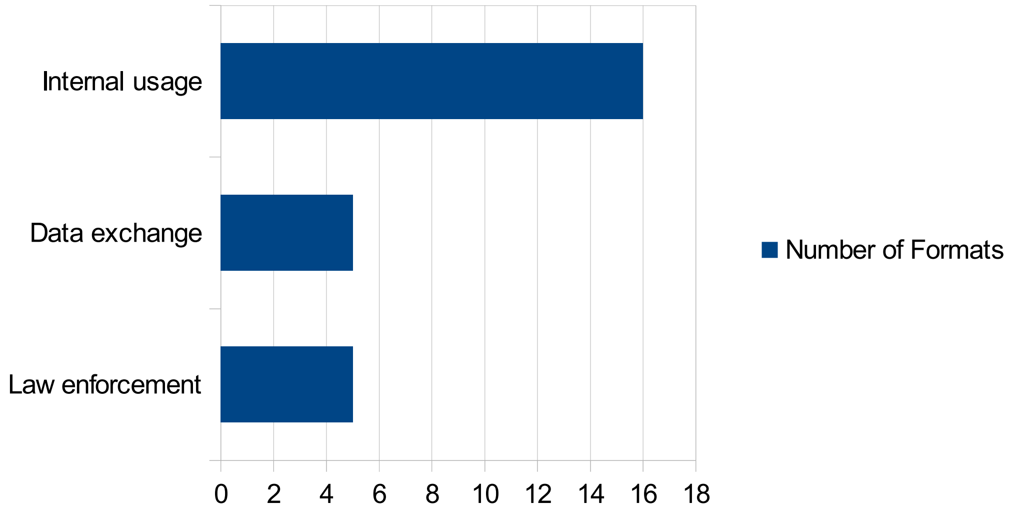


Figure 4: Number of formats supporting a specific user group

- Internal usage in ACDC community: **A, B, C, D, F, G, H, I, J, K, L, M, N, O, P, Q**
- External data exchange with CSIRTs, ISP, Academic/Research, AV: **K, L, M, N, O**
  - Public specification available
  - Textual representation
  - Security requirements met
- Law enforcement agencies: **K, L, M, N, O**
  - Public Specification
  - Textual representation
  - Security requirements met

5.1.4. Overview of data formats and supported sources

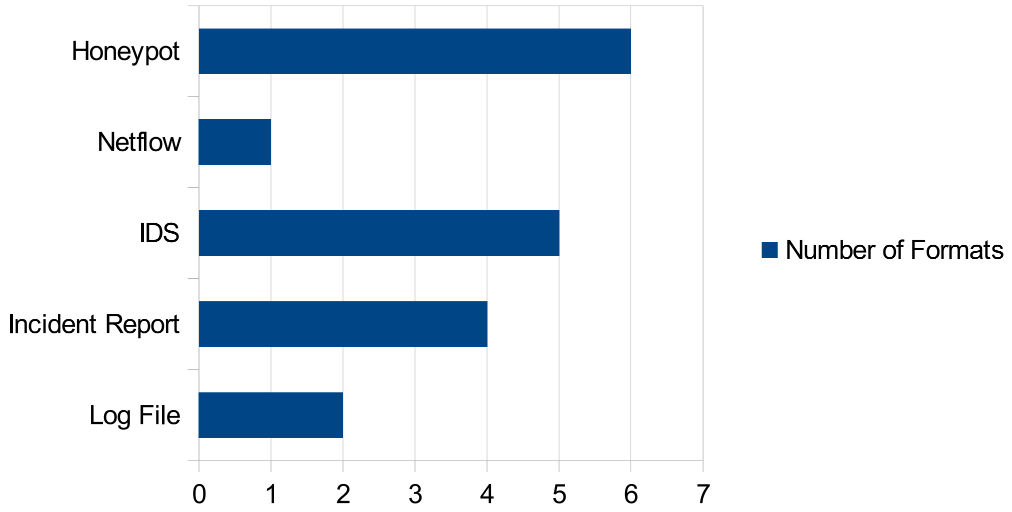


Figure 5: Number of data formats supporting a specific user group

- Honeypots: **E, F, G, J, M, N**
- NetFlow: **I**
- IDS: **E, F, G, H, N**
- Incident reports: **K, L, M, O**
- Log files: **A, P**

5.1.5. Use Cases

A use case is here understood as a scenario where data is exchanged between user groups previously introduced. This includes, for example, the submission of IDS sensor data to a central repository. Our aim is to select typical use cases that are expected to be relevant for the ACDC project. Although a list of general requirements is previously assessed, it is important to note, that each use case may have specific demands. For example, an automated processing of a message requires a formal specification of structure and data types whereas an end user report should be as simple and descriptive as possible. Thus, the selection of a data format cannot be done without a specification of the adherent requirements of the use case that the data format is involved into.

To retain clarity, we use a simplified list of the requirements as described in Section 3. It is important to note, that some requirements are taken from the specification of IODEF and IDMEF. These requirements are explicitly satisfied by nearly all XML and JSON formats. Furthermore, it can be expected that nearly all data formats with a formal specification consider an unambiguous specification of encoding and data types. For the sake of clarity, these requirements are omitted here.

Below the use case and the proposed requirements are listed. After each use case, a proposal of the applicable protocols corresponding to the labelled questionnaires are enumerated.

- Submission of sensor data (honeypot, NetFlow, IDS): **E, F, G, H, I, J, N**
  - Machine readable
  - Formal specification to check correctness
- Data exchange to analyse, aggregate, and enrich the data (internal usage): **A, B, C, E, G, H, I, J, K, L, M, N, O, P**
  - Internal description of format
- Submission of incident (attack) data to central data repository: **E, F, G, H, I, J, K, L, M, N**
  - Machine readable

  - Formal specification to check correctness
  - Security requirements (sensor and recipient authentication)
  - Capability to anonymize
  - Support of the relevant information
  - Support of access control
- Distribution of data and notification to affected/interested stakeholders: **E, F, G, H, K, L, M, N**
  - Textual representation
  - Security requirements (sender and recipient authentication)
  - Public specification
  - Capability to anonymize
  - Support of the relevant information
- Reporting to end users: **M**
  - Human readable
  - Public specification

5.2. Summary of Results

The evaluation of the questionnaires comes to the following conclusions:

- The available data formats support the expected use cases of the ACDC project quite well.
- Shortcoming of the formats are the specification of a fine-grained access control mechanism for specific data entities that may contain person related data. Only IODEF and TAXII consider this. However, the granularity of the access control is very rough.
- Specialised formats are advantageous for the submission of raw data for analysis / aggregation (sFlow, IDMEF) because they support aggregation and/or compression.
- X-ARF is the most versatile format. It is machine as well as human readable and supports all user groups. The format is less complex than IODEF and STIX/TAXII and does not raise the bar for usage. Therefore, it is perfectly suited to submit attack and aggregated data and to exchange data with external sites such as ISPs, law enforcement, and end users. It is important to note that only X-ARF provides alternative parts that address multiple user groups in parallel. On the downside, while not tied to a transport mechanism by the specification, X-ARF reports are usually transmitted via email, which might not be available in every deployment scenario.
- Being equivalent to the machine-readable part of X-ARF, JSON is almost as versatile as the former. It only lacks the support for the end user notification use case. On the plus side, it supports the use case of submitting raw sensor data. It is especially suitable for fully automatic workflows where the human-readable part of X-ARF would not be processed. JSON is readily transmitted via common transport mechanisms.
- IODEF is specialised to exchange data among CSIRTs. It offers more features to express expectations and restrictions on the data usage compared to other incident reporting formats such as X-ARF. The format could be used on a bilateral basis to exchange data with CSIRTs.
- STIX/TAXII is very complex raising the bar for its usage. However, STIX provides good means to characterise and model threats. For example, STIX can be used to model the relationship between attackers, their methods and strategies, and observed incidents. This can, for example, be used to analyse botnets and their characteristics.

6. Data Formats Used in ACDC

While the initial proposal for the CCH was to accept arbitrary data on input and provide that on output, it turned out that a slightly more formalised approach is necessary. With a growing number of partners submitting data into the CCH, it becomes more and more difficult to make an attempt at harmonising the usage of different data fields as well as to handle the different representations of data in the CCH and drive automatic processes with the data available. Not having standards on the representation of data also introduces a level of ambiguity on the side of the data providers in deciding what information to send to the CCH in the first place.

Addressing these problems requires a decision on specific data formats to use for the use cases identified above. Comparing the list of compatible formats for the use cases suggests the following grouping to limit the number of data formats inside the project:

- **Submission and notification of data:** The two submission and one notification use cases mostly share the compatible formats: JSON, data specific (Sflow, hpfeeds), and IDMEF. To lower the amount of work necessary to connect external workflows to ACDC, JSON is selected as a light-weight and flexible format. This does not preclude the adoption of further data formats by the CCH, but it defines the basis that each data format has to support.
- **Aggregated data:** This type of data can also be exchanged with JSON, but is listed separately due to the difference in granularity and properties of the data.
- **End customer notification:** Due to the plain text explanation being necessary to distribute information to end customers, X-ARF is the only format supporting this use case.

Locating these groups in the architecture of the ACDC solution yields Figure 6.

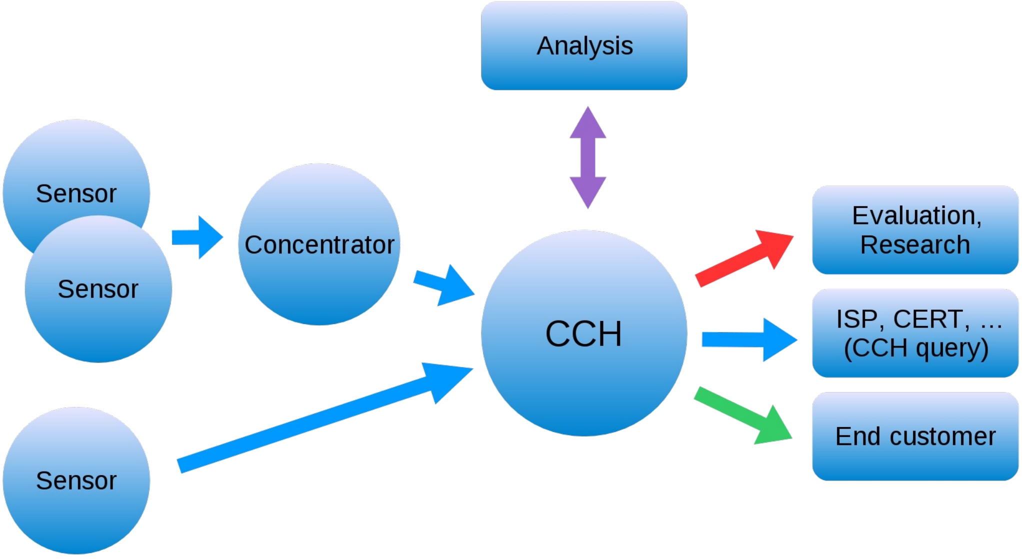


Figure 6: Data format usage in the ACDC solution

The data format usage is shown by the colour of the arrows connecting the individual components using the same colour coding as in the list above. The purple arrow between the CCH and the analysis denotes the possibility to use either the format for submission and notification or for aggregated data in this data transfer.

This division of data formats is a logical one based on the common use cases. It does not imply a mandatory separation in roles nor a physical separation between the contributing parties. An advanced sensor system may for example provide different observations relating to the same attack using the submission and notification format while also providing the correlation between the single events using the format for aggregated data. Thus even complex attacks can be submitted to the CCH by a single sensor.

6.1. Data Formats and Requirements

While some requirements identified in Section 3 are automatically supported by the choice of JSON and X-ARF as the main formats, others have to be satisfied by the actual usage of the formats.

With regards to JSON as the format for submission, notification, and aggregation, this pertains to the following requirements identified for the different use cases in Section 5.1.5:

- **Formal specification to check correctness**

  While the specification of JSON itself is readily available[1], this requirement calls for a specification of the custom data format itself as well. With JSON schema[2], JSON documents can be used to describe a formal specification for custom formats. The following sections develop such a custom format together with JSON schema descriptions, which can be found in the appendices.
- **Security requirements (sensor and recipient authentication)**

  Since the communication with the CCH builds upon SSL together with API key tokens that function as passwords, these requirements are addressed by the use of an appropriate communication protocol.
- **Capability to anonymize**

  The custom data format specified below provides for anonymized or pseudonymized data in its fields together with an annotation to mark the de-identified fields as such.

  While the ACDC workflows support sharing of binary data to some extent, anonymization support on these is not provided by ACDC. The corresponding report categories (e.g., eu.acdc.malware) are only meant for sharing binary data that does not include personal information.

  Sharing personal data in the additional_data fields described below is subject to the applicable ACDC terms of use, especially the CCH Terms of Use available from the ACDC Community Portal. The ACDC data formats have no capability to support or annotate anonymized exports of these fields.
- **Support of the relevant information**

  The specification of the custom data format builds on an analysis of the available data sources and the information provided by them. The necessary and available information on the different kinds of reports are included in the individual report categories while the specification allows providing additional contextual information if available.
- **Support of access control**

  Access control mechanisms are addressed on the level of API keys by the communication protocol. The API key management is provided by the combination of the CCH and the Community Portal. In addition, the data format defined below supports anonymization and pseudonymization on a subset of the fields, facilitating fine-grained access control mechanisms on the level of individual fields.
- **Public specification**

  Besides the public specification of JSON, the schemata of the custom format can be obtained from the Community Portal without registration[3].

With regards to X-ARF as the end customer notification format, this pertains to the following requirements:

- **Human readable**

  The X-ARF reports defined below include a human readable part with the human readable information from the JSON reports and links to further information about ACDC itself.
- **Public specification**

  X-ARF uses JSON schema for the specification of custom reports. The X-ARF schemata used in ACDC are available from the ACDC websites without registration and referenced in every end customer notification.

[1] http://www.ecma-international.org/publications/standards/Ecma-404.htm

[2] http://json-schema.org

[3] https://communityportal.acdc-project.eu/cch-schemata

7. Data Format for Submission and Notification

This section introduces project-wide schemata for data, standardising the selection and usage of different types of data and simplifying the submission and retrieval of data when directly talking to the CCH. Predefined, available schemata are the foundation of reliable automatic processes handling data obtained from the CCH and thus promoting widespread adoption of ACDC. This explicitly excludes information exchanges between different ACDC tools via the CCH that are used to organise the distribution of work between these tools or implement workflows as for example described in [D1.4.2]. These are detailed in the specification deliverables of the corresponding ACDC tool groups in WP1.

This section describes the approach to data schemata used in the ACDC project. It gives an example for a minimal set of data fields that each report has to contain and lists a set of fields, their names, and semantics to be used in schemas in general. This is followed by the current list of schemas used in the ACDC project, which is based on the data the ACDC tools can provide.

The reports are designed to represent the full information known to the CCH about the observation at hand. Updates to a report are possible using the report's ID over the CCH API. This way for example the end of an attack can be signified by an update of the report specifying the duration of the attack.

The analysis is limited to JSON. If other container formats will be supported by the CCH in the future, they SHOULD be used analogously to support a coherent processing of data.

7.1. Minimal Dataset

The minimal dataset serves two purposes: as an example for schemata used in the ACDC project and as a minimal guarantee of information that will be present in every report submitted to and obtained from the CCH. Thus, it is serving as a minimal requirement of information a data source has to provide in a report to the CCH as well as guaranteeing a data consumer the minimal set of information each report queried from the CCH will provide.

The selection of fields in the minimal dataset is driven by the aim of the project to mitigate botnets. It contains the fields that enable the consumer of the information to act on it as well as the fields that facilitate the technical implementation of the project. The high level description of ACDC's minimal dataset is contained in the following table.

ACDC Minimal dataset		
This is the minimal schema a datum submitted to or received from ACDC's CCH must conform to.		
Required fields		
report_category	string	The category of the report. This links the report to one of ACDC's schemata. A report category has the format 'eu.acdc.<identifier>'.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the reported observation took place. This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.
source_key	string enum: ip, malware, subject, uri	The type of the reported object.
source_value	string	The identifier of the reported object like its IP address or URI.

confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 1	The version number of the data format used for the report.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the observation is attributed to. This can for example be the botnet a malware joins, the botnet that sends a spam campaign, or the botnet a bot belongs to.

The following code translates this minimal dataset to JSON Schema. Every report has to provide the fields listed under “properties” and satisfy the requirements given for the fields. The schema says nothing about further fields thus a report for a more complex category is still conforming to this minimal schema as long as it satisfies the requirements on the minimal fields. Additional fields are not restricted and have to be covered by an appropriate schema.

```
{
    "title": "ACDC Minimal dataset",
    "description": "This is the minimal schema a datum submitted to or received
from ACDC's CCH must conform to.",
    "properties": {
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "report_category": {
            "title": "Report category",
            "description": "The category of the report. This links the report to
one of ACDC's schemata. A report category has the format 'eu.acdc.<identifier>'.",
            "type": "string"
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the reported observation took place.
This can for example be when an attack occurred, when a malware hosting was
observed, or when a compromise took place according to log files.",
            "type": "string",
            "format": "date-time"
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object",
            "description": "The type of the reported object.",
            "type": "string",
            "enum": ["ip", "malware", "subject", "uri"]
        },
        "source_value": {
            "title": "Identifier of the reported object",
            "description": "The identifier of the reported object like its IP
address or URI.",
            "type": "string"
        },
        "botnet": {
            "title": "Botnet observation is attributed to",
            "description": "The botnet the observation is attributed to. This can
for example be the botnet a malware joins, the botnet that sends a spam campaign,
or the botnet a bot belongs to.",
            "type": "string"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "minimum": 1
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version"]
}
```

As can be seen each “part” has a short title and a longer description on it. The properties are the individual fields of the report, and each field in turn has a type, which is the basic data type used, and if appropriate some additional constraints on the format or possible values.

The most useful data types to ACDC are:

- string
- integer
- boolean
- array

The most useful additional constraints are:

- “format” with the values
  - uri,
  - date-time – date and time in ISO8601 (i.e., YYYY-MM-DDThh:mm:ssZ in UTC, e.g. 2014-06-17T11:23:00Z for 11:23:00 UTC on the 17th of June 2014),
  - email – an email address,
  - ipv4 – an IPv4 address,
  - ipv6 – an IPv6 address,
- “enum” limiting the possible values for the property (see source_key above), and
- “items”, which itself gives a specification of the items in an array type.

7.2. Fields

With the minimal data set above as a starting point more complex schemata for different types of reports can be defined. To ease the processing of information, ACDC aims for a coherent usage of field names and semantics. The following table lists common field names and their semantics as they SHALL be used in schemata inside ACDC. The fields' titles and descriptions give their general usage and semantics. As a specific schema conveys additional context, the actual title and description in a specific schema MAY be altered to be more suitable.

The focus in the selection of fields for common usage—as well as in the definition of schemata—is put on distributing actionable information over ACDC. Thus, the aim is to standardise fields not for reporting arbitrarily structured sensor data but for reporting preprocessed data on a level of abstraction where the consumer of data from the CCH can easily parse data and drive (automatic) processes with it. This calls for a limited set of fields and in some cases a level of processing on the side of the data provider that goes beyond simply submitting sensor output.

The field type in the following table gives the data type of the field as well as additional constraints on the value. Besides the types and constraints described above, this may contain:

- **object / object(...):** The field is itself structured and contains nested fields. In the second form the nested fields are given in parentheses, for example “object(string, string)” for a structure with two string fields.
- **requires:** If this field is present, the ones listed after “requires” have to be present as well.
- **array(...):** Short notation for an array of given structure, for example “array(string, string)” is an array of two string values.

Field name	Field type	Field title	Field description
additional_data	object	Additional data	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format	string requires: alternate_format_type	Alternate format description of the observation	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	Type of the alternate format	The type of the alternate format description of the observation.
application_protocol	string	Application protocol	The application protocol used for the connection.
bit_rate	integer	Bits per second of traffic	The number of bits per second of traffic transferred.
bot_id	string	Identifier of the bot	The identifier the botnet uses for this bot. Not all botnets have this concept. Since bot IDs are only meaningful in the context of a botnet, a report containing a bot_id should contain a botnet field as well if possible.

Field name	Field type	Field title	Field description
botnet	string	Botnet observation is attributed to	The botnet the observation is attributed to. This can for example be the botnet a malware joins, the botnet that sends a spam campaign, or the botnet a bot belongs to.
c2_ip_v4	string format: ipv4 requires: c2_mode	IPv4 of the C2	The IPv4 of the C2 server.
c2_ip_v6	string format: ipv6 requires: c2_mode	IPv6 of the C2	The IPv6 of the C2 server.
c2_mode	string enum: plain, anon, pseudo	C2 IP mode	The mode of the C2 server IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
c2_port	integer	Port of the C2 connection	The port of the connection to the C2 server.
confidence_level	number minimum: 0.0 maximum: 1.0	Confidence level of the report	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
cpe	string	CPE of the affected platform	The full or partial CPE name binding of the platform affected by the report.
credentials	array items: array(string, string)	Credentials	The credentials used for example in an attack to brute force a login. This is a list of pairs. Each pair consists of a user name and a password as strings.
dst_access_type	string enum: mobile, fixed	Access type of the destination IP	The type of access network used by the destination IP. Mobile signifies address spaces assigned to mobile access technologies like 3G or 4G. Fixed signifies address spaces assigned to fixed access technologies like xDSL or FTTH.
dst_ip_v4	string format: ipv4 requires: dst_mode	Destination IPv4 of the connection	The destination IPv4 of the connection. This is always the remote IP from the perspective of the reported system (i.e., the one identified by source_value). It can for example be the IP of a honeypot that was contacted to infect it.
dst_ip_v6	string format: ipv6 requires: dst_mode	Destination IPv6 of the connection	The destination IPv6 of the connection. This is always the remote IP from the perspective of the reported system (i.e., the one identified by source_value). It can for example be the IP of a honeypot that was contacted to infect it.

Field name	Field type	Field title	Field description
dst_mode	string enum: plain, anon, pseudo	Destination IP mode	The mode of the destination IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
dst_port	integer	Destination port of the connection	The destination port of the connection. This is always the remote port from the perspective of the reported system (i.e., the one identified by source_value). It can for example be the port of a honeypot that was contacted to infect it.
duration	integer minimum: 0	Duration of the observation	The duration of the observation in seconds. This can for example be the duration of DoS attack.
exploits	array items: object(identifier scheme, exploit identifier)	Exploits in the sample	Exploits discovered in the analysed sample. This is an array of objects, each giving an identifier scheme like CVE and an identifier for the actual exploit found.
fast_flux_uri	string format: uri	URI of the fast flux domain	The URI of the fast flux domain connected to this report.
http_request	string	HTTP request	The HTTP request observed. This can be for example a request some attacking machine sent to a sensor or a request a bot sent to a C2 server to query new commands.
ip_protocol_number	integer minimum: 0 maximum: 255	IP protocol number	The IANA assigned decimal internet protocol number of the connection[1].
ip_version	integer enum: 4, 6	IP version number	The IP version of the connection.
mail_body	string	Email body	The body of the email. Varying parts, especially personal information like names or email addresses, must be replaced with the placeholder '{}'.
mail_header	string	Email header	The header of the email.
mime_type	string	MIME type	The MIME type of an object. This can for example be the MIME type of a malware sample.
malicious_uri	string format: uri	URI of malicious content	The URI where the malicious content can be found in the wild like the location of a malware supposed to be downloaded as part of an attack.
packet_rate	integer	Packets per second of traffic	The number of packets per second of traffic transferred.

[1] https://www.iana.org/assignments/protocol-numbers/protocol-numbers.txt

Field name	Field type	Field title	Field description
report_category	string	Report category	The category of the report. This links the report to one of ACDC's schemata. A report category has the format “eu.acdc.<identifier>”.
report_id	string	Report ID	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
report_subcategory	string	Report subcategory	The subcategory of the report. This is used to categorise different types of similar reports that have mostly the same fields. It is defined as an enum in the schema of the report category.
report_type	string	Report type	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
reported_at	string format: date-time	Time of the report's submission	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
sample_b64	string	Source of the sample	The source code of the sample encoded in Base64. Only to be used for data not including personal information.
sample_filename	string	Filename of the sample	The filename used for the sample like the name of an attachment to an email or an upload to a honeypot.
sample_hashes	array items: object(hash function, hash value)	Hashes of the sample	A list of hashes for the sample, each giving a hash function and the corresponding hash value for the sample. This is used as information on a specific sample.
sample_sha256	string	SHA256 of the sample	The SHA256 hash of the sample. This is used to reference a specific sample.
source_key	string enum: botnet, ip, malware, subject, uri	Type of the reported object	The type of the reported object.
source_value	string	Identifier of the reported object	The identifier of the reported object like its IP address or URI.
src_access_type	string enum: mobile, fixed	Access type of the source IP	The type of access network used by the source IP. Mobile signifies address spaces assigned to mobile access technologies like 3G or 4G. Fixed signifies address spaces assigned to fixed access technologies like xDSL or FTTH.

Field name	Field type	Field title	Field description
src_ip_v4	string format: ipv4 requires: src_mode	Source IPv4 of the connection	The source IPv4 of the connection. This is always the IP of the reported system (i.e., the one identified by source_value).
src_ip_v4s	array items: object(timestamp, src_ip_v4)	Source IPv4s and timestamps	A list of IPv4 source addresses together with timestamps associated to the observation. This can for example be the IPs a fast flux domain resolves to.
src_ip_v6	string format: ipv6 requires: src_mode	Source IPv6 of the connection	The source IPv6 of the connection. This is always the IP of the reported system (i.e., the one identified by source_value).
src_ip_v6s	array items: object(timestamp, src_ip_v6)	Source IPv6s and timestamps	A list of IPv6 source addresses together with timestamps associated to the observation. This can for example be the IPs a fast flux domain resolves to.
src_mode	string enum: plain, anon, pseudo	Source IP mode	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
src_port	integer	Source port of the connection	The source port of the connection. This is always the port on the reported system (i.e., the one identified by source_value).
subject_text	string	Subject of the email	The subject text of the email. Varying parts, especially personal information like names or email addresses, must be replaced with the placeholder '{}'.
timestamp	string format: date-time	Time of the reported observation	The timestamp when the reported observation took place. This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.
version	integer minimum: 1	Version of the format	The version number of the data format used for the report.
vulnerabilities	array items: object(identifier scheme, vulnerability identifier)	Vulnerabilities in a system	Vulnerabilities discovered in an analysed system. This is an array of objects, each giving an identifier scheme like CWE and an identifier for the actual vulnerability found.

7.2.1. Additional Data

To provide further context to a report and submit additional information that the standardised fields and schemata do not cover, every report MAY contain an object additional_data. The structure of this object is in the data providers discretion. If it contains fields with the above semantics, then these SHOULD be named accordingly. To support the consumer of the data to process the data, the provider SHOULD either include a string field describing the content and semantics of the object or link to a schema for this purpose.

A report SHALL NOT contain additional information in fields outside of this structure.

7.2.2. Alternate Format

To submit complex structured formats to the CCH but still simplify the data processing for consumers, a limited normalisation of these reports is necessary. For this the standard fields required for an appropriate report SHALL be transcribed from the complex format into a JSON report. The complete structured data MAY nevertheless be added to the report with the alternate_format and alternate_format_type fields to provide complete data to consumers that are able to process the structured format.

Robust, automatic processing of the alternate format is only possible with a reliable way to specify the alternate data format. Therefore, alternate_format_type allows a limited set of defined values for the allowed alternate data formats, namely:

- **CybOX:** The Cyber Observable eXpression language[1], a structured language for cyber observables.
- **hpfeeds:** The message format of hpfeeds[2], a lightweight authenticated publish-subscribe protocol.
- **IDMEF:** The Intrusion Detection Message Exchange Format [RFC 4765], a data format defined for sharing information between intrusion detection and response systems and management systems.
- **IODEF:** The Incident Object Description Exchange Format [RFC 5070], a data format for sharing information commonly exchanged by Computer Security Incident Response Teams about computer security incidents.
- **IPFIX:** The IP Flow Information Export file format [RFC 5655], a file format for storage of flow data based on the IPFIX protocol.
- **NetFlow v9:** The data export format for version 9 of Cisco System’s NetFlow services [RFC 3954].
- **OpenIOC:** The OpenIOC framework for sharing threat intelligence[3].
- **sFlow:** The sFlow format to describe network traffic data [RFC 3176].
- **STIX:** The Structured Threat Information eXpression language[4], a structured language for cyberthreat intelligence information.

New values for the alternate format can be introduced with new versions of the report schemata.

7.2.3. ASN

There is no field for the ASN (Autonomous System Number) an IP address in a report belongs to. The ASN is used in ACDC only as a filter on the IP addresses of reports to manage access to these reports via queries by ISPs or CERTs for example. Thus, lookup and handling of the ASN is done completely in the CCH.

7.2.4. Botnet

This field allows connecting any report to a specific botnet if that information is available. Attributing information to a botnet this way not only provides insight on the size of a botnet but also on its type and frequency of activity.

Making the connection between an observation and the responsible botnet, however, is often difficult if at all possible. From the perspective of a honeypot for example, there is usually no information available to attribute an attacker to a specific botnet in a report. The attacking machine might not be part of a botnet at all. This information may be inferred by correlating separate reports associating the attacking system with a botnet, but the system might be part of multiple botnets or might be located behind a NAT or proxy questioning the validity of a connection of the reports based on the IP address alone.

[1] https://cybox.mitre.org
[2] https://github.com/rep/hpfeeds
[3] http://www.openioc.org
[4] https://stix.mitre.org

Including this field in the reports only offers an opportunity and provides technical support but is all that can be done on a data format level. To get the most value out of the information, there needs to be a common understanding and terminology applied in using this field. As the botnet landscape develops over time so too the values for this field have to change. This can ultimately only be provided by the ACDC Community Portal where changes in the botnet landscape can be observed and properly reflected in the usage of the fields and formats.

7.2.5. c2_mode, dst_mode, src_mode

These fields SHALL be used to support working with reports containing anonymized and pseudonymized IP addresses. They denote the kind of preprocessing applied to the C2, destination, or source IP address, respectively. This way a data receiver knows what properties to expect from the supplied IP addresses. He knows that he'll be able to find them in his log files if they are not preprocessed, to make some correlation on them if they are pseudonymized, or that he cannot infer further information from them if anonymized.

7.2.6. Confidence Level

The confidence level in a report SHALL be used to convey the confidence put into the data by the sender of the data. It is a numeric value between 0.0 and 1.0.

The value used in submitting a report to the CCH SHALL be an estimate for the precision of the submitting sensor, that is an estimate of the probability that the report is indeed accurate and not a false positive.

If an attacker completes a TCP three way handshake with a honeypot and tries to guess passwords for SSH accounts, the confidence_level can be set to 1.0. There is a close to zero probability of the IP address being spoofed because of the completed hand shake. There are no legitimate services on the honeypot, indicating a low probability of a legitimate user accidentally accessing the honeypot and entering their credentials. The latter probability can be further reduced by sending a report only for multiple login retries with passwords indicating brute force or dictionary attacks. In combination, the probability of this event being registered on the sensor but the detected IP not performing a password guessing attack on the sensor is close to zero, the confidence_level is 1.0.

In contrast, detecting a UDP flood with DNS queries on a sensor close to a DNS server cannot result in a DNS flood attack report by the source of the packets with a high confidence_level. Since the packets are sent via UDP, there is no guarantee that the packets actually originate from the source IP contained in the packets. DNS queries eliciting large responses from a DNS server are often used with spoofed source IPs to perform DNS reflection attacks on said source IP. Here spoofed queries are sent to multiple DNS servers that in turn flood the victim at the source IP of the queries with the answers to said queries. In result, while the source may try to flood the DNS server with a high number of queries, the source may itself be the target of the attack. The confidence_level on this detection cannot be high.

Determining the precision of a sensor can be difficult. Firstly, it requires an analysis of the true and false positive rate of the sensor, that is what proportion of actual attacks are correctly classified as attacks and what proportion of benign traffic is classified wrongly as an attack. Secondly, it requires an analysis of the traffic the sensor is exposed to. What proportion of traffic is actually an attack and what proportion is benign. In combination, the proportion of actual attacks classified as attacks to all traffic classified as attacks yields the precision of the sensor. Since especially determining the ratio of benign traffic versus attacks can be difficult and might change dynamically, only an estimate of the precision is required in the reports.

The rationale for introducing this field in this way is the following. On the one hand, it enables the handling of reports with differing precision while maintaining the balance between data privacy and security. Reports of benign traffic should not enter the CCH since ACDC lacks the legitimacy of processing these. Introducing the confidence_level as described above allows to reject reports with a low confidence_level as these have a high probability of being false positives. Reports with a higher confidence_level can be suppressed from being forwarded to ISPs and CERTs until correlation provides corroborating information from other reports in the CCH to further reduce the risk of false positives being sent out to end customers. On the other hand, the confidence_level provides the data receiver with important information in assessing and prioritising the information.

Nevertheless, the confidence level is still an assessment by the data provider. Inaccurate statements of the confidence level, either accidentally or on purpose, cannot be addresses on a data format level. One possible way to deal with this would be introducing a feedback channel from the party receiving a report from the CCH to mark the report as a false positive. This could be used in the CCH to automatically rate reports based on the provider stated confidence level as well as the history of feedback from receivers of the provider's reports.

7.2.7. CPE

Common Platform Enumeration (CPE) is a standardised method to describe and identify applications, operating systems, and hardware devices. CPE defines a machine-readable naming scheme to encode this information in an URI or a colon separated string [NISTIR 7695]. This is used in ACDC's reports to specify the platform that a malware is running on.

If there is a malicious app running on an Android 5.0 system, the corresponding CPE binding would be one of the following two; the first one is using the URI syntax, the second one the colon separated form.

```
cpe:/o:google:android:5.0
cpe:2.3:o:google:android:5.0:*:*:*:*:*:*:*
```

If the exact version of the platform is not available, or not relevant for the malware, it can be left out of the CPE binding to yield one of

```
cpe:/o:google:android
cpe:2.3:o:google:android:*:*:*:*:*:*:*:*
```

7.2.8. Subject Text

While maybe not capturing the most general definition of a spam campaign, the subject text seems to be the most appropriate value to automatically identify a spam campaign. Personalised subjects are taken care of by replacing the varying parts with a placehoder "{}". This removes the personal information from the report while at the same time unifying the subjects in the campaign, making correlation of the emails easier.

Other values introduce more variation across the emails sent in a campaign – like the email body – or introduce more personal information in the identifier – like the email body as well. Also, introducing a different identifier has the problem of how to communicate it to data providers and how to ensure that they use it appropriately in their reports to the CCH.

A possibly minor point raised by the subject is how to deal with encodings of the subject that might differ with the MTAs processing the email and thus introducing some variation in a campaign as well.

For possible improvements in using the subject text refer to Section 7.4.

7.3. Report Schemata

As mentioned for the fields before, the focus in the definition of schemata is on distributing actionable information over ACDC by reporting preprocessed data on a level of abstraction where the consumer of data from the CCH can easily parse data and drive (automatic) processes with it.

The list of required fields for each report is limited to the necessary information to process the reports in ACDC to support the balance of privacy and security.

The complete JSON schemata can be found in Appendix C. Appendix B provides a snapshot of the expected schemata usage as well as potential extensions at the time of schema introduction into the project.

7.3.1. Report Categories

In ACDC we describe different kinds of suspicious or malicious behaviour that we have to map to different report categories to facilitate sharing and acting on them. These categories are devised to describe the malicious behaviour instead of the output of some sensor. This enables the receiver of some report from the CCH to directly act on the information without having to interpret the report in the context of the providing sensor.

Report category names use the reverse domain name notation[1]. All categories start with "eu.acdc.".

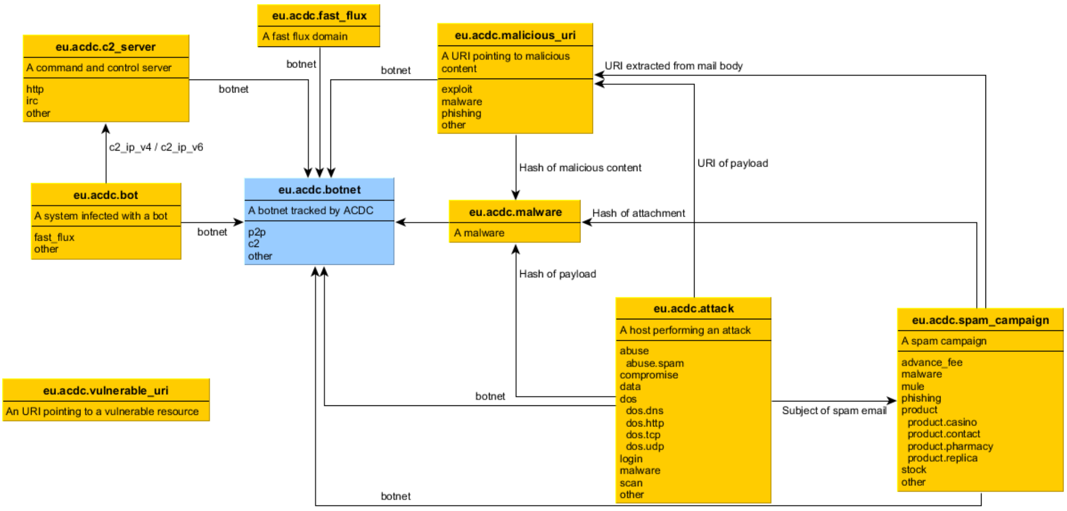


Figure 7: Report categories in ACDC

Figure 7 gives an overview of all report categories and subcategories defined in ACDC. Reports can refer to reports of other categories using their source_value as an identifier. A spam email captured as an eu.acdc.attack can for example refer to a spam campaign via the subject of the email. These connections are depicted in the figure with arrows from the referring report category to the referred report category. The arrows are annotated with the type of information in the referring report that establishes the connection.

7.3.2. Attack (eu.acdc.attack)

The attack report is used to submit a host attacking another one. The associated report_category is *eu.acdc.attack*.

With the sample_uri that can be provided with this report, multiple systems can be part of the attack. This field is however only meant to give context to the attack and not to report the connected malware hosting itself. To report the malware hosting, a separate eu.acdc.malicious_uri report needs to be submitted to the CCH.

Subcategory

An attack report can fall into one of the following subcategories:

- **abuse:** the host tries to abuse a service offered by the attacked machine to attack somebody else; this may be further detailed with the subcategory
  - **abuse.spam:** the host sends a spam email
- **compromise:** the host tries to compromise the target (e.g., compromising a web server with a remote file inclusion)
- **data:** the host tries to exfiltrate data from the target (e.g., obtaining data via SQL injection)
- **dos:** the host performs a DoS or participates in a DDoS on the target; this may be further detailed by using one of the following subcategories:
  - **dos.dns:** the host performs a DNS flood
  - **dos.http:** the host performs an HTTP flood
  - **dos.tcp:** the host performs a TCP flood
  - **dos.udp:** the host performs a UDP flood
- **login:** the host tries to guess a login to a service on the target

[1] https://en.wikipedia.org/wiki/Reverse_domain_name_notation

- **malware:** the host tries to infect the target with some malware (e.g., infecting a client with a worm)
- **scan:** the host scans the target (e.g., to find open ports or vulnerable services)
- **other**

Schema

Attack – eu.acdc.attack		
A host performing an attack.		
Required fields		
report_category	string enum: eu.acdc.attack	The category of the report: an attack on a system.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the attack took place.
source_key	string enum: ip	The type of the reported object: an IP.
source_value	string	The IP of the system performing the attack.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: abuse, abuse.spam, compromise, data, dos, dos.dns, dos.http, dos.tcp, dos.udp, login, malware, scan, other	The type of attack performed.
ip_protocol_number	integer minimum: 0 maximum: 255	The IANA assigned decimal internet protocol number of the attack connection.
ip_version	integer enum: 4, 6	The IP version of the attack connection.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration of the attack in seconds.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the attack can be attributed to. This references a separate eu.acdc.botnet report.

additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
src_ip_v4	string format: ipv4	The source IPv4 of the attack connection. This is always the IP of the attacking system (i.e., the one identified by source_value). This field equals source_value.
src_ip_v6	string format: ipv6	The source IPv6 of the attack connection. This is always the IP of the attacking system (i.e., the one identified by source_value). This field equals source_value.
src_mode	string enum: plain, anon, pseudo	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
dst_ip_v4	string format: ipv4	The destination IPv4 of the attack connection. This is always the IP of the attacked system.
dst_ip_v6	string format: ipv6	The destination IPv6 of the attack connection. This is always the IP of the attacked system.
dst_mode	string enum: plain, anon, pseudo	The mode of the destination IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
src_port	integer	The source port of the attack connection. This is always the port on the attacking system (i.e., the one identified by source_value).
dst_port	integer	The destination port of the attack connection. This is always the port on the attacked system.
application_protocol	string	The application protocol used for the connection.
sample_filename	string	The filename used for the payload that the attack tried to install or run on the attacked system. This should only be used if the payload is uploaded to the attacked system directly. Otherwise, malicious_uri should be used to link this report to an eu.acdc.malicious_uri report that in turn contains the SHA256 hash.
sample_sha256	string	The SHA256 hash of the payload that the attack tried to install or run on the attacked system. This should only be used if the payload is uploaded to the attacked system directly. Otherwise, malicious_uri should be used to link this report to an eu.acdc.malicious_uri report that in turn contains the SHA256 hash.

malicious_uri	string format: uri	The URI of the payload in the wild that the attack tried to install or run on the attacked system. This can for example be the location of a malware offered as a download or a web shell offered as a remote include during an attack.
subject_text	string	The subject of an email sent in a report of subcategory abuse.spam. Varying parts, especially personal information like names or email addresses, must be replaced with the placeholder '{}'. This references a separate eu.acdc.spam_campaign report.
mail_header	string	The header of the email sent in a report of subcategory abuse.spam.
bit_rate	integer	The number of bits per second of traffic transferred, for example in a DoS attack.
packet_rate	integer	The number of packets per second of traffic transferred, for example in a DoS attack.

Dependencies
A report has to provide either src_ip_v4 or src_ip_v6 depending on the ip_version being 4 or 6, respectively. For each provided IP address the corresponding mode has to be specified. If the report contains an alternate_format, it has to specify the alternate_format_type as well.

Example

A denial of service attack might be submitted to the CCH in the following manner.

```
{
       "report_category": "eu.acdc.attack",
       "report_subcategory": "dos",
       "report_type": "TCP SYN Flood",
       "timestamp": "2014-06-15T15:47:12Z",
       "source_key": "ip",
       "source_value": "192.0.2.14",
       "ip_protocol_number": 6,
       "ip_version": 4,
       "src_ip_v4": "192.0.2.14",
       "src_mode": "plain",
       "dst_ip_v4": "198.51.100.111",
       "dst_mode": "anon",
       "dst_port": 80,
       "packet_rate": 200000,
       "confidence_level": 1.0,
       "version": 2
}
```

7.3.3. Bot (eu.acdc.bot)

The bot report is used to report a bot infection of an IP to the CCH. This report is not connected to an actual attack, which would indicate an eu.acdc.attack report, but is triggered for example by a connection to a C2 sinkhole.

Subcategory

A bot report can fall into one of the following subcategories.

- **fast_flux:** a fast flux domain resolves to this bot's IP address
- **other**

Schema

Bot – eu.acdc.bot
A system infected with a bot.

Required fields		
report_category	string enum: eu.acdc.bot	The category of the report: a bot infection.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the bot infection was observed.
source_key	string enum: ip	The type of the reported object: an IP address.
source_value	string	The IP address of the infected system.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: fast_flux, other	The type of bot.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration in seconds during which the bot infection was observed. This can be the timespan during which connections to the C2 server were observed.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the bot is attributed to.
bot_id	string	The identifier the botnet uses for this bot. Not all botnets have this concept. Since bot IDs are only meaningful in the context of a botnet, a report containing a bot_id should contain a botnet field as well if possible.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
ip_version	integer enum: 4, 6	The IP version of the C2 connection.

ip_protocol_number	integer minimum: 0 maximum: 255	The IANA assigned decimal internet protocol number of the C2 connection.
src_ip_v4	string format: ipv4	The source IPv4 of the bot infected system. This field equals source_value.
src_ip_v6	string format: ipv6	The source IPv6 of the bot infected system. This field equals source_value.
src_mode	string enum: plain, anon, pseudo	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
c2_ip_v4	string format: ipv4	The IPv4 of the C2 server.
c2_ip_v6	string format: ipv6	The IPv6 of the C2 server.
c2_mode	string enum: plain, anon, pseudo	The mode of the C2 server IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
src_port	integer	The source port of the connection from the bot to the C2 server. This is always the port on the bot infected system.
c2_port	integer	The port of the C2 connection on the C2 server.
sample_sha256	string	The SHA256 hash of the malware the system is infected with. This references a separate eu.acdc.malware report.
fast_flux_uri	string format: uri	The URI of the fast flux domain resolving to this bot.
Dependencies		
A report has to provide either src_ip_v4 or src_ip_v6 depending on the ip_version being 4 or 6, respectively. For each provided IP address the corresponding mode has to be specified. If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A bot infection detected on a C2 sinkhole can be sent to the CCH in the following way.

```
{
    "report_category": "eu.acdc.bot",
    "report_type": "Connection to Zeus C2",
    "timestamp": "2014-06-15T15:47:12Z",
    "source_key": "ip",
    "source_value": "121.154.32.23",
    "reported at": "2014-06-22T15:47:12Z",
    "confidence_level": 1.0,
    "version": 2,
    "report_subcategory": "other",
    "ip_version": 4,
    "src_ip_v4": "121.154.32.23",
    "src_mode": "plain",
    "c2_ip_v4": "10.1.3.12",
    "c2_mode": "anon"
}
```

7.3.4. C2 Server (eu.acdc.c2_server)

The C2 server report is used to submit a C2 server (Command and Control server) to the CCH. The associated report_category is eu.acdc.c2_server.

Subcategory

A C2 server report can fall into one of the following subcategories:

- **http:** a C2 server using an HTTP based control channel
- **irc:** a C2 server using an IRC based control channel
- **other**

Schema

C2 Server – eu.acdc.c2_server		
A command and control server.		
Required fields		
report_category	string enum: eu.acdc.c2_server	The category of the report: a C2 server.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the C2 server was observed.
source_key	string enum: ip	The type of the reported object: an IP address.
source_value	string	The IP address of the C2 server.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: http, irc, other	The control channel used by the C2.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration in seconds during which the C2 server was observed. This can be the timespan during which connections to the C2 server were successful.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the C2 server is attributed to.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.

alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
ip_version	integer enum: 4, 6	The IP version of the C2 server's IP address.
ip_protocol_number	integer minimum: 0 maximum: 255	The IANA assigned decimal internet protocol number used for C2 connections.
c2_ip_v4	string format: ipv4	The IPv4 of the C2 server.
c2_ip_v6	string format: ipv6	The IPv6 of the C2 server.
c2_mode	string enum: plain, anon, pseudo	The mode of the C2 server IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
c2_port	integer	The port of C2 connections on the C2 server.
Dependencies		
A report has to provide either c2_ip_v4 or c2_ip_v6 depending on the ip_version being 4 or 6, respectively. For each provided IP address the corresponding mode has to be specified. If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A C2 server can be submitted to the CCH in the following manner.

```
{
    "report_category": "eu.acdc.c2_server",
    "report_type": "Zeus C2",
    "timestamp": "2014-06-15T15:47:12Z",
    "source_key": "ip",
    "source_value": "121.154.32.23",
    "reported at": "2014-06-22T15:47:12Z",
    "confidence_level": 1.0,
    "version": 2,
    "report_subcategory": "http",
    "ip_version": 4,
    "c2_ip_v4": "121.154.32.23",
    "c2_mode": "plain"
}
```

7.3.5. Fast Flux Domain (eu.acdc.fast_flux)

The fast flux report is used to submit a fast flux domain to the CCH. The associated report_category is *eu.acdc.fast_flux*. Bots that the domain resolved to are reported with separate eu.acdc.bot reports.

Subcategory

There are no subcategories for the eu.acdc.fast_flux report.

Schema

Fast Flux Domain – eu.acdc.fast_flux		
A fast flux domain.		
Required fields		
report_category	string enum: eu.acdc.fast_flux	The category of the report: a fast flux domain.

report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the fast flux domain was first observed. If the report contains IPs, this is typically the earliest timestamp of the IPs that the domain resolves to.
source_key	string enum: uri	The type of the reported object: a domain URI.
source_value	string format: uri	The fast flux domain URI.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration of the observation in seconds. If the report contains IPs, this is typically the difference between the earliest and the latest timestamp of the IPs that the domain resolves to.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the fast flux domain can be attributed to. This references a separate eu.acdc.botnet report.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
Dependencies		
If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A fast flux domain could be reported to the CCH in the following manner.

```
{
        "report_category": "eu.acdc.fast_flux",
        "report_type": "Fast flux domain of the XYZ bot",
        "timestamp": "2014-06-15T15:47:12Z",
        "source_key": "uri",
        "source_value": "dns:fast_flux.example.com",
        "confidence_level": 1.0,
        "version": 2
}
```

7.3.6. Malicious URI (eu.acdc.malicious_uri)

The report for a malicious URI is used to submit a URI that points to malicious content. The associated report_category is *eu.acdc.malicious_uri*.

Subcategory

A malicious website report can fall into one of the following subcategories:

- **exploit:** the website hosts an exploit or exploit kit
- **malware:** the website hosts a malware download
- **phishing:** the website hosts a phishing site
- **other**

Schema

Malicious URI – eu.acdc.malicious_uri		
A URI pointing to malicious content.		
Required fields		
report_category	string enum: eu.acdc.malicious_uri	The category of the report: a malicious URI.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the malicious URI was observed.
source_key	string enum: uri	The type of the reported object: a URI.
source_value	string format: uri	The URI to the malicious content.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: exploit, malware, phishing, other	The type of the malicious content at the URI.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration of the observation in seconds. If the report contains IPs, this is typically the difference between the earliest and the latest timestamp of the IPs that the domain resolves to.

reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the malicious URI can be attributed to. This references a separate eu.acdc.botnet report.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
ip_version	integer enum: 4, 6	The IP version of the IP address belonging to the malicious URI.
src_ip_v4	string format: ipv4	The source IPv4 associated with the malicious URI.
src_ip_v6	string format: ipv6	The source IPv6 associated with the malicious URI.
src_mode	string enum: plain, anon, pseudo	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
sample_filename	string	The file name of the malicious content if applicable.
sample_sha256	string	The SHA256 hash of the malicious content if applicable.
exploits	array items: object(identifier scheme, exploit identifier)	Exploits discovered in the analysed URI. This is an array of objects, each giving an identifier scheme like CVE and an identifier for the actual exploit found.
Dependencies		
If the report provides an ip_version, it has to provide either src_ip_v4 or src_ip_v6 depending on the ip_version being 4 or 6, respectively. For each provided IP address the corresponding mode has to be specified. If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A website hosting trying to exploit a vulnerability in Firefox might be submitted to the CCH in the following manner.

```
{
    "report_category": "eu.acdc.malicious_uri",
    "report_subcategory": "exploit",
    "report_type": "Exploiting Firefox",
    "timestamp": "2014-03-12T14:11:02Z",
    "source_key": "uri",
    "source_value": "http://example.dr/malicious.html",
    "confidence_level": 1.0,
    "version": 2,
    "exploits": [{
        "type": "cve",
        "value": "CVE-2014-1555"
    }]
}
```

7.3.7. *Malware Sample (eu.acdc.malware)*

This report is used to submit a malware sample to the CCH. The associated report_category is *eu.acdc.malware.*

Subcategory

There are no subcategories for the eu.acdc.malware report.

Schema

Malware Sample – eu.acdc.malware		
A sample of a malware.		
Required fields		
report_category	string enum: eu.acdc.malware	The category of the report: a malware sample.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the sample was obtained.
source_key	string enum: malware	The type of the reported object: a malware sample.
source_value	string	The SHA256 hash of the malware sample.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the sample is attributed to. This reference a separate eu.acdc.botnet report.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.

alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
sample_b64	string	The source code of the sample encoded in Base64. Only to be used for data not including personal information.
mime_type	string	The MIME type of the sample.
cpe	string	The full or partial CPE name binding of the platform affected by the malware.
sample_hashes	array items: object(hash function, hash value)	An array of objects containing hashes for the sample. Each item gives a hash function and the corresponding hash value.
exploits	array items: object(identifier scheme, exploit identifier)	Exploits discovered in the analysed sample. This is an array of objects, each giving an identifier scheme like CVE and an identifier for the actual exploit found.
Dependencies		
If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A malware captured by a honeypot might be submitted to the CCH in the following manner. The sample_b64 is truncated here but SHALL be complete when submitting a sample.

```
{
       "report_category": "eu.acdc.malware",
       "report_type": "Bot from honeypot capture",
       "timestamp": "2014-06-15T15:47:12Z",
       "source_key": "malware",
       "source_value":
"933ff380eb1448c5c583796fdc6ce842eec14a23b686fff4672c85a7ee9d7d0a",
       "cpe": "cpe:/o:microsoft:windows_8.1",
       "sample_b64":
"ewogICAgInRpdGxlIjogIk1hbHdhcmUgU2FtcGxlIiwgCiAgICAiZGVzY3JpcHRpb24iOiAK...",
       "confidence_level": 1.0,
       "version": 2
}
```

7.3.8. Spam Campaign (eu.acdc.spam_campaign)

This report is used to submit spam campaigns to the CCH. Campaigns are identified via their email subject. The report category is *eu.acdc.spam_campaign*.

Subcategory

A spam campaign report can fall into one of the following subcategories:

- **advance_fee:** the campaign proposes some benefit after paying an initial fee, for example 419 scams[1]
- **malware:** the spam messages contain or link to some malware
- **mule:** the campaign tries to recruit mules to forward goods or money
- **phishing:** the campaign tries to capture personal information like passwords or bank account details
- **product:** the campaign promotes websites selling certain products or services, the type of product can be detailed with the subcategories:
  - **product.casino:** casino websites
  - **proudct.contact:** contact information on prospective customers like lists of email addresses
  - **product.pharmacy:** pharmaceutical products

[1] https://en.wikipedia.org/wiki/419_scams

  - **product.replica:** replica products
- **stock:** the campaign promotes a stock to profit via pump and dump schemes[1]
- **other**

Some spam campaigns might be regarded as to fall into multiple of these subcategories. In this case the campaign SHOULD be attributed to the most significant subcategory.

Schema

Spam Campaign – eu.acdc.spam_campaign		
A spam campaign.		
Required fields		
report_category	string enum: eu.acdc.spam_campaign	The category of the report: a spam campaign.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the spam campaign was first observed.
source_key	string enum: subject	The type of the reported object: an email subject.
source_value	string	The common subject of the spam campaign. Varying parts, especially personal information like names or email addresses, must be replaced with the placeholder '{}'.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: advance_fee, malware, mule, phishing, product, product.casino, product.contact, product.pharmacy, product.replica, stock, other	The type of spam messages sent.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration during which the sample was observed.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the spam campaign is attributed to. This reference a separate eu.acdc.botnet report.

[1] https://en.wikipedia.org/wiki/Pump_and_dump

additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
mail_body	string	The body of the email. Varying parts, especially personal information like names or email addresses, must be replaced with the placeholder ‘{}’.
sample_filename	string	The file name of the malicious attachment used in this campaign.
sample_sha256	string	The SHA256 hash of the malware distributed with this campaign. This references a separate eu.acdc.malware report.
malicious_uri	string format: uri	The URI advertised with this campaign. This references a separate eu.acdc.malicious_uri report.
Dependencies		
A report has to provide at least one of sample_sha256 and malicious_uri. If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A spam campaign distributing malware could be sent to the CCH in the following manner.

```
{
       "report_category": "eu.acdc.spam_campaign",
       "report_subcategory": "malware"
       "report_type": " Campaign  occurred..",
       "source_key": "subject",
       "source_value": "Teik it or leave it",
       "timestamp": "2012-01-10 16:45",
       "sample_sha256":
"4d410bc194c5fbdf20d15fae6c6bd807f66b56c36afe3a3def37e4369193ed2e",
       "confidence_level": 1.0,
       "version": 2
}
```

7.3.9. *Vulnerable URI (eu.acdc.vulnerable_uri)*

The report for a vulnerable URI is used to submit a URI that points to a vulnerable system. The associated report_category is *eu.acdc.vulnerable_uri*.

Subcategory

There are no subcategories in the eu.acdc.vulnerable_uri report category.

Schema

Vulnerable URI – eu.acdc.vulnerable_uri
A URI pointing to a vulnerable resource.
Required fields

report_category	string enum: eu.acdc.vulnerable_uri	The category of the report: a vulnerable URI.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the vulnerable URI was observed.
source_key	string enum: uri	The type of the reported object: a URI.
source_value	string format: uri	The URI to the vulnerable resource.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 2	The version number of the data format used for the report.
vulnerabilities	array items: object(identifier scheme, vulnerability identifier)	An array of objects describing vulnerabilities discovered at the vulnerable URI.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
duration	integer minimum: 0	The duration in seconds during which the vulnerabilities at the URI were observed.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
additional_data	object	Additional data for the observation. This allows putting more specific information into a report on a case by case basis in a structured manner. The usage of this field is at the data providers discretion.
alternate_format_type	string enum: CybOX, hpfeeds, IDMEF, IODEF, IPFIX, NetFlow v9, OpenIOC, sFlow, STIX	The type of the alternate format description of the observation.
alternate_format	string requires: alternate_format_type	A description of the observation in an alternate format. This is used to submit complex structured formats like IDMEF to the CCH.
ip_version	integer enum: 4, 6	The IP version of the IP address belonging to the vulnerable URI.
src_ip_v4	string format: ipv4	The source IPv4 associated with the vulnerable URI.
src_ip_v6	string format: ipv6	The source IPv6 associated with the vulnerable URI.

src_mode	string enum: plain, anon, pseudo	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymized IPs, or pseudo for pseudonymized IPs.
Dependencies		
If the report provides an ip_version, it has to provide either src_ip_v4 or src_ip_v6 depending on the ip_version being 4 or 6, respectively. For each provided IP address the corresponding mode has to be specified. If the report contains an alternate_format, it has to specify the alternate_format_type as well.		

Example

A SQL injection vulnerability in a search form can be sent to the CCH in the following way

```
{
    "report_category": "eu.acdc.vulnerable_uri",
    "report_type": "SQL injection in search field",
    "timestamp": "2014-06-15T15:47:12Z",
    "source_key": "uri",
    "source_value": "http://example.de/search",
    "reported_at": "2014-06-22T15:47:12Z",
    "confidence_level": 1.0,
    "version": 2,
    "vulnerabilities": [{
        "type": "cwe",
        "value": "CWE-89"
    }]
}
```

7.3.10. Botnet (eu.acdc.botnet)

The botnet report is used to submit botnets to the CCH. These reports are different from the other ones in that they introduce a connecting element between observations rather than an individual observation itself. Therefore, it does not conform to the minimal dataset as fields like timestamp are not meaningful for this element.

To promote a consistent usage in the other reports, botnet reports SHOULD be organised with a workflow similar to the following:

- only administrator users submit botnet reports
- the botnet reports to submit are agreed on via the Community Portal
- botnet reports do not expire but have to be explicitly removed
- the decision to remove a botnet report is agreed on via the Community Portal

Subcategory

A botnet report can fall into one of the following subcategories

- **p2p:** for a peer to peer (P2P) botnet
- **c2:** for a command and control server (C2) based botnet
- **other:** for example for hybrid control structures

Schema

Botnet		
A botnet tracked by ACDC.		
Required fields		
report_category	string	The category of the report: a botnet.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
source_key	string enum: botnet	The type of the reported object: a botnet.

source_value	string	The identifier of the botnet. This can be the name of a single type of botnet or a combination of a botnet type and an identifier for a specific instance of the botnet.
version	integer enum: 2	The version number of the data format used for the report.
report_subcategory	string enum: c2, p2p, other	The category of the botnet.
Optional fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.

Example

A Zeus botnet could be represented by the following report

```
{
    "report_category": "eu.acdc.botnet",
    "report_type": "ZeuS botnet 42",
    "source_key": "botnet",
    "source_value": "ZeuS-42",
    "reported at": "2014-06-22T15:47:12Z",
    "version": 2,
    "report_subcategory": "p2p"
}
```

7.4. Results from the Experiments and the Pilot

Employing the previously described data format in the experiments in WP3 identified a number of issues that either require clarification or modification of the format. Based on the observations in [D3.3] the following observations were made:

- **Confusion between bot and attack report categories**

  This observation was used as a starting point for a discussion on the descriptions of the mentioned report categories to improve the documentation.
- **Include mail header in the spam bot reports**

  With version 2 of the eu.acdc.attack report, the new optional mail_header field can be used to provide the header for a spam email.
- **More detailed spam campaign information**

  With version 2 of the eu.acdc.spam_campaign report schema, each spam campaign has a subcategory classifying the type of the campaign and an optional mail_body field to provide the body of the campaign emails. The party submitting the report to the CCH has to take care of replacing variable and especially personal information with a placeholder.
- **Include more info about the DDoS attacks**

  With version 2 of the eu.acdc.attack report schema, there are new optional fields bit_rate and packet_rate to provide an estimate of the traffic coming from the attacking system.
- **Identify mobile malware within a report**

  With version 2 of the eu.acdc.malware report schema, each malware can be annotated with a CPE name binding describing the platform that the malware is running on.
- **Give a severity score to the reported events**

  Since the severity of the reported event is in general difficult if at all to assess by the party submitting the report, it remains as a future improvement of the reports to include such information.

Besides the experiments, operation of the pilot revealed some areas for improvement as well. In order to support automatic processing of the reports the following changes should be implemented in the future:

- **subject_text is too general to identify spam campaigns**

  Using the subject text of a spam email as the identifier for a spam campaign raises the question how to handle campaigns that personalise the subject. Notwithstanding the legal aspects, plain usage of the subject does not yield a shared identifier in this case. Possible improvements are:

  - **Removing personalised parts of the subject:** This yields a common identifier but the identifier might be too general to capture one campaign.
  - **Introducing placeholders for personalised parts of the subject:** This yields a common identifier and preserves a larger part of the subject. But it requires agreeing on the way to introduce the placeholders. One way would be to just add an all-purpose placeholder for every variant part of the subject, another would be to introduce different placeholders for different kinds of variables in the subject, for example username or email address. In the order given this also increases the amount of coordination required between the CCH and the data providers in turn increasing the barrier to submit data to the CCH in the first place.

  Even without personalised subjects, the question arises whether the subject_text is specific enough to capture spam campaigns in general, see also the corresponding discussion in Section 7.2.8. To resolve this, more features of the spam emails have to be taken into account with the problem of coordination and danger of lowering data submissions to the CCH.

  A possible solution would be to include patterns in the spam campaign report that describe matching emails for example in YARA[1]. A future tool to be developed by the ACDC project could receive these patterns from the CCH and use them to locally match emails to these signatures and submit corresponding reports to the CCH while removing all personal information from them.

  With version 2 of the eu.acdc.spam_campaign report schema, the subject_text documentation suggests replacing all variable parts of the subject with a placeholder. A pattern matching approach with a tool provided to process patterns and emails should be considered for the future development of the reports.

- **eu.acdc.attack/abuse includes spam:**

  To better deal with spam as a type of abuse, there is a new subcategory abuse.spam with version 2 of the eu.acdc.attack report schema,.

- **No application_protocol in eu.acdc.attack reports:**

  Some attacks target standard services running on non-standard ports, like SSH password guessing against an SSH server on a port other than 22. Just providing the destination port without the application protocol in the eu.acdc.attack report might be misleading in this case and not provide the owner of the attacking host with important information regarding the kind of infection of his system.

  With version 2 of the eu.acdc.attack report schema, application_protocol is an optional field.

- **No eu.acdc.attack subcategory for scanning hosts:**

  To distinguish scans from other types of attacks, there is a new subcategory scan with version 2 of the eu.acdc.attack report schema.

[1] https://plusvic.github.io/yara/

8. Data Format for Aggregated Data

In this section, a data format family is introduced that specifically addresses the requirements for aggregated and statistical data. As such, one of its main goals is to support the work flows as defined by the statistical analysis of the data (WP4) and the research workflow in Section 10.4.5. Furthermore, it is designed to suit the needs of exchanging data resulting from the application of security metrics. The format family as shown distinguishes between different categories:

1. **Anonymized or pseudonymized data:**

   The data format shares most data fields with the minimal data set except that it is specifically designed to avoid any data that directly or indirectly relate to an individual such as IP addresses.

   IP addresses have to be either anonymized or pseudonymized before they are stored in the specific data fields. However, it is important to note that even if the data is pseudonymized, it may still be treated as person related data. For that reason, a legal basis for the exchange of such data might be required. Besides pseudonymized or anonymized data, means are provided to exchange aggregated or statistical data:

2. **Aggregated or correlated data:**

   This category allows aggregating related reports by specifying a container comprising links to all related reports. These are specifically the aforementioned anonymized reports. Typical use case is to detail correlations between multiple related reports, comprising for example a complex incident consisting of multiple steps. It is important to note, that this use case implies that the recipient is able to access all reports referred by the aggregated report.

3. **Statistical data:**

   This comprises time series of events in a specific time window (e.g., the number of attacks affecting an ASN). The aim is to use this format to export statistical properties to, for example, be able to apply models such as the ARIMA family. This category does specifically not contain any person related data.

4. **Metric data:**

   A lot of security metrics have been proposed to measure the performance of the detection mechanism and to compare the effectiveness of ISPs to react to threats. This type is designed for transferring the results of metrics. This category does specifically not contain any person related data.

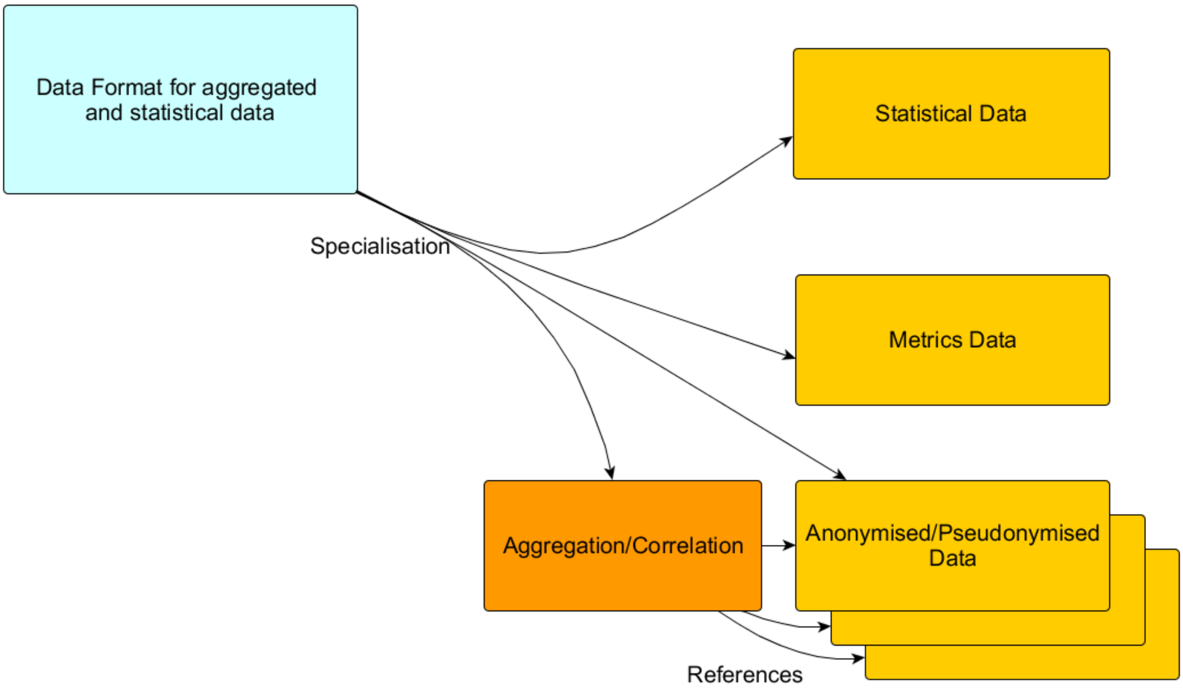

Data Format for aggregated and statistical data
Specialisation
Statistical Data
Metrics Data
Aggregation/Correlation
Anonymised/Pseudonymised Data
References

Figure 8: Overview of the data family for aggregated and statistical data

An overview of the data format family for aggregated and statistical data is shown in Figure 8. As described above the family is split into four categories that are distinguished by the field “report_category” which can contain the following values: “eu.acdc.metric”, “eu.acdc.correlated”, “eu.acdc.statistic”, and “eu.acdc.anonymized”. The aggregation/correlation format provides a container in which other related reports can be referenced. Thus, an incident can be represented by multiple anonymized or pseudonymized reports that are referenced by a root report. It is important to note that the partner submitting the aggregation report has to ensure that all referenced reports are either anonymized (e.g., only first 2 bytes of the IP) or pseudonymized. Since pseudonymized data might be treated as data that allows the identification of an individual, a specific data protection concept might be required to lawfully store and process this data.

8.1. Fields

The following table lists the fields used in the aggregated data format similar to Section 7.2 for the submission and notification format.

Field Name	Field Type	Title	Description
aggregation_criterion	string	Aggregation Criterion	The criterion used to aggregate or correlate the data, this could, for example, be a common ASN or class C network.
dst_port	integer	Destination port of the connection	*As defined above*
duration	integer minimum: 0	Duration of the observation	*As defined above*
incident_description	string	Description of aggregation	Textual description of the aggregation or correlation applied to the data. This field is complementary to the report type to comprise a more specific description if required.
ip_protocol_number	integer minimum: 0 maximum: 255	IP protocol number	*As defined above*
ip_version	integer enum: 4, 6	IP version number	*As defined above*
measurement_window	integer	Measurement Window	Time frame of measurement in seconds. This field is required for all formats pertaining statistical or metric data.
metric_id	integer	Metric Identifier	Identifier for the specific Metric
metric_result	object	Result of Metric	Resulting data (unstructured) of application of metric
report_category	string	Aggregation Type	This field characterises the type of aggregation which is associated with the specific format.
report_id	string	Report ID	*As defined above*
report_subcategory	string	Report subcategory	*As defined above*
report_type	string	Report type	*As defined above*

Field Name	Field Type	Title	Description
reported_at	string format: date-time	Time of the report's submission	*As defined above*
src_asn	string	Source ASN	ASN to which the source IP belongs to
src_city	string	Source City	City, the source belongs to (According to GeoIP Information)
src_country	string	Source Country	Country, the source belongs to (According to GeoIP Information)
src_key	string enum: botnet, ip, malware, subject, uri	Type of the reported object	The type of the reported object.
src_mode	string enum: anon, pseudo	Source IP mode	The type of the reported object. This format specifically avoids storage or transfer of sensitive or person related data. This requires to either fully anonymize the data (e.g., by discarding the last two bytes of the IP) or to replace the IP or URL with a pseudonym.
src_organisation	string	Source Organisation	Organisation (e.g., name of ISP) the source belongs to
src_port	integer	Source Port	*As defined above*
src_sl_domain	string	Second Level Domain	Second-level domain according to reverse DNS data
src_tl_domain	string	Top Level Domain	Top-level domain according to reverse DNS data
src_value	string	Identifier of the reported object	The identifier of the reported object like its IP address or URI.
statistic_description	string	Description of statistic	Textual description of statistics applied to the data. This field is complementary to the report type to comprise a more specific description if required.
statistic_criterion	string	Statistic Criterion	The concrete value (criterion) that has been used to aggregate the data. E.g. AS680 to enumerate all reports within the measurement window originating from that ASN. This field is complementary to the statistic field.
statistic_field	string	Statistic Field	This field specifies the field used as the criterion for the statistic. For example, “source_ASN” states that all reports that have the same value in this field are enumerated in the measurement window.
time_series	array items: integer	Time Series	This field contains an array comprising the values of the statistic. The length of the array corresponds to the number of lags

Field Name	Field Type	Title	Description
type_of_connection	string	Type of Connection	The type of the connection according to external information (e.g. Maxmind DB)

8.2. Report Schemata

The formats are specified using the notation introduced in Section 7.

The complete JSON schemata can be found in Appendix D.

8.2.1. Anonymized or Pseudonymized Data (eu.acdc.anonymized)

The “anonymized” or “pseudonymized” report is used to represent de-identified reports for the research workflow. The associated report_category is *eu.acdc.anonymized.*

The data format specifically avoids person related data, which in general include IP addresses. IP addresses have to be either anonymized or pseudonymized before inserting into these reports.

Subcategory

There are no subcategories for the eu.acdc.anonymized report.

Schema

Anonymized or Pseudonymized Data – eu.acdc.anonymized		
An anonymized or pseudonymized report.		
Required Fields		
report_category	string enum: eu.acdc.anonymized	The category of the report: anonymized or pseudonymized data.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the reported observation took place. This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.
version	integer enum: 2	The version number of the data format used for the report.
source_key	string enum: ip, malware, subject, uri	The type of the reported object.
src_mode	string enum: anon, pseudo	The type of the reported object. This format specifically avoids storage or transfer of sensitive or person related data. This requires to either fully anonymize the data (e.g. by discarding the last two Bytes of the IP) or to replace the IP or URL with a pseudonym.
source_value	string	The identifier of the reported object like its pseudonymized IP address or hash. If the source type “pseudonymous” is selected, a prefix preserving pseudonymization algorithm should be applied that preserves the data type for IP addresses.

src_asn	string	ASN of source
Optional Fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
src_organisation	string	Source of the attack, e.g. Deutsche Telekom A.G.
src_country	string	Source of the attack, e.g. Germany (Geolocation DB).
src_city	string	Source of the attack, e.g. Munich (Geolocation DB).
src_port	integer	The source port of the connection. This is always the port on the reported system (i.e., the one identified by source_value).
dst_port	integer	The destination port of the connection. This is always the remote port from the perspective of the reported system (i.e., the one identified by source_value). It can for example be the port of a honeypot that was contacted to infect it.
src_connection_type	string	The type of the connection according to external information (e.g., Maxmind DB)
src_tl_domain	string	Top-level domain according to reverse DNS data of source IP
src_sl_domain	string	Second-level domain according to reverse DNS data of source IP
additional_data	object	This field could contain arbitrary additional data. The user has to ensure, that no person related data is put into this field.

Example

```
{
	"report_category": "eu.acdc.anonymized",
	"report_type": "Pseudonymised research report: prefix preserving",
	"timestamp": "2014-06-15T15:47:12Z",
	"source_key": "ip",
	"source_value": "1.2.3.4",
	"src_mode": "pseudo",
	"src_asn": "AS680",
	"src_country": "Germany",
	"src_city": "Berlin",
	"src_organisation": "DFN",
	"src_connection_type": "Cable/DSL",
	"ip_protocol_number": 6,
	"src_port": 1025,
	"dst_port": 80,
	"src_tl_domain": "de",
	"src_sl_domain": "test.de",
	"version": 2
}
```

8.2.2. Aggregated or Correlated Data (eu.acdc.correlated)

This report is used to represent aggregated or correlated reports. The report_category is *eu.acdc.correlated*.

The report is designed to support the research workflow as well as advanced sensor systems (see Section 6). The specific fields contain the information to glue related reports (category “correlated”) together that comprise an incident. Incidents can be composed of multiple analogous reports (e.g., aggregation of connection concerning DDoS attempts) or can refer to different reports, e.g. in the format *eu.acdc.anonymized*, that all relate to the same incident (correlation). It is important to note, that there is no automated mechanism to ensure the integrity and availability of reports that are referenced in this data format. For that reason, the user has to take care that the recipient is able to access all referenced reports. This could, for instance, be done by ensuring that appropriate data sharing polices exist.

Subcategory

There are no subcategories for the eu.acdc.correlated report.

Schema

Aggregated or Correlated Data – eu.acdc.correlated		
An aggregation or correlation of multiple reports.		
Required Fields		
report_category	string enum: eu.acdc.correlated	The category of the report: an aggregation or correlation.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp details the starting date of the aggregation window. All reports whose original timestamp (this can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files) falls into the period of the measurement window (timestamp, timestamp + duration) are covered by the report.
duration	integer minimum: 0	The duration in seconds of the aggregation window.
version	integer enum: 2	The version number of the data format used for the report.
aggregation_criterion	string	The criterion used to aggregate or correlate the data, this could, for example, be a common ASN or class C network.
related_reports	array items: string	This field contains a list of reports that relate the current report. The current report can be considered to be a container that comprises links to all related reports.
incident_description	string	Textual description of the aggregation or correlation applied to the data. This field is complementary to the report type to comprise a more specific description if required.
Optional Fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.

reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.

Example

```
{
        "report_category": "eu.acdc.correlated",
        "report_type": "Attack correlation: DDoS",
        "timestamp": "2014-06-15T15:47:12Z",
        "duration": 3600,
        "aggregation_criterion": "Common ASN",
        "related_reports": [
                "54ae7d207765620ef2e20700",
                "54ae7e4d7765623b41312600",
                "54ae81d17765623700166f00"],
        "incident_description": "DDoS against an IRS server run at irc.eaxample.com.
The incident ID is DFN-CERT#12345678",
        "version": 2
}
```

8.2.3. Statistical Data (eu.acdc.statistic)

The statistical report is used to represent time series data of selected reports. The associated report_category is *eu.acdc.statistic*.

This format is defined for statistical data that is represented as a time series.

Subcategory

There are no subcategories for the eu.acdc.statistic report.

Schema

Statistical Data – eu.acdc.statistic		
Statistical data on selected reports.		
Required Fields		
report_category	string enum: eu.acdc.statistic	The category of the report: statistical data.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp details the starting date of the measurement windows. All reports whose original timestamp (This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.) falls into the period of the measurement window (timestamp, timestamp + measurement_window) are covered by the report.
version	integer enum: 2	The version number of the data format used for the report.
measurement_window	integer minimum: 0	Time frame of the overall measurement window in minutes.

statistic_field	string	This field specifies the field used as the criterion for the statistic. For example, “source_ASN” states that all reports that have the same value in this field are enumerated in the measurement window.
statistic_criterion	string	The concrete value (criterion) that has been used to aggregate the data. E.g. AS680 to enumerate all reports within the measurement window originating from that ASN. This field is complementary to the statistic field.
time_series	array items: integer	The time series is represented by an array consisting of an integer value for each lag.
statistic_description	string	Textual description of statistics applied to the data. This field is complementary to the report type to comprise a more specific description if required.
Optional Fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.

Example

```
{
	"report_category": "eu.acdc.statistic",
	"report_type": "Statistic of daily reports per submission key",
	"timestamp": "2014-06-15T15:47:12Z",
	"measurement_window": 86400,
	"statistic_field": "api_key_id",
	"statistic_criterion": "453",
	"time_series": [100, 101],
	"statistic_description": "Number of reports originating from a submission key
within a day",
	"version": 2
}
```

8.2.4. Results of Metrics (eu.acdc.metric)

These fields apply to data resulting from a security metric. The supported metrics are defined in [D4.4].

Subcategory

An eu.acdc.metric report can fall into one of the following subcategories:

- **event_based_metric:** metrics counting reports
- **id_based_metric:** metrics counting unique identifiers (e.g., bot IDs)
- **ip_based_metric:** metrics counting unique ip addresses
- **quality_metric:** metrics intended to measure the quality of the data

Schema

Results of Metrics – eu.acdc.metric		
Metric evaluation of reports.		
Required Fields		
report_category	string enum: eu.acdc.metric	The category of the report: metric evaluation of reports.

report_subcategory	String enum: event_based_metric, id_based_metric, ip_based_metric, quality_metric	The subcategory of the metric.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb, this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp details the starting date of the measurement windows. All reports whose original timestamp (This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.) falls into the period of the measurement window (timestamp, timestamp + measurement_window) are covered by the report.
version	integer enum: 2	The version number of the data format used for the report.
metric_id	integer minimum: 0	Identification number of applied metric. Applied metrics are specified in a separate document which enumerates them and provides information about the resulting values.
measurement_window	integer minimum: 0	Time frame of measurement window in minutes
metric_result	object	Resulting data. The value depends on the specific metric. This is an integer for all metrics that count events.
metric_description	string	Textual description of the security metric. This field complements the report type is a more specific description is intended.
Optional Fields		
report_id	string	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.

Example

```
{
        "metric_id": 2,
        "report_type": "[METRIC][IP_METRICS][TUD] Unique IPs per Country",
        "measurement_window": 259200,
        "timestamp": "2015-05-07",
        "version": 2,
        "metric_description": "Unique IPs per Country",
        "metric_result": {
                "RU": {"c2_server": {"706": {"data": 1}}},
                "DE": {"attack": {"412": {"data": 61}}},
                "GB": {"c2_server": {"706": {"data": 1}}},
                "US": {"c2_server": {"706": {"data": 10}}}},
        "report_category": "eu.acdc.metric",
        "report_subcategory": "ip_based_metric"
}
```

8.3. *Results from the Computation of Metrics*

The computation of metrics in WP4 revealed a couple of possible improvements of the report schemata. The following observations were made:

- **Bot ID missing in eu.acdc.bot report:**

  Meaningful metrics on botnets benefit substantially from IDs internally used by botnets to identify individual bots. Based on these IDs more accurate estimates of the size of botnets as well as validation of other observations like DHCP churn are possible.

  With version 2 of the eu.acdc.bot report, there is a new field bot_id for the ID of the reported bot. Since bot IDs are not available to every kind of sensor and some bonets do not use internal IDs at all, the field is optional.
- **Spam cannot be distinguished from other abuse:**

  Computing dedicated metrics on spam is difficult because spam is subsumed in the abuse subcategory of eu.acdc.attack.

  With version 2 of the eu.acdc.attack report, there is a new subcategory abuse.spam to explicitly submit spam reports to the CCH.

9. Data Format for End Customer Notification

While suited for automatic processing by parties directly accessing the CCH, the previously described format cannot be used on its own to notify for example end customers. Parties not directly involved with ACDC lack the information necessary to interpret these informations and putting the burden to search for that information on the receiving party reduces the chances of the reports being acted upon.

As can be seen from the previous analysis of data formats, X-ARF is a format that offers on the one hand a plain text description of what is reported, providing the information to interpret the report to the receiving party. On the other hand, the machine readable part of X-ARF reports already uses JSON schema and is thus similar to the previously described JSON format to communicate with the CCH.

Since there is a difference between the existing X-ARF report schemata and the ACDC report categories with regards to the information contained in the reports, ACDC defines its own X-ARF schemata. This is in line with the established usage policy of the X-ARF project[1].

The following six ACDC report categories can be meaningfully converted to X-ARF and sent to customers: eu.acdc.attack, eu.acdc.bot, eu.acdc.c2_server, eu.acdc.fast_flux, eu.acdc.malicious_uri, and eu.acdc.vulnerable_uri. The report categories eu.acdc.malware, eu.acdc.spam_campaign, and eu.acdc.botnet represent information that is not sent out to end customers. These categories are not necessary to process a single incident but they only provide context to other reports. For example an eu.acdc.malware report describing the hosted malware behind an eu.acdc.malicious_uri report or an eu.acdc.spam_campaign report describing the spam campaign connected to an eu.acdc.attack:abuse report.

The X-ARF standard requires the definition of some additional fields for each converted ACDC report:

- **Reported-From:** The email address of the sender. This is ACDC sending the report to the recipient, not the party originally submitting the report to the CCH.
- **Category:** A general categorisation of the report into one of five categories defined in X-ARF

Category	Explanation
abuse	technical abusive behaviour - any kinds of attacks like virus, malware, bot, logins, etc.
fraud	financial abuse like credit card fraud, etc.
auth	misuse or failure of authentication methods, SSL, SSH, POP3, etc.
info	all sorts of pure informational reports like blacklistings, delistings
private	all sorts of closed information exchange between 2 or more parties

All converted ACDC reports are assigned to the abuse category.

- **Report-Type:** The report_category of the ACDC report.
- **User-Agent:** The name and version of the software converting the reports.
- **Report-ID:** The ID of the report with a domain part to ensure uniqueness over data providers. This is the report_id in the CCH with the suffix “@acdc-project.eu”.
- **Date:** The timestamp of the incident occurring. This is the timestamp in the ACDC report.
- **Source:** The source of the abusive behaviour. This is the source_value in the ACDC report.
- **Source-Type:** The type of the Source. This is the source_key in the ACDC report with the split of the source_key “ip” into “ipv4” and “ipv6” according to the ip_version.
- **Attachment:** Whether the X-ARF report has an attachment with additional information. Currently no further information is attached to the reports and this is set to “none”.

[1] https://github.com/abusix/xarf-schemata

- **Schema-URL:** A URL pointing to the JSON schema describing the X-ARF report. X-ARF puts the version number of the applicable X-ARF standard in each report, see also the next item. The version of the schema is instead part of the schema URL. The schema version is a three part numerical string of the form x.x.x. To synchronise the submission and notification schemata with the X-ARF schemata, the first part of the X-ARF schema version SHALL be the same as the version field of the represented CCH report. The second and third part MAY be used to signify changes only relevant to the X-ARF schema itself. Thus, a CCH report using the eu.acdc.attack schema in version 1, MUST be represented by an X-ARF report following an eu.acdc.attack_1.x.x schema. The schemata for the above mentioned report categories can be found in the appendix.
- **Version:** The version of the relevant X-ARF specification. The current version of the specification at the time of writing is 0.2.
- **Occurrences:** The number of similar observations aggregated in this report. This field is optional for X-ARF and not set in the conversion of ACDC reports as there is no aggregation performed.
- **TLP:** The sensitivity of the report in the Traffic Light Protocol[1]. This field is optional for X-ARF and not set in the conversion of ACDC reports.

For an example of the X-ARF reports, we start with the sample for an eu.acdc.attack report from above:

```
{
       "report_category": "eu.acdc.attack",
       "report_subcategory": "dos",
       "report_type": "TCP SYN Flood",
       "timestamp": "2014-06-15T15:47:12Z",
       "source_key": "ip",
       "source_value": "192.0.2.14",
       "ip_protocol_number": 6,
       "ip_version": 4,
       "src_ip_v4": "192.0.2.14",
       "src_mode": "plain",
       "dst_ip_v4": "198.51.100.111",
       "dst_mode": "anon",
       "dst_port": 80,
       "confidence_level": 1.0,
       "version": 1
}
```

The converted X-ARF report would look like the following, where the horizontal line marks the boundary between two MIME parts in the email sent.

```
Dear customer,

this is an X-ARF report on activity related to your ASNs and networks that was
recently submitted to the ACDC project.

A host attacks another system:
  TCP SYN Flood

For more information on the ACDC project please visit
  http://www.acdc-project.eu/

For more information on X-ARF please visit
  http://www.x-arf.org/
```

```
Attachment: none
Category: abuse
Confidence-Level: 1.0
Date: '2014-06-15T15:47:12Z'
Dst-Ip-V4: 198.51.100.111
Dst-Mode: anon
Dst-Port: 80
```

[1] https://www.trusted-introducer.org/ISTLPv11.pdf

```
Ip-Protocol-Number: 6
Ip-Version: 4
Report-Description: 'TCP SYN Flood'
Report-ID: some-report-id@acdc-project.eu
Report-Subcategory: dos
Report-Type: eu.acdc.attack
Reported-At: '2014-06-15T15:44:45Z'
Reported-From: xarf@acdc-project.eu
Schema-URL: https://www.acdc-project.eu/eu.acdc.attack_2.0.0.json
Source: 141.39.242.16
Source-Type: ipv4
Src-Ip-V4: 192.0.2.14
Src-Mode: plain
User-Agent: acdc-xarf-export/1.0
Version: 0.2
```

Sending multiple reports is supported by X-ARF bulk messages, which collect several individual reports into one email.

The complete X-ARF schemata used for ACDC notifications can be found in Appendix E.

10. Data Submission and Retrieval Workflows

ACDC potentially enables a huge number of heterogeneous stakeholders to participate in data sharing activities on a European level via its solutions. However, due to ACDC's complex requirements on data management and the diversity of the potential stakeholders, no single way of access to sharing data via ACDC is possible. Instead, the required access parts and related workflows differ depending on the type of stakeholders as well as the type of data that shall be shared.

This section describes concrete workflows for the different stakeholder types or user groups as identified in [D6.1.2] who want to submit or retrieve data via the ACDC solution.

Besides the workflows described here, individual data sharing is possible with a mutual sharing agreement between parties. The sharing can be set up by using the Data Access Manager inside of the ACDC community portal. For further documentation refer to the documentation in the Data Access Manager section of the Community Portal[1].

10.1. Workflow description

Workflows that are described here are intended to provide interested stakeholders an easy way to determine what they need to do to send data to ACDC or to receive data from the ACDC solutions.

In order to provide that, there should be one intended way of interaction with ACDC depending on what the stakeholder wants to do. For data submissions the stakeholder has to determine what he is allowed to send and what restrictions he has to apply on the datasets he submits. Data submission workflows to interact with ACDC are more technically determined, for example, by the type of the sensor which generates the data to be sent. Data retrieval workflows on the other hand are based on the stakeholder's Cyber Positioning, as this defines what he is allowed to see and how he is allowed to use it.

As these workflows also define how potentially private data is handled within the ACDC solution, they have to be checked against the project's legal requirements. This is achieved by including the task leader of Task 1.2.1 and Task 1.2.2 in the workflow definition by the means of providing them use cases to check.

The following will focus on technically derived workflows for data submission and exactly one workflow per Cyber Positioning of the interested stakeholder for data retrieval. The workflow requirements below provide easy rules which should ensure this differentiation.

10.2. Workflow requirements

In order to ensure defined workflows are capable of enabling data sharing between the different interested stakeholders the following requirements and restrictions MUST be followed by each individual workflow.

- All workflows MUST describe how the stakeholder is identified and by whom.
- All workflows MUST be defined inside this document.
- All workflows MUST be coordinated by one responsible ACDC partner.
- All workflows MUST be following the ACDC legal requirements for information sharing.

For data submission workflows the following MUST be satisfied in addition.

- There MAY be data usage restrictions applied for data submitted to ACDC. These restrictions – if applied – MUST follow the restriction possibilities documented inside the ACDC community portal.
- Workflows for data submission MAY be determined by other means than the stakeholders Cyber Positioning.
- There MAY be more than one workflow for data submission into ACDC per Cyber Positioning.

For data retrieval workflows the following MUST be satisfied in addition.

- There MUST be only one workflow for data retrieval per Cyber Positioning.

[1] https://communityportal.acdc-project.eu/

- A stakeholder MAY fall into more than one Cyber Positioning and MAY thus be receiving data following more than one workflow.
- All workflows for data retrieval MUST define the usage restrictions for data retrieved from ACDC.
- All workflows for data retrieval MUST define the scope of the data the stakeholder is allowed to retrieve (e.g., AS number or network range) and any needed authorisation methods for this scope MUST be defined.

10.2.1. Basic Stakeholder Identification

The required level of identification used in a workflow depends on the sensitivity of data transferred using the workflow. Workflows that only submit data as well as workflows that do not receive sensitive data, for example only data in anonymized form, may use the basic stakeholder identification. The basic identification only requires valid credentials to access the CCH. This relies on the process in place to join the ACDC project and the API keys to access the CCH only being provided after completing this process.

To join ACDC, a stakeholder has to apply through the Community Portal. This application includes basic identifying information like the name of the stakeholder. The application is then reviewed by an Application Manager in the Community Portal. A detailed description of the registration can be found in [D6.2.1].

10.3. Data Submission Workflows

This section lists the defined workflows for data submission to ACDC.

10.3.1. Format and Description of a Data Submission Workflow

To define a data submission workflow the template below must be used. It includes explanations about the semantics of the different fields and paragraphs.

$NAME data Submission Workflow

Please provide a convenient name for the workflow (e.g., "Darknet", "DDoS-sinkhole", "Malware").

Responsible ACDC partner: *name of the ACDC partner coordinating this workflow*
Version of this Workflow: *version number* of this workflow (e*.g. 1.0)*
Aim of this workflow:

Free-text description of what this workflow wants to address.

State of this workflow:

Free-text description of the state of implementation.

Configurable usage restrictions on the submitted data:

Describe any usage restrictions which MUST apply on the data submitted following this workflow. Any restrictions defined here MUST be in accordance with the capabilities of the CCH and the Community Portal.

Identification of the stakeholder:

Describe how the stakeholder is identified and who does identify the stakeholder. Also describe how the stakeholder's data is kept up-to-date.

Data formats or schemata sent when using this workflow:

Please describe what data formats and schemata are used for the data sent to ACDC. Please make sure that either the schema is already defined in D1.7.2 or follow the schema processes to introduce a new one, if needed. Also, the ACDC "Description of Work" lists aims concerning data formats and schemata which should be taken into account.

Textual description of the workflow:

Provide a textual description of the workflow including what the stakeholder, the ACDC project, and any other interacting entity has to do to follow the workflow.

10.3.2. Basic Data Submission Workflow

Responsible ACDC partner: DFN-CERT
Version of this Workflow: 1.0

Aim of this workflow:

This workflow is intended for the delivery of sensor data into the CCH. It provides basic requirements for the use of sensor data in ACDC. More targeted workflows MAY be used for sensor data where available.

State of this workflow:

This workflow is in draft-state.

Configurable usage restrictions on the submitted data:

Usage restrictions on data submitted following this workflow MUST NOT be applied. Instead, any data which is not intended for general sharing with affected parties in unanonymized form MUST BE anonymized upon data submission.

Identification of the stakeholder:

The data submitter is identified by means of an API Key and token issued by the CCH through the Community Platform.

This workflow only submits data. It uses the basic identification for stakeholders.

Data formats or schemata sent when using this workflow:

Data sent following this workflow MUST be JSON data conforming to one of the report categories defined in Section 7.

Textual description of the workflow:

To submit data using this workflow, a stakeholder has to join ACDC and obtain an API key and token to access the CCH. The API key and token are used to identify the stakeholder in sensor data reports sent to the CCH.

Upon receipt of data, the CCH authenticates the stakeholder based on API key and token. Once authenticated, the reports are subject to the general processing in the CCH.

10.3.3. Correlator Data Submission Workflow

Responsible ACDC partner: ATOS

Version of this Workflow: 0.9

Aim of this workflow:

This workflow is used exclusively to submit information derived from the correlation of data obtained from the CCH with the Correlator Data Retrieval Workflow in Section 10.4.3.

State of this workflow:

Not implemented during the ACDC project duration, as no correlators by partners outside of ACDC were planned during the project duration.

Configurable usage restrictions on the submitted data:

Submitted data does not have any usage restrictions besides the one established by default in the CCH (and governed by Data Retrieval workflows) and through the Community Platform (i.e., sharing policies), which are handled by the CCH component itself.

Identification of the stakeholder:

The Correlator component is identified by means of an API Key and token issued by the CCH through the Community Platform.

This workflow only transfers personal data in pseudonymized form. It uses the basic identification for stakeholders.

Data formats or schemata sent when using this workflow:

Reports submitted according to this workflow SHALL be JSON data conforming to the following categories defined in Section 7: eu.acdc.bot, eu.acdc.attack, eu.acdc.malware, eu.acdc.fast_flux, eu.acdc.vulnerable_uri, eu.acdc.malicious_uri, eu.acdc.spam_campaign and eu.acdc.c2_server.

Textual description of the workflow:

To submit data using this workflow, a stakeholder has to join ACDC and obtain an API key to access the CCH.

To follow this workflow, the Correlator SHALL obtain the data to work with using the Correlator Data Retrieval Workflow, see 10.4.3. The Correlator MAY use other sources to inform the correlation process. If the correlation yields new information on pseudonymized assets retrieved from the CCH that the Correlator wants to submit back to the CCH, the report SHALL use this pseudonymized identifier. This workflow SHALL NOT be used to submit any information not based on reports previously obtained via the Correlator Data

Retrieval Workflow. All information submitted to the CCH SHALL be expired from the correlation component.

Upon submission of a report under this workflow, the CCH depseudonymizes the report to reattribute it to the affected asset. This introduces the submitted report into the usual handling reports in the CCH.

10.4. Data Retrieval Workflows

The following section lists the defined workflows for data retrieval.

10.4.1. Format and Description of a Data Retrieval Workflow

To define a data retrieval workflow the template below must be used. It includes explanations about the semantics of the different fields and paragraphs.

$NAME Data Retrieval Workflow

Please provide a convenient name for the workflow (e.g. “Research”, “National LEA”, “CERT”).

Cyber Positioning: *name of the Cyber Positioning(s) this workflow addresses*
Responsible ACDC partner: *name of the ACDC partner coordinating this workflow*
Version of this Workflow: *version number of this workflow (e.g. 1.0)*
Aim of this workflow:

Free-text description of what this workflow wants to address.

State of this workflow:

Free-text description of the state of implementation.

Usage restrictions on the received data:

Please describe how the receiving party is allowed to handle the data. Please be aware that you most possibly just can describe what he MAY do with the data as well as what he MUST NOT do with it. Enforcing a stakeholder to guarantee usage of the data in a certain way (e.g. enforcing him to provide data to LEA) might be impossible due to the stakeholder's local regulations, contractual or legal situations. This paragraph implicitly defines what data in what form (e.g. anonymized) is sent out to the stakeholder as it defines what the stakeholder is allowed to do with it. Datasets which are not allowed to be handled by the stakeholder at all or where the usage in the described way would not be possible MUST NOT be sent out by the CCH over this defined workflow. However, the stakeholder MAY receive data following more than one workflow depending on his Cyber Positioning.

Identification of the stakeholder:

Describe how the stakeholder is identified and who does identify the stakeholder. Also describe how the stakeholder's data is kept up-to-date.

Scope of the data retrievable by this workflow:

Describe which data is receivable by the stakeholder following this workflow by means of the data scope, meaning: for which IP-addresses, which domains / TLDs, which ASN, or data without any scope (e.g. malware binaries for research, statistical information). Also describe how the scope is determined (e.g. RIPE data, assurance) and the state of anonymization / pseudonymization if applicable.

Data formats or schemata received when using this workflow:

Please describe what data formats and schemata are used for the data retrievable by the stakeholder. Please make sure that either the schema is already defined in D1.7.2 or follow the schema processes to introduce a new one, if needed. Also, the ACDC “Description of Work” lists aims concerning data formats and schemata which should be taken into account.

Textual description of the workflow:

Provide a textual description of the workflow including what the stakeholder, the ACDC project, and any other interacting entity has to do to follow the workflow.

10.4.2. CERT Data Retrieval Workflow

Cyber Positioning: Operational team: CERT, CSIRT
Responsible ACDC partner: DFN-CERT
Version of this Workflow: 1.0

Aim of this workflow:

This workflow is intended for CERTs to provide them with information about infected, attacked or attacking systems inside their constituency. Data obtained with this workflow MUST NOT be restricted in a way so that notification and support of the constituency would not be possible for the CERTs receiving the data. Data needed for general research as well as data which is not allowed to be shared with its constituency MUST be obtained by other workflows.

State of this workflow:

This workflow is functional.

It is temporary disabled or its data output is redirected to test mailboxes in times where experimental data or test datasets are submitted into the CCH.

Usage restrictions on the received data:

The data received via this workflow MAY be shared within the constituency of the receiving CERT as needed by its incident handling and support procedures. Data sets MAY include a TLP field allowing the CERT to share the data set beyond its community. If the CERT provides LEA support for its constituency, sharing the obtained data with LEA is allowed.

Identification of the stakeholder:

The identification of the stakeholder is done by the Trusted Introducer[1] (TI) body, which provides this service for its accredited as well as certified members. TI provides identification and authorisation methods for CERTs as well as enforcing a high degree of data quality. All its member teams are required to regularly, at least every four months, update their member data to keep it up-to-date.

Scope of the data retrievable by this workflow:

Any CERT following this workflow MAY receive data for its constituency, which is defined by IP networks and ASNs. Domain- or TLD-based constituencies are not yet possible due to the ownership of data belonging to these not yet being decided inside ACDC.

The stakeholder assures towards TI that it is eligible to receive data for the constituency scope as defined inside its member data sets.

Data Formats or Schemata received when using this workflow:

The stakeholder will receive X-ARF data by email using the format described in Section 9. If the API key is provided to the stakeholder, it will also be able to query the CCH directly.

Textual description of the workflow:

The CERT interested in receiving data from ACDC needs to obtain the level “accredited” or above from the Trusted Introducer service. This requires an active sponsorship of two current members of the TI community in order to reach the “listed” level. As a “listed” team it can apply for TI Accreditation. The process to become an “accredited” team usually takes between two and ten weeks. During this time the applying team needs to explain its services, constituency, point of contact and some important incident handling related policies. The team also needs to declare its acceptance of certain standard procedures like the TLP.

Once accredited, the team is able to edit its team information through a TI self-service interface. Using this interface an accredited team can configure the IP addresses and ASN numbers for which it wants to receive data from ACDC. Additionally, a specific email address for receiving all data from ACDC may be defined. Standard TI processes including the requirement to keep the team information up-to-date ensure that all data are actively maintained satisfying all further ACDC requirements.

Based on the configured entries, the TI service will either add, remove or modify the related API user entry at ACDC's CCH to adapt to the constituency scope defined above.

ACDC will then sent out data using X-ARF following the configuration of the API user configured above.

10.4.3. Correlator Data Retrieval Workflow

Cyber Positioning: Technology Provider
Responsible ACDC partner: ATOS
Version of this Workflow: 0.9
Aim of this workflow:

[1] http://www.trusted-introducer.org/

This workflow is intended to be used by technology providers integrating their correlation solution with the ACDC project. It provides data to the Correlator pseudonymized in a way that allows the CCH later reattribution to affected assets for data resubmitted with the Correlator Data Submission Workflow described in Section 10.3.3.

State of this workflow:

Not implemented during the ACDC project duration, as no correlator by partners outside of ACDC were planned during the project duration.

Usage restrictions on the received data:

These data sets MUST NOT be made available for commercial purposes. These data sets MAY be used for correlation purposes. These data sets MAY be stored in a temporal database for a predefined (configurable) period, enough for performing the correlation processes defined. These data sets MUST be deleted once the storage period is expired. These data sets MAY be used for computing statistics during the storage period as long as this computation is done by a software component deployed in the correlation component environment (i.e. the statistics module is part of the correlation component).

Identification of the stakeholder:

The Correlator component is identified by means of an API Key and token issued by the CCH through the Community Platform.

This workflow does not transfer personal data. It uses the basic identification for stakeholders.

Scope of the data retrievable by this workflow:

The correlation component will be able to retrieve data sets from the CCH (CCH reports) complying with the data format defined in Section 7. All CCH report categories listed below must be received by the correlation component. CCH reports containing information belonging to all the ASNs and all IP ranges must be received by the correlation component. Any personal data included in the CCH reports and received by the correlation component must be pseudonymized by the CCH in a way that

- is stable over a time window of one day allowing consistent correlation over this period,
- is reversible only by the CCH allowing the reattribution of the correlation results to the affected systems.

Data formats or schemata received when using this workflow:

Reports retrieved from the CCH according to this workflow (CCH report) comply with those defined in D1.7.2 for categories: eu.acdc.bot, eu.acdc.attack, eu.acdc.botnet, eu.acdc.malware, eu.acdc.fast_flux, eu.acdc.vulnerable_uri, eu.acdc.malicious_uri, eu.acdc.spam_campaign and eu.acdc.c2_server.

Textual description of the workflow:

To retrieve data using this workflow, a stakeholder has to join ACDC as a technology provider and obtain an API key to access the CCH.

The data obtained via this workflow SHALL only be used for correlation purposes. Correlation results SHALL be submitted back to the CCH using the Correlator Data Submission Workflow (see 10.3.3) to assure proper reattribution of the results in the CCH. All information the stakeholder retrieves from the CCH SHALL be expired from the correlation component.

Reports obtained via this workflow SHALL be pseudonymized by the CCH as described above. The key used in the pseudonymization SHALL be stable only for one day. This precludes depseudonymization and the creation of profiles by the Correlator via long term observations.

10.4.4. Internet Service Provider Data Retrieval Workflow

Cyber Positioning: Internet Service Provider (ISP), Hosting provider
Responsible ACDC partner: ECO
Version of this workflow: 0.9
Aim of this workflow:

This workflow is intended for ASN or Network Owners (Local Internet Registry (LIR), Internet Service Providers, Hosting Providers, Companies) as far as they are configured as the abuse-contact of the respective ASN or Network objects in the RIPE database. It shall enable them to gain full data about incidents in their address space via ACDC in order to inform/mitigate incidents in their address space.

Any Network Owner which is not configured as abuse contact is seen as having delegated the abuse handling, and as such not eligible to follow this workflow.

State of this Workflow:

The manual workflow is operational. The proposed automatic workflow is work in progress.

Usage restrictions on the received data:

There is no restriction regarding a network owners own address space, so a network owner MAY use retrieved data to mitigate the incident in his infrastructure or to inform the infected end user about workstation infection or server compromise with all technical details available.

Identification of the Stakeholder:

Identification of the stakeholder is done using the information stored with RIPE. Every network owner has filed an abuse email address at RIPE via which he can be reached. The "Abuse contact info" states the abuse address of an ASN / IP owner and can be used to validate a stakeholder's claim to be responsible for a certain set of ASNs / IPs.

The information on assigned ASN and IP ranges is kept up to date via RIPE, which performs and registers changes of ASN and IP assignments to network owners.

Scope of the data retrievable by this workflow:

Any stakeholder following this workflow MAY receive data for its constituency, which is defined by IP-networks and ASNs. Domain- or TLD-based constituencies are not yet possible due to the ownership of data belonging to these not yet being decided inside ACDC.

Data Formats or Schemata received when using this workflow:

The stakeholder will receive JSON reports conforming to the report schemata defined in Section 7.

Textual description of the workflow:

The initial identification for the stakeholder is shown in Figure 9. Every network owner has to file an abuse email address with RIPE via which he can be reached. This email address can easily be retrieved by performing a RIPE database search for an ASN or an IP address. The "Abuse contact info" states the abuse address of the ASN / IP owner and can be used to make initial contact, once a network owner applies for a membership.

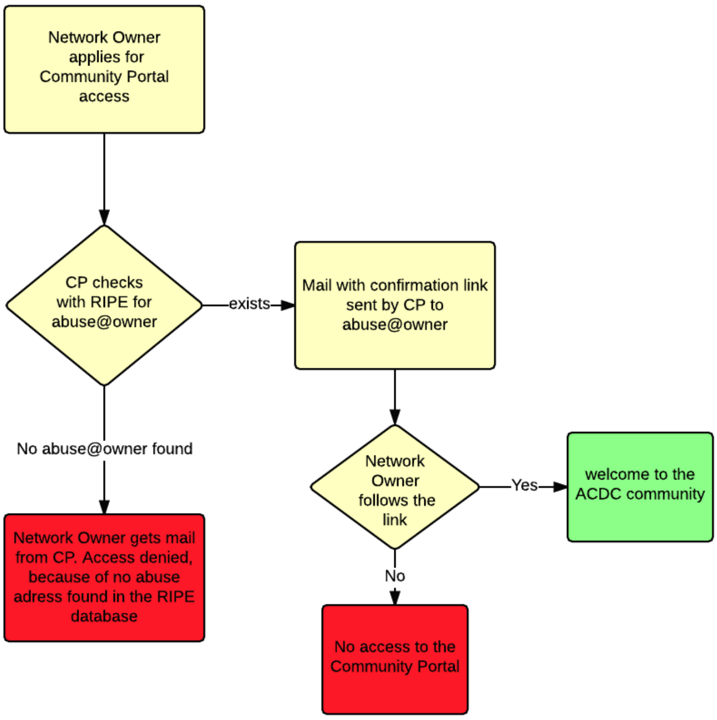

Network Owner applies for Community Portal access
CP checks with RIPE for abuse@owner
exists
Mail with confirmation link sent by CP to abuse@owner
No abuse@owner found
Network Owner follows the link
Yes
welcome to the ACDC community
No
Network Owner gets mail from CP. Access denied, because of no abuse adress found in the RIPE database
No access to the Community Portal

Figure 9: Validation of an ISP by the Community Portal

The Community Portal checks the RIPE database for the existence of the abuse address and sends an email with a confirmation link to the address. Once the network owner follows this confirmation link, the registration is complete and the network owner joined the ACDC community.

Process of ASN verification:

All European ISPs and LIRs – as far as they are directly working with Internet number resources within Europe – have to be member of the RIPE NCC, which oversees the allocation and registration of Internet number resources for Europe, the Middle East and parts of Central Asia. The RIPE database contains registration information for networks in the RIPE NCC service region and related contact details.

A network owner can state his ASNs and IPs in the Community Portal to create CCH API keys to read data which falls into his ASN / IP range. These ASNS or IPs have to be checked to make legally sure the ISP owns these particular ASNs / IPs.

Workflow as it is in present:

At the moment this is done manually as shown in Figure 10: The network owner enters his ASN list in the Community Portal and the Community Portal sends a note to a CCH manager. The manager manually checks the IPs and ASNs, can modify them in case of typos, and finally can approve or deny the request.

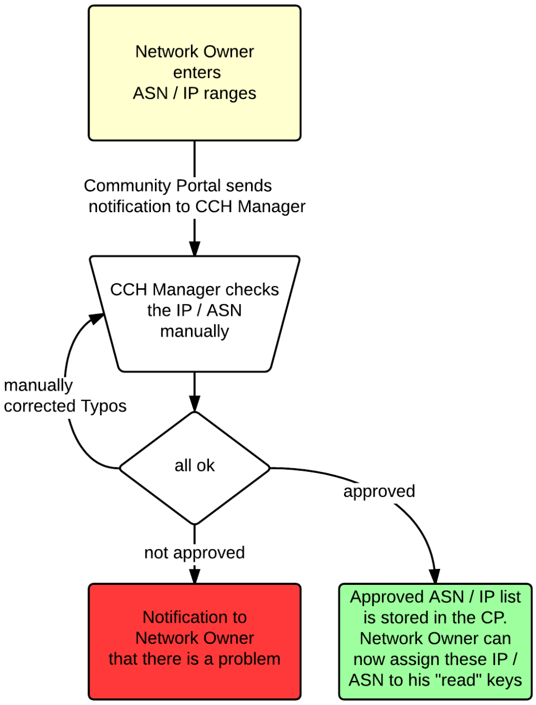

Network Owner enters ASN / IP ranges
Community Portal sends notification to CCH Manager
CCH Manager checks the IP / ASN manually
manually corrected Typos
all ok
approved
not approved
Notification to Network Owner that there is a problem
Approved ASN / IP list is stored in the CP. Network Owner can now assign these IP / ASN to his "read" keys

Figure 10: ASN and IP range creation and approval process “as it is”

After the request is approved, the network owner can create read keys and assign his approved ASNs / IPs to them.

A solution without manual interference would be highly appreciated.

Proposed workflow:

The same process as the initial network owner verification could do the verification of ASN and IP ranges. As all network owners’ ASN and IP ranges are registered at the RIPE database and therefore an abuse contact address is stated, the Community Portal can check the RIPE database for the “Abuse contact info” of a given ASN or IP range as depicted in Figure 11.

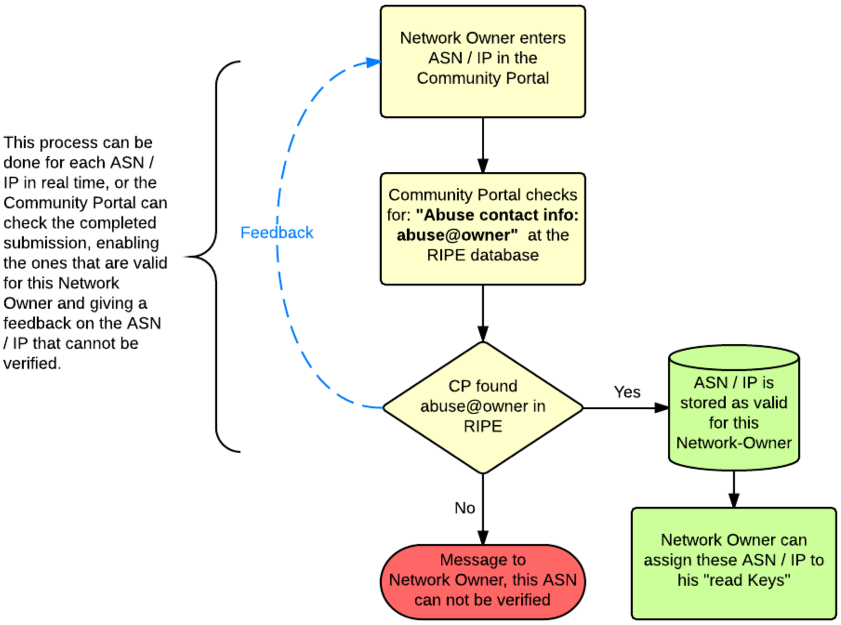

Network Owner enters ASN / IP in the Community Portal
Community Portal checks for: "Abuse contact info: abuse@owner" at the RIPE database
CP found abuse@owner in RIPE
Yes
ASN / IP is stored as valid for this Network-Owner
Network Owner can assign these ASN / IP to his "read Keys"
No
Message to Network Owner, this ASN can not be verified
Feedback
This process can be done for each ASN / IP in real time, or the Community Portal can check the completed submission, enabling the ones that are valid for this Network Owner and giving a feedback on the ASN / IP that cannot be verified.

Figure 11: Automated ASN / IP approval process by the CP

If the “Abuse contact info” matches the network owner’s given abuse information, the ASNs / IPs belong to them and can be approved by the Community Portal. If not, the network owner will be informed that this particular ASN / IP cannot be verified. It is then up to the network owners to check for their “Abuse contact” settings within RIPE or to check if there was an error in the ASN / IP entered.

Periodic check on data integrity:

As ASN / IP ranges can be easily reassigned to other companies or given back to RIPE, there must be a process to check these assets of registered network owners for changes.

This can be performed by the Community Portal as it stores the ASN / IP list of every registered user. A verification process similar to Figure 12 can be run on these lists on a daily basis to maintain data consistency.

If a change is found, reassigned ASNs / IPs have to be deleted from the network owners list. A message about the finding has to be send to the network owner. Additionally, the API keys have to be updated so that the network owner will not receive data about somebody else’s constituency.

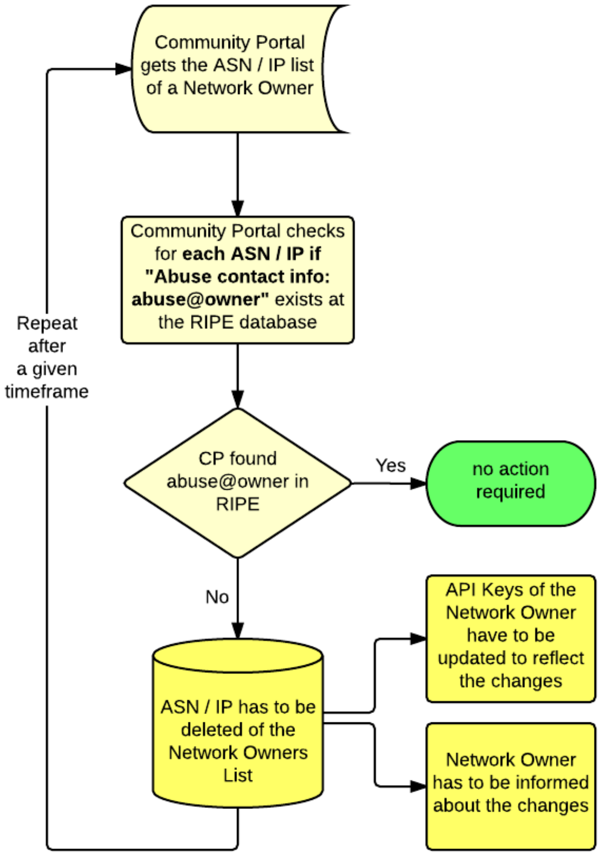

Community Portal gets the ASN / IP list of a Network Owner
Community Portal checks for **each ASN / IP if "Abuse contact info: abuse@owner"** exists at the RIPE database
Repeat after a given timeframe
CP found abuse@owner in RIPE
Yes
no action required
No
API Keys of the Network Owner have to be updated to reflect the changes
ASN / IP has to be deleted of the Network Owners List
Network Owner has to be informed about the changes

Figure 12: Periodical check of network resources

10.4.5. Research Data Retrieval Workflow

Cyber Positioning: Research
Responsible ACDC partner: TU-Delft
Version of this Workflow: 0.8
Aim of this workflow:

This workflow is intended to support research of botnets using real, de-identified data. It provides access to anonymized, pseudonymized, and statistical data exclusively *for* research purposes.

State of this workflow:

The workflow is functional for statistical data. Provision of de-identified reports is planned as a future extension of the workflow.

Usage restrictions on the received data:

The data received via this workflow SHALL be used under standard research terms including citation of ACDC as a data source. These data sets MUST NOT be made available for commercial purposes but MAY be included in research works and MAY be made available to third parties for validation purposes.

Identification of the stakeholder:

The research stakeholder is identified by means of an API Key and token issued by the CCH through the Community Platform.

This workflow does not transfer personal data. It uses the basic identification for stakeholders.

Scope of the data retrievable by this workflow:

The stakeholder following this workflow may receive statistical data on the reports in the CCH. The stakeholder may receive data sets without any personal data as well as any data sets where all included personal data is anonymized or pseudonymized.

Data formats or schemata received when using this workflow:

Statistical data is transferred using the report schemata defined in Section 8. De-identified data uses the report schemata defined in Section 7 using only the anonymized or pseudonymized version of the fields.

Textual description of the workflow:

To retrieve data using this workflow, a stakeholder has to join ACDC with the Cyber Positioning research and obtain an API key to access the CCH.

11. Conclusion

This document analyses data formats being in use by the ACDC project partners and defines a new set of simple formats to represent reports submitted to and consumed from the CCH together with corresponding workflows. To assess the employed data formats a survey has been conducted using a questionnaire distributed to all ACDC partners. The resulting set of formats comprises 15 data formats that can be partitioned into 13 textual and 2 binary formats. From these formats, 10 have a formal definition of their structure and data types which allows an automatic processing. The list of data formats contains some well-known formats including IODEF, sFlow, and X-ARF. To manage the diversity of data formats and reduce the interpretation work of data consumers, a new set of light-weight data formats is defined that covers all types of data submitted to the CCH by project partners while being flexible enough to preserve complex data in other formats for consumers that can process these. The new set of data formats is complemented by a set of workflows formalising the interactions with the CCH depending on the type of data as well as the role of the party communicating with the CCH.

In the first part of this document a set of user groups and general technical and organisational requirements has been assessed that are relevant for data formats. In Section 5.1, these requirements have been assigned to use cases that are relevant for the project. For example, such a use case is the submission of IDS sensor data to the CCH. Furthermore, the basic requirements that are crucial for these use cases have been stated.

From this, the data formats resulting from the survey of ACDC partners are assigned to the individual use cases they support. It turns out that the available data formats already cover the use cases quite well. X-ARF is the most versatile data format. Its strength is the data exchange with external sites such as CSIRTs, ISPs, and law enforcement. IODEF is specialised towards the data exchange between CSIRTs. However, the format is more complex and requires much more effort for processing. Because of the features especially addressed to the CSIRT community, its strength is the data exchange with selected partners that are capable of handling the format. The same is true for STIX/TAXII. This format provides powerful means to model complex threats such as botnets while being still more complex to process by the communication partners.

Another important requirement is anonymization and access control of data. Anonymization requires a data format in which all data types are specified. This enables to identify all data fields containing data to be anonymized. IODEF, X-ARF and STIX satisfy this requirement. A shortcoming of all data formats is the lack of access control.

Based on the identified requirements, the analysis of the data formats in use, and the use cases, the second part of the document defines a new set of data formats together with corresponding workflows for interaction with the CCH. The set of formats is logically split into a format for submission and notification of single reports, a format for aggregation and metrics, and a format for end customer notification. While the two former formats are based on JSON, the latter format uses X-ARF due to the convenient delivery by email and the human-readable description towards a probably less technical audience not knowing ACDC and the defined report format. The main goal in designing these formats was to unify the data exchanged with the CCH. This provides parties submitting data to the CCH with a guide on what information to send while substantially lowering the interpretation work for the party receiving the data. During the project, the formats have been used in the contexts of WP3 and WP4. This usage revealed a set of possible improvements, which have been included in updated versions of the formats.

Complementary to the data formats, a scalable data sharing solution requires formalised workflows to interact with the CCH. While mutual sharing policies between individual parties are generally possible, workflows were defined that depend on the type of data and role of the party interacting with the CCH.

In the case of CERTs, this results in an integration with the Trusted Introducer Service (TI). Each CERT accredited by TI can obtain X-ARF reports on its constituency with TI certifying the IP ranges and ASNs the CERT is responsible for. No registration with the CP is necessary and no infrastructure to interact with the CCH needs to be set up.

While addressing ISPs and network owners follows a similar workflow design by using an external data source (RIPE) for verification, additional workflows broaden the range of data receivers by deidentification of the shared data. In that way, statistical data for research as well as data correlation can be obtained without the need for mutual sharing agreements between individual parties.

To summarize, each party interacting with the CCH as seen in Figure 6 has been addressed. While data sharing throughout the project duration and between the contracting partners has also been possible by other measures like direct sharing or individual configurations of the CCH, a viable solution has been created and evaluated in WP3 and WP4 which allows data to be shared between participating parties of an ongoing ACDC solution. In addition, CERTs and network owners can receive helpful data related to them, without the need for establishing mutual and individual sharing policies. This leads to a lowered barrier for incident sharing and an enhancement of the cybersecurity situation throughout Europe.

12. References

D1.2.2 ECO, ACDC Deliverable D1.2.2: Specification of Tool Group "Centralised Data Clearing House", 2015

D1.4.2 Fraunhofer FKIE, ACDC Deliverable D1.4.2: Specification of Tool Group "Malicious or Vulnerable Websites", 2015

D1.8.1 KU Leuven, B-CCENTRE, ACDC Deliverable D1.8.1: Legal Requirements, 2014

D3.3 INCIBE, ACDC Deliverable D3.3: Control reports, monitoring, timetables, agenda & experiment warnings, 2015

D4.4 ATOS, ACDC Deliverable D4.4: Publicly Accessible Database of Botnet Metrics, 2015

D6.1.1 Engineering Ingegneria Informatica, ACDC Deliverable D6.1.1: User profiles and categorization, 2013

D6.1.2 Engineering Ingegneria Informatica, ACDC Deliverable D6.1.2: Identified user list across the different selected organisations, 2013

D6.2.1 Engineering Ingegneria Informatica, ACDC Deliverable D6.2.1: ACDC Social Platform deployed, Online platform, 2014

DoW ECO, ACDC Description of Work, 2013

NISTIR 7695 B. A. Cheikes, D. Waltermire, K. Scarfone, NIST Interagency Report 7695: Common Platform Enumeration: Naming Specification Version 2.3, 2011

RFC 2119 S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, 1997

RFC 3176 P. Phaal, S. Panchen, N. McKee, InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, 2001

RFC 3954 B. Claise, Ed., Cisco Systems NetFlow Services Export Version 9, 2004

RFC 4765 H. Debar, D. Curry, B. Feinstein, The Intrusion Detection Message Exchange Format (IDMEF), 2007

RFC 5070 R. Danyliw, J. Meijer, Y. Demchenko, The Incident Object Description Exchange Format, 2007

RFC 5655 B. Trammell, E. Boschi, L. Mark, T. Zseby, A. Wagner, Specification of the IP Flow Information Export (IPFIX) File Format, 2009

RFC 7011 B. Claise, Ed., B. Trammell, Ed., P. Aitken, Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, 2013

A. Data Format Survey

For reference we first list the complete questionnaire that was sent to the project consortium and afterwards the filled questionnaires with abbreviated questions. To better distinguish the questions from the individual answers, the questions are presented in a different colour.

A.1. Questionnaire

1. What is the name of the data format and which version is currently in use?
2. Specific use case:
   Use cases: Please describe the specific use case or cases regarding the data exchange format. This includes the following points:
   (a) What is your role and the role of other participating sites? Why do you use the specific format?
   (b) Which workflows with respect to the import, exchange, and export of the data are involved?
   (c) Which productive software components and interfaces are used?
   (d) What are your experiences? Are there any points in the format you want to improve or are any features missing?
   (e) If available, please submit any samples.
   (f) Are there any licenses or patents that have to be considered concerning the application of the data format?
3. Data format details:
   Please provide us with technical details concerning the data format(s)
   (a) Is there a binding to a specific Internet protocol for the transport?
   (b) How is the data format structured or specified?
      i. Is there a formal specification of the structure (e.g. to be machine understandable)?
      ii. Is the specification publicly available? Where are they published? Are there any standards or RFCs released providing a specification?
      iii. Is it possible to extend the format?
      iv. Is it possible to validate the correctness of the message syntax/semantics?
      v. How is the message represented (textual, binary, or other)?
   (c) Please, describe the type of data or threat for which the format is designed.
   (d) Which security aspects are implemented by the data format and its related transport protocols?
      i. Confidentiality and integrity?
      ii. Sender and recipient authentication?
      iii. Availability?
   (e) Is the format adapted to the provisions of a targeted user group? If not, what are the addressed user communities (e.g. end user, CERT, ISP).
   (f) Is a specific communication infrastructure preferred?
      i. Peer to peer?
      ii. Centralised?
      iii. Closed user group?
   (g) Which software components to produce, import, export, parse, and process the data are available? Are these publicly released? Are there any licenses or patents related to the software that have to be considered?

A.2. *Questionnaire A*

1. Format name and version

Our tool Evidence Seeker helps operators to extract evidences from a log file.

Input: Offline

- Plain text files, generally log files

Output:

- Plain text files. There are generated 2 files, one with contact information and the other one with evidences

2. Use case

(a) Role and rationale

Evidence Seeker helps operators to extract evidences from a log file.

Using log files as input has the key advantage that Evidence Seeker can receive the data as is, in the format that is generated by the application that creates the log, without any previous manipulation.

(b) Workflows

As this tool will be part of the Centralised Data Clearing House, Evidence Seeker can be used in any workflow where there is a need to process a log file searching for IPs suspicious of have been compromised

(c) Software components and interfaces

The tool doesn’t use any productive software component or interface.

(d) Experiences

Plain text is a good option for evidence extraction. As logs are usually generated in plain text, there is no need to parse the log. Other plain text advantage is that is easily understandable and universally accepted in any operating system.

(e) Samples

(f) Licenses or patents

As the input and output are in plain text, there is not bound to any licenses or patents.

3. Format details

(a) Transport protocol

Evidence Seeker doesn’t interact with any other tool, so there is no binding to any specific Internet protocol for the transport.

(b) Structure or specification

i. Formal specification

The input and output structures follow INTECO specifications, but these specifications do not necessarily adhere to any standard.

In order to satisfactorily process the IPs, the input file must have at the beginning of each line the IP in numeric format.

ii. Availability of specification

The specification follows INTECO defined structure, but it does not necessary follow any standard.

iii. Extending the format

If it is wanted to extend the format it must be taken into consideration the purpose of Evidence Seeker.

iv. Validate syntax and semantics

The input is generally log files, so they follow a structured syntax that makes possible to validate the input

The same happens with the output, it follows a structured syntax that makes possible to validate it.

v. Representation

Information is saved textually in plain text.

(c) Type of data or threat

Input data are log files, designed for log event recording in a system.

Output data is designed to group IPs detected in the log file, in a structured way.

(d) Security aspects

i. Confidentiality and integrity

No, the output is in plain text without any kind of encryption or security measures

ii. Authentication

Output is generated without any consideration about the recipient.

But security restrictions can be implemented at OS level into folders where the files are expected to be saved.

iii. Availability

Both input and output are files, that means that information is stored into file system and can be accessed when desired.

(e) User group

Evidence Seeker is designed to facilitate the notification process obtaining evidences from a log file, so the tool is basically thought for CERTs.

(f) Communication infrastructure

Evidence Seeker currently doesn’t coordinate with any other tool or service, so there is no specific communication infrastructure.

i. Peer-to-peer

ii. Centralised

iii. Closed user group

(g) Software components

EvidenceSeeker is offered with all the components it needs to handle information.

A.3. Questionnaire B

1. Format name and version

Flux-Detect detects and monitors domains using fast-flux techniques.

INPUT:

- Plain text files, with domains lists

OUTPUT: it doesn’t generate any output; it saves information in databases.

2. Use case

(a) Role and rationale

The role of Flux-Detect is to give feedback about domains determining if they are fast-flux or not.

(b) Workflows

Flux-Detect returns feedback about if a domain is fast-flux or not. So, both the input and output are used or can be used in conjunction with other ACDC tools.

(c) Software components and interfaces

There are no productive software components or interfaces used.

(d) Experiences

Input and output follows our needs. Since Flux-Detect fulfils INTECO expectations, we do not plan further improvements right now, but input and/or output can be adapted if needed in order to integrate it with other ACDC tools.

(e) Samples

INPUT

```
begin
google.es
google.com
end
```

(f) Licenses or patents

The input is in plain text so there is no need of licenses or patents considerations. As there is no output (information is saved in a data base) there is neither any need of licenses or patents considerations for output.

3. Format details

(a) Transport protocol

Whois port 43 of TCP.

(b) Structure or specification

i. Format specification

The input structure follows an INTECO specification, but the specification does not necessarily adhere to any standard.

The input is a plain text file that has a list of domains to check, each of them in a different line.

ii. Availability of specification

The input structure follows INTECO specifications, but these specifications do not necessarily adhere to any standard.

iii. Extending the format

Flux-Detect works only with domains, so in the way the program is designed, there is no necessity to extend the format. But it is possible to extend it if necessary.

iv. Validate syntax and semantics

As the files received must follow the structured defined, it is perfectly possible to validate the correctness of the input files.

v. Representation

INTPUT:

- The message is always represented textually

OUTPUT:

- Information is saved in SQL databases

(c) Type of data or threat

INPUT: Flux-Detect receives web domains to check if they are fast-flux or not

OUTPUT: Flux-Detect determines if a domain is fast-flux or not and saves the information in databases.

(d) Security aspects

i. Confidentiality and integrity

Flux-Detect does not implement any encryption or data security measures. But, because there is no output as information is saved in databases, it would be possible to implement security restrictions in the database.

ii. Authentication

There are no sender or recipient authentication implementations

iii. Availability

As information is saved in databases, it can be said that it is always available.

(e) User group

There is no output format as data are saved in databases. Although the output format is not adapted to any specific targeted group, the information may be suitable for different user communities, for example, it may be suitable for statistical purposes, for CERTs in order to know if a domain is fast-flux or not, etc.

(f) Communication infrastructure

The information generated by Flux-Detect, nowadays, is not publicly spread, but only provided by a web interface to operators carrying out domain security investigation duties

i. Peer to peer

ii. Centralised

iii. Closed user group

(g) Software components

Flux-Detect is offered with all the components it needs to handle information.

A.4. *Questionnaire C*

1. Format name and version

Skanna, checks the security level of several domains. For each domain checks several parameters as the software installed and version.

Skanna performs the following actions:

1. Gathering of the domains to check
2. Domains information gathering and analysis

1: Domains to check gathering

The list of domains to check can be obtained in different ways:

Source	Description	Data obtained	Method for obtaining data
Nic.es	Entity responsible for the .es domains management	.es domains list registered since 2007	Download of the published PDF file (http), parser and domains extraction
VeriSign	Obtaining of all DNS zones of the .com, .net and .name TLD	The new domains .com, .net and .name registered daily	Reception of a file with the domains, one domain per line.
Manually	Used for re-scanning	Domains to check	An operator indicates manually the domain to check

2: Domains information gathering and analysis

For each domain obtained in previous step, the following actions are performed:

- Obtain the index page source code of the website
- Identify the software and technologies used by the website
- Indexation of the index page source code
- Antivirus analysis of the downloaded source code, in order to identify malware or compromise signals

To obtain information about each domain it is used WhatWeb (http://www.morningstarsecurity.com/research/whatweb), the information received from this source is processed by Skanna in order to save it in databases.

The input and output are as follows:

- **INPUT**: domains input
- **OUTPUT**: there is no output. Information is saved in databases

2. Use case

(a) Role and rationale

Currently Skanna doesn't interact with other tools and/or services of ACDC. But, in order to perform its activities, Skanna interacts with different tools and/or services external to ACDC, and this interaction is always done requesting information to these tools/services.

(b) Workflows

The workflow is always the same; Skanna needs some information and send a request to the tool/service needed in each moment.

(c) Software components and interfaces

Skanna interacts with the following tools/services that are not part of the ACDC project:

1. Nic.es: Skanna downloads a PDF file with the new domains
2. WhatWeb

(d) Experiences

Both input and output follow our specifications. Since Skanna fulfils INTECO expectations, we do not plan further improvements right now, but input and/or output can be adapted if needed in order to integrate Whois with other ACDC tools.

(e) Samples

Attached at the end of the questionnaire

(f) Licenses or patents

Skanna doesn’t use specifically any data format as it only gathers and process structured information.

The only consideration is that WhatWeb has GPLv2 license.

3. Format details

(a) Transport protocol

The download of the PDF file from Nic.es is done by HTTP.

The interactions with the other tools/services are done locally, thus there is no information transmission on the internet.

(b) Structure or specification

i. Format specification

Input data is obtained from different sources and the format specification is defined by each source.

There is no output, as data are saved in different databases.

ii. Availability of specification

Input data from VeriSign is a structured file with a domain list, one per line.

The information gathered from the different sources is structured. The output from WhatWeb is in XML and follows XML standards.

There is no output, but only information saved in databases.

iii. Extending the format

It would be possible to gather information from other sources or use other formats but it would be necessary to adapt Skanna in order to make it able to perform those new actions.

iv. Validate syntax and semantics

As all the data received are in a structured format, it would be possible to check the different inputs using regular expressions, but nowadays there is no specific mechanism to implement this action.

The output received from WhatWeb is in XML so it would be especially easy to validate the correctness of the information.

v. Representation

The information is presented textually, but part of the data is saved in a DataBase following SQL specifications.

(c) Type of data or threat

Skanna is designed to get a map about the security level of the domains inspected.

(d) Security aspects

i. Confidentiality and integrity

Skanna does not perform any confidentially or integrity checks.

There is no output as information is saved in databases, but it would be possible to implement security restrictions in the database.

ii. Authentication

Skanna does not perform any recipient authentication

iii. Availability

The information is saved in database, so it is available when needed.

(e) User group

Skanna is a tool that aggregates relevant information about domains (technologies inventory used by the domains, domain index and domain index analysis by an AV). This information is provided by a web interface to operators carrying out domain security investigation duties, allowing operator to check, search or exploit the information.

(f) Communication infrastructure

The information gathered by Skanna, nowadays, is not publicly spread, but only provided by a web interface to operators carrying out domain security investigation duties

i. Peer to peer

ii. Centralised

iii. Closed user group

(g) Software components

It is very important to note that Skanna interacts with other external tools/services needed for a proper performance of Skanna.

Nic.es

Example of the websites for April: http://www.dominios.es/dominios/sites/default/files/files/Altas%20abril%202013%20%28espanol%29.pdf

VeriSign

```
Domain 1
Domain 2
…
```

WhatWeb XML example

```
<?xml version="1.0"?><?xml-stylesheet type="text/xml" href="whatweb.xsl"?>
<log>
<target>
        <uri>http://www.osi.es</uri>
        <http-status>200</http-status>
        <plugin>
                <name>HTTPServer</name>
                <string>Apache</string>
        </plugin>
        <plugin>
                <name>Google-Analytics</name>
                <account>UA-17786431-4</account>
        </plugin>
        <plugin>
                <name>Apache</name>
        </plugin>
        <plugin>
                <name>IP</name>
                <string>195.235.9.101</string>
        </plugin>
        <plugin>
                <name>JQuery</name>
        </plugin>
        <plugin>
                <name>HTTP-Headers</name>
                <string>cache-control: store, no-cache, must-revalidate, post-check=0,
pre-check=0,connection: close,content-length: 64612,content-type: text/html;
charset=utf-8,date: Tue, 14 Feb 2012 09:02:53 GMT,expires: Sun, 19 Nov 1978
05:00:00 GMT,last-modified: Tue, 14 Feb 2012 09:02:56 GMT,server: Apache,set-
cookie: SESS66c3c803e511690dab0e8d70f3f0cf31=oqdkjeqlv4lun5ntlprk0ut6b0;
expires=Thu, 08-Mar-2012 12:36:13 GMT; path=/; domain=.osi.es,vary: Accept-
Encoding</string>
        </plugin>
        <plugin>
                <name>Drupal</name>
        </plugin>
        <plugin>
                <name>MD5</name>
                <string>b2c6f30f1355d0482e04ad869a7bd68b</string>
        </plugin>
        <plugin>
                <name>Cookies</name>
                <string>SESS66c3c803e511690dab0e8d70f3f0cf31</string>
        </plugin>
        <plugin>
                <name>Title</name>
                <string>Oficina de Seguridad del Internauta</string>
        </plugin>
        <plugin>
                <name>Country</name>
                <string>SPAIN</string>
                <module>ES</module>
        </plugin>
```

```
</target>
</log>
```

A.5. *Questionnaire D*

1. Format name and version

Whois automates relevant IPs lookup. This service provides whois information in an efficient and easily parseable manner.

Input:

- A single IP
- Plain text files

Output:

- Text

2. Use case

(a) Role and rationale

The role of Whois is to provide whois information. Thus, it works as a service for other ACDC tools that need that information.

Whois output information can easily be read by humans or processed by a machine.

(b) Workflows

Whois runs in service mode (receives a request and returns an answer).

(c) Software components and interfaces

Whois queries different RIR services in order to obtain and/or update information.

It also uses a free database of Maxmind in order to obtain the country an IP belongs to.

(d) Experiences

Both input and output follows our specifications. Since Whois fulfils INTECO expectations, we do not plan further improvements right now, but input and/or output can be adapted if needed in order to integrate Whois with other ACDC tools.

(e) Samples

INPUT

- Plain text file with IPs

```
begin
verbose
8.8.8.8
193.245.3.4
end
```

OUTPUT

- IPS text response

```
15169 | 8.8.8.8 | 8.8.8.0/24 | US | Arin | cfg:soc@us-cert.gov cpg:phishing-
report@us-cert.gov r:arin-contact@google.com r:axelrod@google.com r:ir-contact-
netops-corp@google.com r:kk@google.com | GOOGLE - Google Inc.
6848 | 193.245.3.4 | 193.244.0.0/15 | BE | Ripe | cg:cert@belnet.be cg:cert@cert.be
r:frank.terlinck@kbc.be | TELENET-AS Telenet N.V.
```

(f) Licenses or patents

Both input and output are textual, and it is also possible to receive the input in a file in plain text, so there is no need of license or patents considerations.

3. Format details

(a) Transport protocol

It receives input files through Whois port (TCP 43)

(b) Structure or specification

INPUT

- IPs file

The file must be created according to the following format:

First line: `begin`
Second line: *parameters*
IP addresses, *one per line*
Last line: end

OUTPUT

- IPS query.

The output in response to an IP query is as follows:

- AS (Autonomous System) Number
- IP address
- CIDR (Classless Inter-Domain Routing)
- Country code
- RIR (Regional Internet Registry) the IP belongs to
- IP contacts, with different TAGS
- AS (Autonomous System) Name

i. Format specification

The input and output structures follow INTECO specifications, but the specifications do not necessarily adhere to any standard.

Our own specification is the one showed before, the input is a single IP or a file in plain text, and the output are several fields separated by the character |

ii. Availability of specification

The input and output structures follow INTECO specifications, but the specifications do not necessarily adhere to any standard.

iii. Extending the format

If it is wanted to extend the format, it must be taken into consideration the purpose of Whois and that the only data that it receives are IPs.

iv. Validate syntax and semantics

As the input and output follow a strict syntax, yes, it would be possible to validate the correctness of the message

v. Representation

The output is represented textually.

(c) Type of data or threat

Output is designed to provide contact information about an IP. The output has several fields, each of them with different information about the IP, but arranged following a fixed structured showed before, in order to be easily understandable.

This information can be obtained either by an operator through command line interface or by another program.

(d) Security aspects

i. Confidentiality and integrity

Neither the data format nor its related transport protocols support any security measures

ii. Authentication

Whois may check the origin of the query (will check the IP) and return more or less information depending on the questioner

iii. Availability

There is no special protection for availability

(e) User group

Whois is suitable for any user/program, but basically focused on CERTs, that need to know the contact data for IPs.

(f) Communication infrastructure

Whois currently doesn’t coordinate with any other tool or service, so there is no specific communication infrastructure.

i. Peer to peer

ii. Centralised

iii. Closed user group

(g) Software components

Whois is offered with all the components it needs to handle information.

A.6. Questionnaire E

1. Format name and version

Suricata engine is being used as a NIDS engine on a wireless AP, which is used as a gateway for mobile devices. The NIDS engine allows us to monitor and analyse network traffic of mobile devices running over wireless AP. The traffic can be captured in PCAP format and, moreover, off-line (almost realtime) analysis of PCAP files is possible. Additionally, logging to database with possibility of e-mail notifications is also possible. There are multiple possible log outputs (configurable):

- Line based alerts log (fast.log)
- Log output for use with Barnyard (unified.log)
- Alert output for use with Barnyard (unified.alert)
- Packet log (pcap-log)
- Files log (json format)

For us most important is **files-json.log** which holds data for every single file that crossed your http pipe. Using additional **fuse** file-system library (e.g. ClamAV[1]) we can integrate other tools for further analysis of the traffic captured.

2. Use case

(a) Role and rationale

We use the described formats in order to easily analyse the output from the NIDS and import it into HBase database running on Hadoop. Additional analytics can be done over HBase database (for further big-data analytics).

(b) Workflows

The output from NIDS can be transformed from CSV or JSON string formats practically into any format data (e.g. TAB delimited format) that is needed to be transported to other module in the workflow. We are using Flume to scan **/tmp/logs** directory for parsed **files-json.log** files and stores them into HBase database for further analysis.

(c) Software components and interfaces

Suricata engine, Flume as a transportation level of captured data in HBase; data in HBase is ready for further analysis. Google Cloud Messaging is used to push messages towards mobile clients.

(d) Experiences

Not yet clear about missing features. Mow, we are able to detect some anomalies (e.g. possible scans from mobile devices, detection of downloading of malware software – with the use of third-party tool for analysing the malware content of downloaded packages).

(e) Samples

An example of package detection while downloading specific package from the Android marketplace.

```
{
  "timestamp": "04\/25\/2013-10:01:22.552241",
  "ipver": 4,
  "srcip": "173.194.70.100",
  "dstip": "172.16.118.69",
  "protocol": 6,
  "sp": 80,
  "dp": 54356,
  "http_uri": "\/market\/download\/Download?
packageName=com.overlook.android.fing&versionCode=210&token=AOTCm0QMRhNQIC-VmjtrRg-
uK3lCqs-
g4kqRcfv4Mp40sMxtyZ4B9I0X1_ksrJbGpNyz3PIwGJWPUDcbaTSc6JUz28gTuDkp5srwtfV5vf0&downlo
adId=1108987573357907795",
  "http_host": "android.clients.google.com",
  "http_referer": "<unknown>",
  "http_user_agent": "AndroidDownloadManager\/4.2.2 (Linux; U; Android 4.2.2;
Galaxy Nexus Build\/JDQ39)",
  "filename": "\/market\/download\/Download",
  "magic": "unknown",
```

[1] http://www.clamav.net/lang/en/

```
  "state": "CLOSED",
  "stored": false,
  "size": 572
}
```

An example of PDF file detection while downloading the file using mobile device.

```
{
  "id": 8,
  "timestamp": "05\/08\/2013-13:50:21.732132",
  "ipver": 4,
  "srcip": "173.1.226.155",
  "dstip": "192.168.14.201",
  "protocol": 6,
  "sp": 80,
  "dp": 47101,
  "http_uri": "\/pdfs\/PrimoPDF_V5_User_Guide.pdf",
  "http_host": "www.primopdf.com",
  "http_referer": "http:\/\/www.google.si\/search?
q=pdf+manual&ei=MS6KUamIM8m1PM6zgMgF&start=10&sa=N&biw=360&bih=567",
  "http_user_agent": "Mozilla\/5.0 (Linux; U; Android 4.2.2; en-us; Galaxy Nexus
Build\/JDQ39) AppleWebKit\/534.30 (KHTML, like Gecko) Version\/4.0 Mobile
Safari\/534.30",
  "filename": "\/pdfs\/PrimoPDF_V5_User_Guide.pdf",
  "magic": "PDF document, version 1.4",
  "state": "UNKNOWN",
  "stored": true,
  "size": 25736
}
```

(f) Licenses or patents

No. Format is JSON and a result of an open source engine.

3. Format details

(a) Transport protocol

No. However, HTTP is usually used with JSON.

(b) Structure or specification

i. Format specification

The format is in JSON and there is no formal specification of the data format. However, we are providing informal data format here.

```
{
"timestamp": <time stamp>,
"ipver": <ip version>,
"srcip": <source IP>,
"dstip": destination IP>,
"protocol": <protocol id - 6-TCP>[1],
"sp": <source port>,
"dp": <destination port>,
"http_uri": <uri part after http_host>,
"http_host": <host>,
"http_referer": <link from which source accessed the destination>,
"magic": <file command's magic pattern file>[2],
"state": "CLOSED",
"md5": <md5 hash of the file>,
"stored": <was the file stored on file system>,
"size": <file size>
}
```

ii. Availability of specification

There is no formal specification of the output format. However, the input (

[1] http://en.wikipedia.org/wiki/List_of_IP_protocol_numbers

[2] Same output as “file” or “magic” command

iii. Extending the format

The format can be extended using plugins or addons after the log has been created.

iv. Validate syntax and semantics

It can be validated with a simple JSON validation program or script.

v. Representation

It is represented as text.

(c) Type of data or threat

The format is designed to describe every single file that crosses configured HTTP pipe and is (can be) captured by Suricata’s engine.

(d) Security aspects

i. Confidentiality and integrity

It is core data format and is not being exchanged with external components (yet). It is used by the component of the framework for mobile devices security.

ii. Authentication

None security aspects are implemented by the data format in this aspect.

iii. Availability

None security aspects are implemented by the data format in this aspect.

(e) User group

The format is not adapted to the provisions of a target user groups. The format presents the foundation of the information produced by additional analysis tools taking into account data captured within **files-json**. The results of the analysis are sent to CERTs and possibly ISP for further analysis. We are not using any other format for exchanging information with external entities.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

Preferred communication infrastructure is centralized since we need central endpoint to aggregate information from the **files-json**. However, the architecture of the system under the aggregation end-point can be designed to be highly available and distributed.

(g) Software components

There is plethora of open-source tools available for processing the JSON data format (python libraries, libraries for java). All are publicly available and easily extensible. There are no licences or patents related to the software. The use of custom created script with MySQL or PostgreSQL import (bulk or continuous) or importing it directly to MongoDB (native import of JSON files) are already available on the web page of Suricata[1]. As already described, Apache Flume[2] framework can be used to import output (files-json) into big-data framework for further analitics.

[1] https://redmine.openinfosecfoundation.org/projects/suricata/wiki/What_to_do_with_files-jsonlog_output

[2] http://flume.apache.org/FlumeDeveloperGuide.html

A.7. Questionnaire F

1. Format name and version

JSON

2. Use case

JSON is a general-purpose data format to exchange information between two entities.

(a) Role and rationale

We run a set of different Honeypots and additional passive sensors. The gathered information of all sensors has to be correlated into a single report representing individual incidents. In order to not lose any information, the utilized data format needs to be able to hold all information generated by the used set of tools. Since we cannot forecast the future and anticipate any future information that might be generated by updated or new tools, the data format has to be flexible to hold arbitrary future data as well. As a result, we opted for JSON, which is fully flexible, human and machine-readable, produces only little overhead and is out-of-the box supported by major programming languages.

(b) Workflows

A set of different tools generates individual JSON reports that are sent to a correlation server. This server correlates all reports belonging to the same incident and forwards the information to subscribed clients. One of these clients stores generated reports in a NoSQL database (MongoDB) that also handles JSON natively.

(c) Software components and interfaces

We use internal implementations to generate JSON reports from particular honeypots and passive sensors. These include p0f, snort, dionaea, glaspot and kippo. Sensors developed by us support JSON natively. No additional software is required for parsing JSON messages in python, for java we use Jackson.

(d) Experiences

Like any other text-based reporting format, JSON is rather inefficient for transmitting binary data since it has to be encoded (e.g. base64).

(e) Samples

Attached below.

(f) Licenses or patents

No

3. Format details

(a) Transport protocol

No. JSON can be transmitted by using arbitrary transport protocols.

(b) Structure or specification

i. Format specification

Yes. http://tools.ietf.org/html/rfc4627

ii. Availability of specification

Yes. http://tools.ietf.org/html/rfc4627

iii. Extending the format

According to the RFC “A JSON parser MAY accept non-JSON forms or extensions.”. Anyway, this would rather be an exception since it is generally not necessary to extend the format itself.

iv. Validate syntax and semantics

Yes. This is done by default libraries of many programming languages and can be done by various other tools.

v. Representation

Pure textual.

(c) Type of data or threat

JSON is not threat-bound. It is used for arbitrary data and was originally designed to represent JavaScript objects.

(d) Security aspects

i. Confidentiality and integrity

None. Security aspects have to be implemented by underlying transport protocols, like SSL.

ii. Authentication

None. Security aspects have to be implemented by underlying transport protocols, like SSL.

iii. Availability

None. Security aspects have to be implemented by underlying transport protocols, like SSL.

(e) User group

The format is general-purpose.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

Communication infrastructure solely depends on the underlying transport protocol, which is completely independent from JSON.

(g) Software components

An extensive list of software components can be found here: http://www.json.org/index.html (you need to scroll down a little bit)

Sample JSON message

```
{
    "endtime": {
        "$date": 1368970600734
    },
    "whois": {
        "cc": "RU",
        "owner": "MORDOVIA-AS OJSC Rostelecom",
        "BGP_prefix": "87.119.224.0/19",
        "asn": 34449
    },
    "geoip": {
        "city": "Saransk",
        "region_name": "Mordovia",
        "region": "46",
        "area_code": 0,
        "time_zone": "Europe/Samara",
        "longitude": 45.18330001831055,
        "metro_code": 0,
        "country_code3": "RUS",
        "latitude": 54.18330001831055,
        "postal_code": null,
        "dma_code": 0,
        "country_code": "RU",
        "country_name": "Russian Federation"
    },
    "remotehost": "XX.XX.XX.XX",
    "connections": [
        {
            "connection_type": "accept",
            "remoteport": 2001,
            "p0f_profile": {
                "uptime": "-1",
                "dist": "15",
                "fw": "0",
                "tos": "",
                "detail": "2000 SP4, XP SP1+",
                "link": "IPv6/IPIP",
                "nat": "0",
                "genre": "Windows"
            },
            "protocol": "smbd",
            "localport": 445,
            "starttime": {
                "$date": 1368970592858
            },
            "endtime": {
                "$date": 1368970593438
            },
            "transport": "tcp"
        },
        {
            "remoteport": 2013,
            "endtime": {
                "$date": 1368970595108
            },
            "localport": 139,
            "starttime": {
                "$date": 1368970594042
            }
        },
        {
            "connection_type": "accept",
            "remoteport": 2004,
            "p0f_profile": {
                "uptime": "-1",
                "dist": "15",
                "fw": "0",
                "tos": "",
                "detail": "2000 SP4, XP SP1+",
                "link": "IPv6/IPIP",
                "nat": "0",
                "genre": "Windows"
            },
            "protocol": "smbd",
            "localport": 445,
            "downloads": [
                {
                    "peid": {},
                    "virustotal": {
                        "date": 1368904680,
                        "report": {
                            "Microsoft": "Worm:Win32/Gnoewin.A",
                            "Norman": "Inject.AQTC",
                            "Panda": "Suspicious file",
                            "ESET-NOD32": "a variant of Win32/Injector.AFKU",
                            "VBA32": "Worm.VBNA"
                        },
                        "ratio": 5
                    },
                    "url": "https://hotfile.com/dl/223458246/4bd6f53/g1.exe",
                    "md5hash": "0a0375431f8d125bfc12950abd98876e",
                    "peXaminer": {
                        "File Statistics": {
                            "Attributes": {
                                "created": "Sun May 19 13:42:51 2013",
                                "file_name":
"/data/binaries/0a0375431f8d125bfc12950abd98876e",
                                "last_accessed": "Sun May 19 13:42:51 2013",
                                "last_modified": "Sun May 19 13:42:51 2013",
                                "entropy": 7.318794553483753,
                                "file_size": 115633
                            },
                            "Hashes": {
                                "sha256":
"e69a9c7e442adb837f7af1d3a935965623b9d4354d68b66b21b28ae75b430847",
                                "sha512":
"4679cd287cd2ebd62d56e510ac20d88ae04b02966dfad21fcb0090b8421ddc075fac3ca769296b70bd
1fd4f5569a61be5532116ded0fc196508d4e305a58f258",
                                "md5": "0a0375431f8d125bfc12950abd98876e",
                                "sha1": "3d7614ca28f924e459c3e86c0b661021f01c54b1"
                            }
                        },
                        "PE Characteristics": {
                            "Optional Header": {
                                "SectionAlignment": 4096,
                                "SizeOfCode": 35328,
                                "Magic": "32bit",
                                "SizeOfUninitializedData": 0,
                                "MinorSubsystemVersion": 1,
                                "MajorLinkerVersion": 10,
                                "ImageBase": 4194304,
                                "SizeOfInitializedData": 18944,
                                "SizeOfImage": 77824,
                                "NumberOfRvaAndSizes": 16,
                                "FileAlignment": 512,
                                "MajorSubsystemVersion": 5,
                                "CheckSum": {
                                    "given": 116746,
                                    "true": 120800
                                },
                                "Subsystem": "GUI",
                                "MinorLinkerVersion": 0,
                                "AddressOfEntryPoint": 6351,
                                "SizeOfHeaders": 1024
                            },
                            "File Header": {
                                "TimeDateStamp": {
                                    "UTC": "Sat May 18 01:25:29 2013",
                                    "numerical": 1368840329
                                },
                                "Machine": "i386",
                                "Characteristics": [
                                    "Executable Image",
                                    "32bit"
                                ],
                                "NumberOfSymbols": 0,
                                "NumberOfSections": 5,
                                "SizeOfOptionalHeader": 224
                            },
                            "DOS Header": {
                                "e_lfanew": 224
                            },
                            "Sections": [
                                {
                                    "Name": " .text",
                                    "Characteristics": [
                                        "execute",
                                        "read"
                                    ],
                                    "SizeOfRawData": 35328,
                                    "Entropy": 6.5357099216549095,
                                    "VirtualSize": 35192,
                                    "VirtualAddress": 4096,
                                    "PhysicalAddress": 1024,
                                    "md5": "729dfe04aad1c60369dec9455decd4ed"
                                },
                                {
                                    "Name": " .rdata",
                                    "Characteristics": [
                                        "read"
                                    ],
                                    "SizeOfRawData": 9216,
                                    "Entropy": 4.772166382739247,
                                    "VirtualSize": 9128,
                                    "VirtualAddress": 40960,
                                    "PhysicalAddress": 36352,
                                    "md5": "805bc471f9d81754b3780a657d5c2f14"
                                },
                                {
                                    "Name": " .data",
                                    "Characteristics": [
                                        "read"
                                    ],
                                    "SizeOfRawData": 4096,
                                    "Entropy": 2.1265588781733644,
                                    "VirtualSize": 15680,
                                    "VirtualAddress": 53248,
                                    "PhysicalAddress": 45568,
                                    "md5": "34ba24583e66905e5c218214d52df071"
                                },
                                {
                                    "Name": " .rsrc",
                                    "Characteristics": [
                                        "read"
                                    ],
                                    "SizeOfRawData": 2048,
                                    "Entropy": 5.129072542887932,
                                    "VirtualSize": 1764,
                                    "VirtualAddress": 69632,
                                    "PhysicalAddress": 49664,
                                    "md5": "2d470a068fec565e16520f1ffb5f13a4"
                                },
                                {
                                    "Name": " .reloc",
                                    "Characteristics": [
                                        "read"
                                    ],
                                    "SizeOfRawData": 3584,
                                    "Entropy": 4.933846775740402,
                                    "VirtualSize": 3366,
                                    "VirtualAddress": 73728,
                                    "PhysicalAddress": 51712,
                                    "md5": "77c55c138cb3daed098db14f48b15e49"
                                }
                            ],
                            "Data Directories": {
                                "Imports": {
                                    "Descriptors": [
                                        {
                                            "KERNEL32_dll": [
                                                "LockResource",
                                                "LoadResource",
                                                "FindResourceA",
                                                "GetProcAddress",
                                                "GetModuleHandleA",
                                                "Sleep",
                                                "GetCommandLineA",
                                                "HeapSetInformation",
                                                "HeapAlloc",
                                                "SetUnhandledExceptionFilter",
                                                "GetModuleHandleW",
                                                "ExitProcess",
                                                "DecodePointer",
                                                "WriteFile",
                                                "GetStdHandle",
                                                "GetModuleFileNameW",
                                                "GetModuleFileNameA",
                                                "FreeEnvironmentStringsW",
                                                "WideCharToMultiByte",
                                                "GetEnvironmentStringsW",
                                                "SetHandleCount",

"InitializeCriticalSectionAndSpinCount",
                                                "GetFileType",
                                                "GetStartupInfoW",
                                                "DeleteCriticalSection",
                                                "EncodePointer",
                                                "TlsAlloc",
                                                "TlsGetValue",
                                                "TlsSetValue",
                                                "TlsFree",
                                                "InterlockedIncrement",
                                                "SetLastError",
                                                "GetCurrentThreadId",
                                                "GetLastError",
                                                "InterlockedDecrement",
                                                "HeapCreate",
                                                "QueryPerformanceCounter",
                                                "GetTickCount",
                                                "GetCurrentProcessId",
                                                "GetSystemTimeAsFileTime",
                                                "MultiByteToWideChar",
                                                "ReadFile",
                                                "UnhandledExceptionFilter",
                                                "IsDebuggerPresent",
                                                "TerminateProcess",
                                                "GetCurrentProcess",
                                                "EnterCriticalSection",
                                                "LeaveCriticalSection",
                                                "IsProcessorFeaturePresent",
                                                "SetFilePointer",
                                                "RtlUnwind",
                                                "LoadLibraryW",
                                                "HeapFree",
                                                "GetCPInfo",
                                                "GetACP",
                                                "GetOEMCP",
                                                "IsValidCodePage",
                                                "SetStdHandle",
                                                "GetConsoleCP",
                                                "GetConsoleMode",
                                                "FlushFileBuffers",
                                                "CloseHandle",
                                                "CreateFileW",
                                                "HeapSize",
                                                "HeapReAlloc",
                                                "LCMapStringW",
                                                "GetStringTypeW",
                                                "WriteConsoleW",
                                                "SetEndOfFile",
                                                "GetProcessHeap"
                                            ]
                                        }
                                    ],
                                    "NumberOfImports": 70
                                },
                                "Resources": {
                                    "number_of_resources": 3,
                                    "total_size": 0,
                                    "entries": [
                                        {
                                            "type": "",
                                            "sub_entries": 2,
                                            "size": 0
                                        },
                                        {
                                            "type": "RT_VERSION",
                                            "sub_entries": 1,
                                            "size": 0
                                        },
                                        {
                                            "type": "RT_MANIFEST",
                                            "sub_entries": 1,
                                            "size": 0
                                        }
                                    ]
                                }
                            }
                        }
                    },
                    "mime": "PE32 executable for MS Windows (GUI) Intel 80386 32-
bit",
                    "ssdeep":
"3072:gV6BJx9epPREuGO7CERO9dBZiAUW4HnnnshDHV:o6BJx9epP+71ZirxMd1"
                }
            ],
            "smb_profile": {
                "smb_dcerpc_requests": [
                    {
                        "dcerpcrequest_uuid": "4b324fc8-1670-01d3-1278-
5a47bf6ee188",
                        "dcerpcrequest_opnum": 31
                    }
                ],
                "smb_dcerpc_binds": [
                    {
                        "dcerpcbind_uuid": "b3332384-081f-0e95-2c4a-302cc3080783",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "a71e0ebe-6154-e021-9104-5ae423e682d0",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "7f4fdfe9-2be7-4d6b-a5d4-aa3c831503a1",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "d89a50ad-b919-f35c-1c99-4153ad1e6075",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "9f7e2197-9e40-bec9-d7eb-a4b0f137fe95",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "8b52c8fd-cc85-3a74-8b15-29e030cdac16",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "9acbde5b-25e1-7283-1f10-a3a292e73676",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "c0cdf474-2d09-f37f-beb8-73350c065268",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "ea256ce5-8ae1-c21b-4a17-568829eec306",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    },
                    {
                        "dcerpcbind_uuid": "7d705026-884d-af82-7b3d-961deaeb179a",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
```

```
                    },
                    {
                        "dcerpcbind_uuid": "4b324fc8-1670-01d3-1278-5a47bf6ee188",
                        "dcerpcbind_transfersyntax": "8a885d04-1ceb-11c9-9fe8-
08002b104860"
                    }
                ]
            },
            "transport": "tcp",
            "starttime": {
                "$date": 1368970593019
            },
            "download_offers": [
                {
                    "url": "https://hotfile.com/dl/223458246/4bd6f53/g1.exe"
                }
            ],
            "endtime": {
                "$date": 1368970600734
            },
            "emu_profile": [
                {
                    "return": "0x7df20000",
                    "args": [
                        "urlmon"
                    ],
                    "call": "LoadLibraryA"
                },
                {
                    "return": "0",
                    "args": [
                        "",
                        "https://hotfile.com/dl/223458246/4bd6f53/g1.exe",
                        "20.exe",
                        "0",
                        "0"
                    ],
                    "call": "URLDownloadToFile"
                },
                {
                    "return": "32",
                    "args": [
                        "20.exe",
                        "895"
                    ],
                    "call": "WinExec"
                },
                {
                    "return": "0",
                    "args": [
                        "-1"
                    ],
                    "call": "Sleep"
                }
            ]
        }
    ],
    "flags": [
        "download",
        "dlserver",
        "scan_vertical"
    ],
    "dns": "87-119-XX-XX.saransk.ru.",
    "_id": {
        "$oid": "5198d6db0cf2f2bc1bd5cb34"
    },
    "starttime": {
        "$date": 1368970592858
    },
    "localhost": "XX.XX.XX.XX"
}
```

A.8. Questionnaire G

1. Format name and version

JSON

2. Use case

(a) Role and rationale

We are using JSON format since it is native output from our ACDC components implemented in Python.

Our role is sending bulk reports about malware URLs, C&C, fast-flux domains and spam campaigns. We use it because it allows large quantities of data at once without much overhead. When used over SSL, it is not limited by maximum allowed attachment size of SMTP servers

(b) Workflows

We inport various data formats into our system and export data from local DB in JSON format.

(c) Software components and interfaces

We import several data formats by our components, but the role of mediation server is to normalize all imported data into unique format and this data is then exported to Central Clearing House in JSON format according to our schema. Mediation server and JSON are the only interface to CCH.

(d) Experiences

Experiences with JSON are OK.

We still have not defined how to encode binary samples to be transferred to Central Clearing House

(e) Samples

Samples are in the attachment

(f) Licenses or patents

No

3. Format details

(a) Transport protocol

There is no binding, but our preferred transport is JSON over SSL

(b) Structure or specification

i. Format specification

JSON schema

ii. Availability of specification

No, specification is defined by us.

iii. Extending the format

Yes

iv. Validate syntax and semantics

Yes

v. Representation

Textual

(c) Type of data or threat

List of fast-flux domains, list of malware URLs, list of spambots in spam campaigns, list of IP addresses related to botnet C&C

(d) Security aspects

i. Confidentiality and integrity

Is related to SSL

ii. Authentication

Is related to SSL

iii. Availability

No

(e) User group

User of this data is ACDC project.

(f) Communication infrastructure

i. Peer to peer

No

ii. Centralised

Yes, we are sending data to CCH

iii. Closed user group

No

(g) Software components

Any language supporting JSON. There are no licences or patents

Attacker data and attacking malware(usually exploit(S))

```
"HoneypotAttackersData"={
    "AttackerData": [
        "timestamp": "2013-04-29 14:02:38",
        "attackerIP": "5.34.247.100",
        "srcPort": "58063",
        "dstPort": "80",
        "protocol": "http",
        "countryCode": "None" ,
        "sample": ["902fe4a680a1b42cdba57c551b32c13b", ""]
        "compromisedURL": ["http://Jinn-tech.com/wikka/DinosgVealpr
%3ERecommended+Resource+site%3C/a%3E", ""]
        ]
       }
```

Hosts serving Malware URL, phishing or C&C

```
"CompromisedHostsData"={
    "CompromisedHost": [
        "IP": "62.73.4.10",
        "domain": "heuro-vacances.fr",
        "country": "FR",
        "type":"malware|c&c|phishing"
        "malwareData":[
            {
            "timestamp": "2013-04-30 07:03:42.530230",
            "infectedURLs": ["heuro-vacances.fr/5nW.exe","",""]
            }
            ]
    ]
       }
```

samples

```
"SamplesData"={
    "sample": [
        "timestamp": "2013-04-29 14:02:38",
        "compromisedHost":"url|attachment",
        "source":"spamtrap|honeypot",
        "data":{
            "attackerIP": "5.34.247.100",
            "protocol": "http",
            "countryCodeIP": "None",
            "checksum":"9e3185c2dfed567442cddf466f20f9a0"
            }
    ]
}
```

Passive DNS replication(fast flux domains)

```
"pDNSData" = {
       "domains": [
            { "domain" : {
```

```
                                 "domain_name": "example.ru",
                                 "time_first": "2012-01-10 16:45",
                                 "time_last": "2012-01-10 16:45",
                                 "ips": [["IP":["121.454.32.23", "198.193.53.141"],
"timestamp": "2012-01-10 16:45:00 UTC"],
                                       ["IP":["132.123.193.23", "198.193.46.1"],
"timestamp": "2012-01-10 16:55:00 UTC"]]
                                 }}
        ]
}
```

Spambots participating in detected spam campaigns

```
 "spamtrapCampaigns"={
      "campaign":[{
             "startTimestamp":"2012-01-10 16:45",
             "endTimestamp":"2012-01-12 19:45",
             "total_spams":"22",
             "spamSubject":"Teik it or leave it",
             "has_malware":"1",
        "spambot":[
            { "ip":"127.0.0.1"
                "asn":"2108"
                "timestamp":"2012-01-10 16:45",
            }]
             }
      ]
}
```

A.9. Questionnaire H

1. Format name and version

Out tool, MMT, allows monitoring and analysing network traffic and any structured data (logs, business activity, messages...). It is composed of several modules: **MMT_Core** that does data extraction (e.g., using DPI); **MMT_QoS/QoE** that does performance analysis; and, **MMT_Security** that analyses data to detect anomalous behaviour. Functionality can be extended via plugins.

Input (offline, i.e., reading a file, or online, listening to a network interface)

1. PCAP v2.4
2. Any structured data (by writing a plugin)

Output:

1. CSV
2. Database tables (e.g., PostgresSQL)
3. XML
4. Any format (by adapting the main)

2. Use case

(a) Role and rationale

The role of our tool is to analyse traffic data to recuperate information that can be useful for detecting botnets and other abnormal or malicious behaviour. The tool can be installed anywhere to analyse network interfaces to generate reports (e.g., alarms or messages) that can be sent to any stakeholder using any means (HTTP/RESTful, emails, SQL data...). The reports contain information that can be read by humans or processed by a machine.

(b) Workflows

The tool that can be used as part of any workflow where there is a need to analyse structured data or communication protocol exchanges and generate reports.

(c) Software components and interfaces

Modules developed by us that use the PCAP interface for network traffic extraction.

(d) Experiences

It is possible to create new plugins to analyse new types of data or adapt the main to produce new results with different formats. The analysis that is performed is based on a given set of security rules. These rules need to be carefully specified to avoid detecting to many false positives or to few true negatives.

(e) Samples

1. Detection of malicious nodes in an ad-hoc network:

Input:

- TDMA, Time Division Multiple Access, protocol traces from OSI layer 1+2 generated by a Omnet++ simulator in ASCII+HEXA format, as for instance:

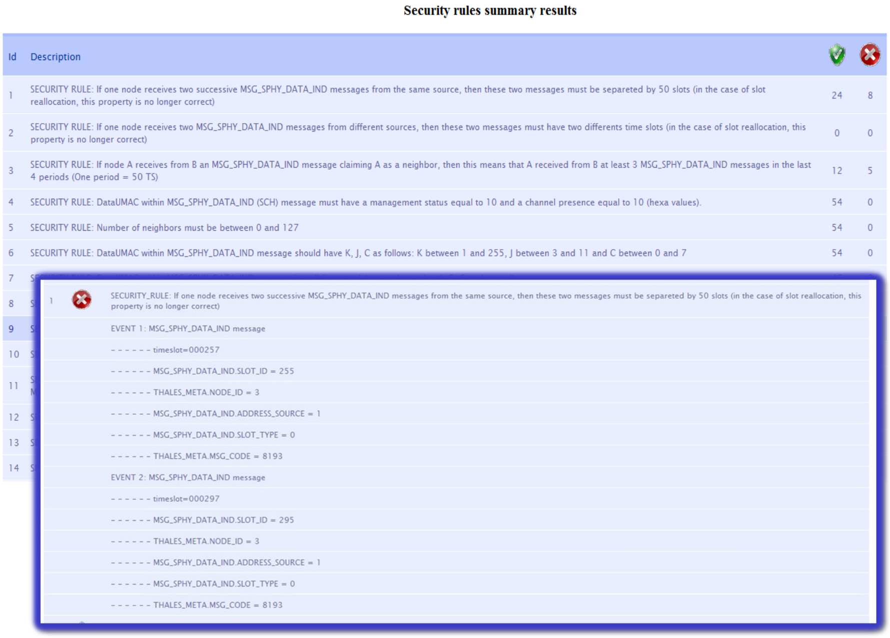
```
TS=5: smac[0x0002]: Reception SPHY_DATA_IND(SCH)            0000 01 2001 0001
00000005 0000 00 00 0030 0E 000014F0 00000000 000007D0 000007D0 00000000 00000003
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01 00 00 08 10 10 00 08 0A 02 00 02 000200 000100
...
```

Output:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="results.xsl"?>
<results>
<detail>
<occurence>
  <property_id>1</property_id >
    <verdict>not_respected</verdict>
  <description>
  ATTACK: A node is repeatedly sending MSG_SPHY_DATA_IND messages using incorrect
slots, provoking repeated slot reallocation. Could be interpreted as a DoS attack.
  </description>
<!--description of events that triggered the rule -->
<event>
```

```
<attribute><attribute_value>- - - - -
-timeslot=000005</attribute_value></attribute>
<description>EVENT: MSG_SPHY_DATA_IND message  received</description>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.ADDRESS_SOURCE =
10:10:00:08:0A:02:00:00</attribute_value></attribute>
<attribute><attribute_value>- - - - - - BASE.TIME_SLOT =
5</attribute_value></attribute>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.SLOT_ID =
1</attribute_value></attribute>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.SLOT_TYPE =
0</attribute_value></attribute>
<attribute><attribute_value>- - - - - - BASE.PROTO =
801</attribute_value></attribute>
</event>
<event>
<attribute><attribute_value>- - - - - -
timeslot=000005</attribute_value></attribute>
<description>EVENT: MSG_SPHY_DATA_IND messages must to be separated by 50
slots</description>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.ADDRESS_SOURCE =
10:10:00:08:0A:02:00:00</attribute_value></attribute>
<attribute><attribute_value>- - - - - - BASE.TIME_SLOT =
5</attribute_value></attribute>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.SLOT_ID =
30</attribute_value></attribute>
<attribute><attribute_value>- - - - - - MSG_SPHY_DATA_IND.SLOT_TYPE =
0</attribute_value></attribute>
<attribute><attribute_value>- - - - - - BASE.PROTO =
801</attribute_value></attribute>
</event>
</occurence>
...
```

That viewed with a browser gives something like this:



Security rules summary results

Id	Description	✓	✗
1	SECURITY RULE: If one node receives two successive MSG_SPHY_DATA_IND messages from the same source, then these two messages must be separeted by 50 slots (in the case of slot reallocation, this property is no longer correct)	24	8
2	SECURITY RULE: If one node receives two MSG_SPHY_DATA_IND messages from different sources, then these two messages must have two differents time slots (in the case of slot reallocation, this property is no longer correct)	0	0
3	SECURITY RULE: If node A receives from B an MSG_SPHY_DATA_IND message claiming A as a neighbor, then this means that A received from B at least 3 MSG_SPHY_DATA_IND messages in the last 4 periods (One period = 50 TS)	12	5
4	SECURITY RULE: DataUMAC within MSG_SPHY_DATA_IND (SCH) message must have a management status equal to 10 and a channel presence equal to 10 (hexa values).	54	0
5	SECURITY RULE: Number of neighbors must be between 0 and 127	54	0
6	SECURITY RULE: DataUMAC within MSG_SPHY_DATA_IND message should have K, J, C as follows: K between 1 and 255, J between 3 and 11 and C between 0 and 7	54	0
7	S		
8	S		
9	S		
10	S		
11	S M		
12	S		
13	S		
14	S		

1 ✗ SECURITY_RULE: If one node receives two successive MSG_SPHY_DATA_IND messages from the same source, then these two messages must be separeted by 50 slots (in the case of slot reallocation, this property is no longer correct)

```
EVENT 1: MSG_SPHY_DATA_IND message
- - - - - - timeslot=000257
- - - - - - MSG_SPHY_DATA_IND.SLOT_ID = 255
- - - - - - THALES_META.NODE_ID = 3
- - - - - - MSG_SPHY_DATA_IND.ADDRESS_SOURCE = 1
- - - - - - MSG_SPHY_DATA_IND.SLOT_TYPE = 0
- - - - - - THALES_META.MSG_CODE = 8193
EVENT 2: MSG_SPHY_DATA_IND message
- - - - - - timeslot=000297
- - - - - - MSG_SPHY_DATA_IND.SLOT_ID = 295
- - - - - - THALES_META.NODE_ID = 3
- - - - - - MSG_SPHY_DATA_IND.ADDRESS_SOURCE = 1
- - - - - - MSG_SPHY_DATA_IND.SLOT_TYPE = 0
- - - - - - THALES_META.MSG_CODE = 8193
```

(f) Licenses or patents

It is not bound to any licenses or patents.

3. Format details

(a)Transport protocol

No. The tool can detect and analyse more than 600 different protocols (that includes all the most common internet protocols and web applications), and more can be added if necessary.

(b) Structure or specification

i. Format specification

Input and output data is formally specified and can be machine processed.

ii. Availability of specification

Input data is specified by IETF (in the case of Internet protocols) or could be specific to certain applications/services/systems (in the case of, e.g., Business Activity Monitoring).

Output data is formally specified but is defined as needed and does not necessarily follow any standards.

iii. Extending the format

Both input and output can be extended to include new formats.

iv. Validate syntax and semantics

In most cases, yes, tools exist that can validate the correctness of input and output.

v. Representation

As preferred: textual, binary, XML, SQL...

(c) Type of data or threat

Output data is designed to detect any abnormal behaviour. For this, security properties or rules need to be defined that describe the sequence of events that can be considered a vulnerability or a threat. The tool will use these rules to detect occurrences of these sequences in the input and produce the results as output.

The security properties (that can be considered as internal data used by the tool) are written using a proprietary XML format. They can be specified by us or by others but require very good knowledge of the input that will be analysed and what can be considered correct or incorrect behaviour.

(d) Security aspects

i. Confidentiality and integrity

Output data can be encrypted.

ii. Authentication

Using public key encryption.

iii. Availability

Depends on the communication channel used.

(e) User group

Yes, to any user that needs to analyse communication traffic.

(f) Communication infrastructure

No special preferences, supports all communication infrastructures.

i. Peer to peer

Ok

ii. Centralised

Ok

iii. Closed user group

Ok

(g) Software components

A version of the **MMT_Core** module will be made available as freeware. This module captures and extracts the data needed from the input.

A version of the **MMT_Security** module will be available as Open Source. This module analysed the date extracted by MMT_Core and produces the output. Depending on the format of this output, other freely available tools probably exist that can be used to visualize or process it

The **MMT_QoS/QoE** module will be available only through licensing or special agreements.

Commercial use of any of the modules is subject to licensing or special agreements.

A.10. Questionnaire I

1. Format name and version

Sflow 5.0 – will be replaced by IPFix later this year

2. Use case

Exporting of Sflow samples.

(a) Role and rationale

Because its the only format our hardware supports

(b) Workflows

Receivers of format must sign to anonymize the data

(c) Software components and interfaces

Force10/Dell switches now. Alcatel-Lucent routers later this year.

(d) Experiences

We are happy with it

(e) Samples

See relevant RFCs

(f) Licenses or patents

no

3. Format details

(a) Transport protocol

Sflow, Netflow

(b) Structure or specification

i. Format specification

See RFCs

ii. Availability of specification

Yes, RFCs

iii. Extending the format

No

iv. Validate syntax and semantics

v. Representation

binary

(c) Type of data or threat

(d) Security aspects

i. Confidentiality and integrity

Content of data is confidential

ii. Authentication

no

iii. Availability

(e) User group

CERT, statisitics

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

Definitely yes – legal aspects apply

(g) Software components

Any sflow/netflow software like Arbor...

A.11. Questionnaire J

1. Format name and version

HPFEEDS is not a data format, but a transport protocol over TCP used to convey honeypots data feeds.

More information can be found at https://redmine.honeynet.org/projects/hpfeeds/wiki

It is widely used and developed by honeynet project crew (http://www.honeynet.org/about)

2. Use case

We use HPFEEDS protocol to collect data from heterogeneous honeypots belonging to our honeynet.

It is currently supported by a variety of honeypots:

1. dionaea http://dionaea.carnivore.it/,
2. kippo https://code.google.com/p/kippo/
3. glastopf http://glastopf.org/

and also by cuckoo sandbox http://www.cuckoosandbox.org/

Any kind of data format can be carried by this protocol without any constraints.

(a) Role and rationale

The "hpfeeds" project implements a lightweight authenticated publish/subscribe protocol for exchanging live datafeeds.

Different feeds are separated by channels and support arbitrary binary payloads. This means that the channel users have to decide about the structure of data. This could for example be done by choosing a serialization format.

It provides authentication of each subscriber/publisher over each channel and optionally the protocol can be run on top of SSL/TLS.

(b) Workflows

The main component is the so called “broker” that collects and dispatches live feeds among publishers and subscribers through authenticated channel. Each source can send and/or receive information in real time by publishing and/or subscribing to different channels.

Data format carried by each channel is not defined by the protocol, but have to be previously set by parties interested in the communication over the specific channels.

Nowadays most of existing channels uses JSON (http://www.json.org/) to exchange data

(c) Software components and interfaces

We use available implementation of broker and publisher (honeypot plugin/patches) component provided by honeynet project team (https://github.com/rep/hpfeeds).

Beside this we implemented proprietary software to parse and collect data (subscribers).

For debugging purposes Wireshark dissector has been implemented and included from latest Wireshark release.

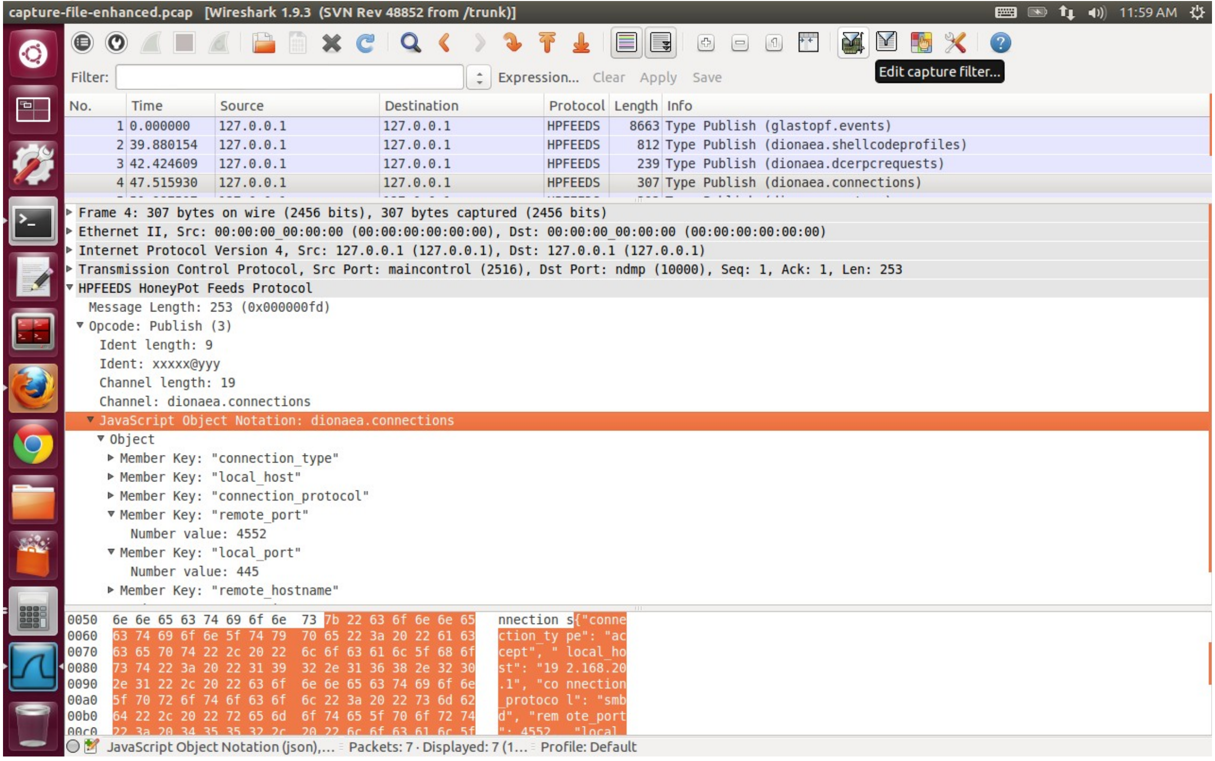
(d) Experiences

Very simple and flexible protocol, easy to set up and operate.

Scalability of the solution should be analysed/improved. Now there is one central point (the broker) that receives and relays all the messages. This should be a bottleneck and single point of failure in a big deployment.

(e) Samples

For demonstration purposes, some example messages analysed with wireshark:



capture-file-enhanced.pcap [Wireshark 1.9.3 (SVN Rev 48852 from /trunk)]

```
No.   Time        Source      Destination   Protocol  Length Info
    1 0.000000    127.0.0.1   127.0.0.1     HPFEEDS     8663 Type Publish (glastopf.events)
    2 39.880154   127.0.0.1   127.0.0.1     HPFEEDS      812 Type Publish (dionaea.shellcodeprofiles)
    3 42.424609   127.0.0.1   127.0.0.1     HPFEEDS      239 Type Publish (dionaea.dcerpcrequests)
    4 47.515930   127.0.0.1   127.0.0.1     HPFEEDS      307 Type Publish (dionaea.connections)
```

```
▸ Frame 4: 307 bytes on wire (2456 bits), 307 bytes captured (2456 bits)
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▸ Transmission Control Protocol, Src Port: maincontrol (2516), Dst Port: ndmp (10000), Seq: 1, Ack: 1, Len: 253
▾ HPFEEDS HoneyPot Feeds Protocol
    Message Length: 253 (0x000000fd)
  ▾ Opcode: Publish (3)
      Ident length: 9
      Ident: xxxxx@yyy
      Channel length: 19
      Channel: dionaea.connections
  ▾ JavaScript Object Notation: dionaea.connections
    ▾ Object
      ▸ Member Key: "connection_type"
      ▸ Member Key: "local_host"
      ▸ Member Key: "connection_protocol"
      ▾ Member Key: "remote_port"
          Number value: 4552
      ▾ Member Key: "local_port"
          Number value: 445
      ▸ Member Key: "remote_hostname"
```

```
0050  6e 6e 65 63 74 69 6f 6e  73 7b 22 63 6f 6e 6e 65   nnection s{"conne
0060  63 74 69 6f 6e 5f 74 79  70 65 22 3a 20 22 61 63   ction_ty pe": "ac
0070  63 65 70 74 22 2c 20 22  6c 6f 63 61 6c 5f 68 6f   cept", " local_ho
0080  73 74 22 3a 20 22 31 39  32 2e 31 36 38 2e 32 30   st": "19 2.168.20
0090  2e 31 22 2c 20 22 63 6f  6e 6e 65 63 74 69 6f 6e   .1", "co nnection
00a0  5f 70 72 6f 74 6f 63 6f  6c 22 3a 20 22 73 6d 62   _protoco l": "smb
00b0  64 22 2c 20 22 72 65 6d  6f 74 65 5f 70 6f 72 74   d", "rem ote_port
```

JavaScript Object Notation (json),... Packets: 7 · Displayed: 7 (1... Profile: Default

(f) Licenses or patents

HPFEEDS protocol is released under GNU PUBLIC LICENSE version 3
https://github.com/rep/hpfeeds/blob/master/LICENSE

3. Format details

HPFEEDS is not a data format specifications so many of the following questions do not apply in this context.

Any kind of data format can be carried by this protocol without any constraints.

(a) Transport protocol

The protocol is carried by TCP optionally the protocol can be run on top of SSL/TLS.

(b) Structure or specification

i. Format specification

ii. Availability of specification

iii. Extending the format

iv. Validate syntax and semantics

v. Representation

(c) Type of data or threat

(d) Security aspects

i. Confidentiality and integrity

Available only if protocol is run on top of SSL/TLS.

ii. Authentication

Currently supported

iii. Availability

(e) User group

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

This is the way the protocol works as the central node is the broker component.

iii. Closed user group

(g) Software components

Protocol implementation is available at https://github.com/rep/hpfeeds

Most channels use JSON http://www.json.org as data format.

A.12. Questionnaire K

1. Format name and version

IODEF (Incident Object Description Exchange Format) RFC 5070

2. Use case

(a) Role and rationale

CyDef receives some data in IODEF format from other parties, and also exports some data in this format. The biggest factor for using IODEF is that it's fairly simple and tailor-made for exchanging incident reports with CSIRTs.

(b) Workflows

CyDef doesn't store any data in IODEF format, but only converts to and from when exchanging data with other response teams.

(c) Software components and interfaces

Bespoke parsing library.

(d) Experiences

IODEF works very well when exchanging blacklistings and similar data. Through XML extensibility, it is able to include other events, such as attack patterns, vulnerabilities etc. However, our opinion is that when using extended data types, STIX offers a more promising solution (although we are yet to use it).

For this reason, we would recommend either using STIX for exchanging blacklisting data (pro: more standardisation; con: quite bloated for simple blacklisting data), or using STIX for the majority of data types with one or two exceptions, such as for blacklisting data.

(e) Samples

Samples available in RFC 5070: http://tools.ietf.org/html/rfc5070#section-7

(f) Licenses or patents

Rights are retained by the data owners. For full details, see IETF BCP 78 and IETF BCP 79.

3. Format details

(a) Transport protocol

No. Any protocol meeting certain requirements (confidentiality, integrity, authenticity, suitable compression & reliability) is suitable.

(b) Structure or specification

i. Format specification

Yes.

ii. Availability of specification

Publicly available on IETF's website. RFC 5070 covers the core specification, with others for extensions (e.g. RFC 5901 for phishing).

iii. Extending the format

Yes, but not realistic or advisable.

iv. Validate syntax and semantics

Yes, but no scripts are provided by IETF for this purpose. Bespoke code according to the specification is required.

v. Representation

XML.

(c) Type of data or threat

Principally for exchanging blacklists and other incident reports. But also contains other extensions (for phishing, attack patterns, vulnerabilities etc.).

(d) Security aspects

i. Confidentiality and integrity

Both are left to the transport protocol (it is not tied to a specific protocol).

ii. Authentication

Left to the transport protocol.

iii. Availability

Not covered by the data format and not possible to be covered by the transport protocol.

(e) User group

It is targeted towards CSIRTs, but has also been widely-used internally by corporations.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

Preferred, as it was designed to be exchanged with full knowledge between individual parties.

(g) Software components

No official software tools. However, several have been publicly released by CERTs and CSIRTs.

A.13. Questionnaire L

1. Format name and version

We are using IODEF format to exchange data on detected intrusion between SIEM systems and our cyber security hypervisor.

It is possible to reuse this format to exchange malware information from our malware analyzer to the security hypervisor.

2. Use case

IODEF data from customer networks to SOC hypervisor

(a) Role and rationale

Role of CSD (Cassidian CyberSecurity): security management

Role of customer: target of intrusions

Format used to describe event flows (src/tgt), nature of the incident, date of occurrence, impact assessment

(b) Workflows

The SOC team manages different customers at the same time. Gathered Incident data are imported by CSD from these customers. Depending on the customer, the incident resolution is assigned to a specific team.

(c) Software components and interfaces

Interface between SIEM and hypervisor is IODEF\SOAP

(d) Experiences

IODEF is a very detailed format but most of the information is not used by SIEM systems

A lightest exchange format would be preferable. Sometimes we use syslog interfaces when SIEM product has little information to transmit to the hypervisor.

(e) Samples

(f) Licenses or patents

SIEM are commercial products. Hypervisor is property of CSD.

IODEF\SOAP interface may be reused as web service (wsdl available).

3. Format details

(a) Transport protocol

IODEF\SOAP is an Http request

(b) Structure or specification

i. Format specification

WSDL which is an XML description of the requests and responses supported by the web service

ii. Availability of specification

WSDL can be delivered by CSD, a document describing the interface is also available explaining the purpose and structure of the requests (function calls) & responses (function returns)

iii. Extending the format

It is possible to add functions and/or to add parameters to existing functions

iv. Validate syntax and semantics

Yes it is: use of IODEF xsd

v. Representation

XML

(c) Type of data or threat

Designed for cyber security incidents

(d) Security aspects

i. Confidentiality and integrity

Both

ii. Authentication

Yes

iii. Availability

The format leverages the robustness of the HTTP protocol

(e) User group

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

Preferred

iii. Closed user group

(g) Software components

Software applications used in this kind of exchanges are commercial products.

A.14. Questionnaire M

1. Format name and version

Name is X-ARF, version of the specification is v0.2.

2. Use case

X-ARF data exchange between CSIRTs

(a) Role and rationale

DFN-CERT uses the X-ARF format to report incidents to other CSIRTs. The key advantage of the format is the flexibility. X-ARF contains both a textual human readable as well as structured part. The textual part can be understood without knowledge of the format and is therefore intended for sites that are not used to X-ARF reports. However, the format allows other sites to automate the processing of the reports.

(b) Workflows

The incident data is imported by DFN-CERT from different sources. Workflows exist to assign the source of an incident to the appropriate site or CSIRT. The data is then used to produce an X-ARF report that is sent to the site.

(c) Software components and interfaces

DFN-CERT uses an internal implementation of interfaces to import, parse, and export X-ARF messages. Additionally, a sample script exists that inspects SSH server logs for attacks and produces X-ARF reports.

(d) Experiences

X-ARF performs well for manual and automatic processing. A drawback is the inefficient data transport when a separate mail for each event is transferred. This is especially true for bulk data. To overcome this, an extension is part of the standard that provides a specification to optionally aggregate multiple incidents in a single message. Moreover, a compression of the textual data on the transport would lead to a further improvement of its efficiency. In the current specification, X-ARF messages are transferred by email. Future releases may consider a transport channel by HTTP (e.g. HTTP REST interface).

(e) Samples

It is attached below.

(f) Licenses or patents

It is not bound to any licenses or patents.

3. Format details

(a) Transport protocol

X-ARF messages are transferred by email (SMTP). Future specifications may also consider the transport by HTTP/REST.

(b) Structure or specification

i. Format specification

X-ARF messages are separated into three parts. The first is a textual description of the content. The second part consists of a machine-readable part. Its structure is provided by YAML/JSON. The third part is optional and may contain evidence of the incident (e.g. logs) or malware samples.

ii. Availability of specification

Yes, at http://x-arf.org

iii. Extending the format

Yes, the specification includes a private schema. Additionally, other schemas regarding other attack data can be proposed in collaboration with the working group.

iv. Validate syntax and semantics

Yes, this is true for the second part (validation of correct syntax)

v. Representation

All parts contain textual data.

(c) Type of data or threat

X-ARF provides multiple schemas related to different attack data. Schemas exist for port-scanning activity, spam, and malware.

(d) Security aspects

i. Confidentiality and integrity

Yes, by using S/MIME signatures and encryption

ii. Authentication

Yes, by using S/MIME signatures

iii. Availability

The format leverages the robustness of the SMTP protocol.

(e) User group

X-ARF addresses different user groups. The first informal part is intended for users that are not familiar with X-ARF while the second part is machine-readable and supports automation.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

X-ARF supports all communication infrastructures.

(g) Software components

The software is available at http://x-arf.org. It is not bound to any licenses or patents.

Sample of X-ARF message

```
From: xxxxx@xxxxxxxx.de
To: xxxx@xxxxxxxx.de
Reply-To: xxxx@xxxxxxxx.de
X-Data-Format: X-ARF
Organisation: xxxxxxxx
X-System-Id: xxxxx.xxxxxxxx.de
X-Script-Version: 2010-12-21
X-Script-Name: xarf-ssh-reporter.sh
X-ARF: yes
Auto-Submitted: auto-generated
Subject: abuse report about xxx.xxx.129.56 - 2012-06-10
Mime-Version: 1.0
Content-Type: multipart/mixed; charset=utf8; boundary="Abuse-
64a4e26a2f19ad1616aa764f5edf8679"
Message-Id: <20120610060031.2BBABA0250@xxxxxxxx.de>
Date: Sun, 10 Jun 2012 08:00:31 +0200 (CEST)

This message is in MIME format. But if you can see this,
you aren't using a MIME aware mail program. You shouldn't
have too many problems because this message is entirely in
ASCII and is designed to be somewhat readable with old
mail software.


--Abuse-64a4e26a2f19ad1616aa764f5edf8679
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8;


Dear DFN-CERT,

this is an automated report for ip address xxx.xxx.129.56 in format "X-ARF"
generated on 2012-06-10 08:00:31 +0200

ip address xxx.xxx.129.56 produced 314 log lines, sample log lines attached.

Regards,
DFN-CERT Team


--Abuse-64a4e26a2f19ad1616aa764f5edf8679
MIME-Version: 1.0
```

```
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8; name="report.txt";

---
Reported-From: xxxxxx@xxxxxxxx.de
Category: abuse
Report-Type: login-attack
Service: ssh
Port: 22
User-Agent: xarf-ssh-reporter.sh 2010-12-21
Report-ID: 13392288495782@xxxxxxxx.de
Date: Sat, 09 Jun 2012 10:00:49 +0200
Source: xxx.xxx.129.56
Source-Type: ipv4
Attachment: text/plain
Schema-URL: http://www.x-arf.org/schema/abuse_login-attack_0.1.1.json

--Abuse-64a4e26a2f19ad1616aa764f5edf8679
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8; name="logfile.log";

2012-06-09 10:00:49 +0200 XXXXXX sshd[26790]: Did not receive identification string
from xxx.xxx.129.56
2012-06-09 10:05:40 +0200 XXXXXX sshd[27285]: Invalid user abdulghaffar from
xxx.xxx.129.56
2012-06-09 10:05:47 +0200 XXXXXX sshd[27305]: Invalid user abdulkader from
xxx.xxx.129.56
-- MARK --
2012-06-09 10:42:30 +0200 XXXXXX sshd[970]: Invalid user atmail from xxx.xxx.129.56
2012-06-09 10:42:41 +0200 XXXXXX sshd[1000]: Invalid user atn from xxx.xxx.129.56
2012-06-09 10:42:49 +0200 XXXXXX sshd[1023]: Invalid user atowar from
xxx.xxx.129.56

--Abuse-64a4e26a2f19ad1616aa764f5edf8679--
```

A.15. Questionnaire N

1. Format name and version

IDMEF

2. Use case

(a) Role and rationale

The specific use case of IDMEF is the transport of IDS data such as Snort to a central storage centre. For example, the CarmentiS early warning system is capable of processing IDMEF reports.

(b) Workflows

The primary purpose of IDMEF is enabling transportation of attack data from a distributed network of IDS sensors.

(c) Software components and interfaces

NIDS such as Snort and Prelude export data. In addition, the Prelude framework provides a programming library to produce, process, and import IDMEF data. The Prelude library is also part of CarmentiS to process data.

(d) Experiences

The formats work pretty well for NIDS data. A nice feature is its capability to aggregate multiple correlated events.

(e) Samples

Samples are provided in RFC4765

(f) Licenses or patents

No

3. Format details

(a) Transport protocol

No. Since the format is based on XML all protocols can be used that support XML.

(b) Structure or specification

i. Format specification

Yes, it is structured by XML; see RFC4765 for further details.

ii. Availability of specification

Yes, it is detailed in RFC4765

iii. Extending the format

Yes, IDMEF provides some means to extend the format.

iv. Validate syntax and semantics

Yes

v. Representation

Representation is textual. However, some programs such as Prelude provide a binary representation of IDMEF data.

(c) Type of data or threat

The format is devoted to IDS alerts.

(d) Security aspects

i. Confidentiality and integrity

Yes, e.g. by the Prelude library.

ii. Authentication

Yes, e.g. by the Prelude library.

iii. Availability

No

(e) User group

Since the format is intended to submit IDS data in an automated way, it is not addressed to a specific user group.

(f) Communication infrastructure

i. Peer-to-peer

Yes

ii. Centralised

Yes

iii. Closed user group

No

(g) Software components

For example, the Prelude framework provides a free version of a library to process IDMEF messages. It is published under the terms of the GNU General Public License.

A.16. Questionnaire O

1. Format name and version

STIX (Structured Threat Information eXpression) V1.0

2. Use case

(a) Role and rationale

STIX (http://stix.mitre.org/) is a community driven effort to develop a standardized threat information format. It is coordinated by Mitre, and as such it extends work on previous standards they have produced, with a STIX message potentially including Cyber Observables (CybOX), Malware Definitions (MAEC) and Attack Patterns (CAPEC). STIX combines structured XML that describes observed security related events and artefacts with a framework that caters for analysis elements.

There is a great deal of interest in using STIX, as it appears to offer high functionality in a well defined standard that will facilitate automated exchange of threat information. LSEC, the ACDC WP2 leaders, are therefore initiating a small project to examine how STIX could be utilised as a data format for tool reporting in cooperation with a small number of ACDC tool providers.

(b) Workflows

It is intended that information logged or otherwise provided by tools will be either directly created as STIX messages, or converted from their current format into STIX. The STIX messages will then be stored centrally, where the benefits of a common format reported by disparate tools can be examined.

(c) Software components and interfaces

A STIX demonstrator system will be produced, which will provide a centralised, database backed store, with a web service interface that allows authorized tools to submit data in STIX format. The web service may be an implementation of the Trusted Automated eXchange of Indicator Information (TAXII) protocol with XML binding or a simpler interface depending upon ease of integration. The STIX demonstrator will be a STIX consumer, tools that send STIX information will be STIX producers. STIX producers can be integrated directly with the STIX demonstrator, or a command line stub will be made available that will allow easy integration into existing reporting mechanisms without significant work by the tool partner.

(d) Experiences

At the moment this work has only just started, however the full intention is to provide extensive feedback to the rest of the ACDC project.

(e) Samples

A number of examples of STIX documents can be found at https://github.com/STIXProject/schemas/tree/master/samples however one example is also included at the end of the questionnaire.

(f) Licenses or patents

The copyright for STIX and all associated Mitre initiatives belongs to the Mitre Corporation, who openly grant a royalty free license for use (http://stix.mitre.org/about/termsofuse.html).

3. Format details

(a) Transport protocol

STIX messages are well formed XML documents, and could be transported using many Internet protocols, however there is a specific transport specification called TAXII which includes bindings to HTTP and to XML for web services.

(b) Structure or specification

i. Format specification

Yes, STIX and all included data formats are defined by schema files maintained by Mitre on behalf of the community.

ii. Availability of specification

Yes, the current versions of the schema files are available here: http://stix.mitre.org/language/version1.0/

iii. Extending the format

From the website: “STIX also offers a set of loosely coupled schema extension points and related default extensions for various purposes, such as use of externally-defined schemas for relevant information, data marking models and controlled vocabularies.”.

iv. Validate syntax and semantics

Yes, by validating a STIX xml message against the schema files.

v. Representation

Messages are represented as well formed XML documents.

(c) Type of data or threat

STIX provides multiple schemas for representing various types threats and actors, including Indicators, Threat Actors, Campaigns, Incidents, and Tactics, Techniques and Procedures (TTP). The inclusion of elements from other standards such as CybOX allows many types of observables to be included within the STIX document, such IP Addresses, E-mails, Attachments, files, network packets etc.

(d) Security aspects

i. Confidentiality and integrity

If the STIX documents are transported via TAXII this could be done via TLS.

ii. Authentication

This is not part of STIX or TAXII, but would be the responsibility of the either the TAXII back-end architecture, or security provided by the infrastructure, such as certificate authentication on TLS.

iii. Availability

Nothing in the standards themselves.

(e) User group

It is not aimed at any particular target group, but is intended to be able to widely support the sharing of threat intelligence amongst interested parties. The standard has wide support amongst CERTS, ISACs, commercial and government organisations in the US, ACDC could be an early adopter in the EU.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

TAXII describes “Hub and Spoke” which is a centralised approach, Full peer-to-peer, and Source/Subscriber, which allows a source to push information to all subscribers.

(g) Software components

Python examples are supplied that demonstrate STIX document parsing (https://github.com/STIXProject/python-stix), STIX is mainly about the definitions supplied by the schema files, which can be readily handled using standard XML library tools in most languages.

The following shows a STIX document that represents a threat indicator – in this case a list of malicious URLs. The URLs are represented as a CybOX element within the STIX document.

```
<!--
       STIX IP Watchlist Example

       Copyright (c) 2013, The MITRE Corporation. All rights reserved.
    The contents of this file are subject to the terms of the STIX License located
at http://stix.mitre.org/about/termsofuse.html.

       This example demonstrates a simple usage of STIX to represent a list of URL
indicators (watchlist of URLs). Cyber operations and malware analysis centers often
share a list of suspected malicious URLs with information about what those URLs
might indicate. This STIX package represents a list of three URLs addresses with a
short dummy description of what they represent.

       It demonstrates the use of:

          * STIX Indicators
          * CybOX within STIX
          * The CybOX URI Object (URL)
          * CybOX Patterns (apply_condition="ANY")
          * Controlled vocabularies

       Created by Mark Davidson
-->
<stix:STIX_Package
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:stix="http://stix.mitre.org/stix-1"
    xmlns:indicator="http://stix.mitre.org/Indicator-2"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:URIObject="http://cybox.mitre.org/objects#URIObject-2"
    xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
    http://stix.mitre.org/stix-1 ../stix_core.xsd
    http://stix.mitre.org/Indicator-2 ../indicator.xsd
    http://cybox.mitre.org/default_vocabularies-2
../cybox/cybox_default_vocabularies.xsd
    http://stix.mitre.org/default_vocabularies-1 ../stix_default_vocabularies.xsd
    http://cybox.mitre.org/objects#URIObject-2 ../cybox/objects/URI_Object.xsd"
    >
    <stix:STIX_Header>
        <stix:Title>Example watchlist that contains URL information.</stix:Title>
        <stix:Package_Intent xsi:type="stixVocabs:PackageIntentVocab-
1.0">Indicators - Watchlist</stix:Package_Intent>
    </stix:STIX_Header>
    <stix:Indicators>
        <stix:Indicator xsi:type="indicator:IndicatorType" id="example:Indicator-
db4a6ffe-61f0-488d-85a1-20bd5e360f37">
            <indicator:Type xsi:type="stixVocabs:IndicatorTypeVocab-1.0" >URL
Watchlist</indicator:Type>
            <indicator:Description>Sample URL Indicator for this
watchlist</indicator:Description>
            <indicator:Observable id="example:Observable-Pattern-cc5c00ce-98a6-
4cbe-8474-59eaecdb018f">
                <cybox:Object>
                    <cybox:Properties xsi:type="URIObject:URIObjectType">
                        <URIObject:Value condition="Equals"
apply_condition="ANY">http://example.com/foo/malicious1.html,http://example.com/foo
/malicious2.html,http://example.com/foo/malicious3.html</URIObject:Value>
                    </cybox:Properties>
                </cybox:Object>
            </indicator:Observable>
        </stix:Indicator>
    </stix:Indicators>
</stix:STIX_Package>
```

A.17. Questionnaire P

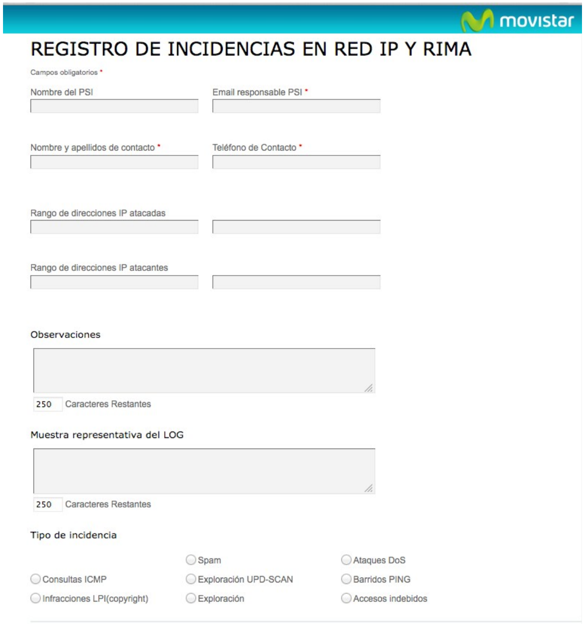
1. Format name and version

I+D (TID) tools data formats are:

- Spam-bot and DNS-bot detector, part of a DPI in house product, with real inline traffic. Inputs and outputs are based on CSV formatted files for aggregate information. Planning for a Standard data format like IODEF is scheduled.
- SDN Malware detector based on beta Commercial product use standard SYSLOG protocol as an output.

ISP network haves his own data format use:

- Manual (e-mail & phone) information exchange with authorities, CERTs, ISPs, etc.... There is no data format define at the moment.
- E-mail abuse-mailbox: xxxxxx@txxxxxx.es Support email with open text format with claims related to SPAM and botnet activity. Only one requirement: needs, as probe of the offense, the original SPAM offending mail with ALL mail headers.
- HTTP in web page (http://www.xxxxx.es/xxxxx) for complaints and abuse from final users, ISP clients or other ISP. Includes options for:
  - ISP complainant name & contact email
  - Complainant person identification
  - IP origin of attack
  - IP destination of attack
  - Comments
  - Log data of the complaint.
  - Type of complaint (scanning, infringement of IPR, SPAM, DoS,..)



REGISTRO DE INCIDENCIAS EN RED IP Y RIMA

Campos obligatorios *

Nombre del PSI

Email responsable PSI *

Nombre y apellidos de contacto *

Teléfono de Contacto *

Rango de direcciones IP atacadas

Rango de direcciones IP atacantes

Observaciones

250 Caracteres Restantes

Muestra representativa del LOG

250 Caracteres Restantes

Tipo de incidencia

○ Spam ○ Ataques DoS

○ Consultas ICMP ○ Exploración UPD-SCAN ○ Barridos PING

○ Infracciones LPI(copyright) ○ Exploración ○ Accesos indebidos

Following details are from TID tools.

2. Use case

(a) Role and rationale

Spam-bot & DNS-bot detectors allow inspecting network traffic and detecting ISP users infected with a botnet. This solution is expected to be deploy inside a ISP Networks (not at this moment). CSV Format is human readable, allow easy conversion to other formats and integration in Databases.

SDN Malware detector allow detecting infected employees by botnet inside a Enterprise. Syslog protocol allow near time real incident alert.

(b) Workflows

SDN Malware detector tool generate output flows (syslog) of real time detections of infected user.

Spam.bot & DNS–bot module in DPI generate and export aggregate files with detections between a time of period (default value is 15 minutes). Also as part of a workflow we are planning reports generation.

These tools are source of detections and therefore can export the information to a Centralized point. Inputs requirements can be updates of IPs and Domains from Centralized point or third sources to increase number of detections and mitigations.

(c) Software components and interfaces

SDN malware detector is based on Hardware switches with Openflow support, and a Virtual Software OpenFlow Controller of a Third Vendor Beta product able to receive DNS traffic and checks domains against several domains Blacklist. Positive detections generate syslog messages.

Spam-bot & DNS-bot are proprietary software analysis module running in a Linux system. Analysis is done with the information received from a generic HW DPI over proprietary format. Python scripting libraries for integration with new data formats are preferred.

(d) Experiences

Live pilot experience with SDN Malware detector in TID network show that Syslog protocol needs syslog servers infrastructure but also that generate accurate information (real time botnets activity). Perfect for centralized solutions like ACDC.

We would like to improve reports capacity generation in both tools.

We are planning a extension to new data formats. We are willing to have a common reference to develop inside of the ACDC project.

(e) Samples

Output CSV file sample from Spam-bot detector DPI module:

```
ANALYSIS DATE:      1363507205
1363507242     1.1.1.1          0   166  0  0   0  388   MOBILE     11454
1363507223     100.100.100.100  0   160  0  0   0  352   LANDLINE   11475
1363507493     10.10.10.10      0   149  0  0   0  389   MOBILE     10723
```

Input file of Botnet Domains for DNS-bot detector DPI module:
Malwarefamily.dbl:

```
99-300.ru
360safeupdate02.gicp.net
3apa3a.tomsk.tw
```

Syslog message from SDN Malware detector:

```
Mon May 27 13:03:11 CEST 2013  CEF:0|Vendor|Controller|1.0.0|55|DNS query
notification|6|msg=OF Switch ID: 00:00:00:00:00:00:00:00 InPort: 25 Score: 80,
Tags: Botnet dvc=10.0.1.1 src=10.1.2.138 act=DROP_NOTIFY dhost=malware.domain

Wed May 29 15:28:38 CEST 2013  CEF:0|Vendor|Controller|1.0.0|55|DNS query
notification|6|msg=OF Switch ID: 00:00:00:00:00:00:00:00 InPort: 27 Score: 0, Tags:
Custom blacklist (Web app) dvc=10.0.1.1  src=10.1.2.138 act=NOTIFY
dhost=custom_malware.domain
```

(f) Licenses or patents

SDN Malware Detector is based in beta testing Commercial product with license cost. Syslog content message format is proprietary.

Spam-bot & DNS-bot detector module is based on proprietary and patented protocols of developed DPI. Output formats can be standard formats. Licenses model is being studied.

3. Format details

(a) Transport protocol

SDN Malware Detector uses UDP/514 Syslog protocol.

Spam-bot & DNS-bot detector module doesn’t have any transport protocol requirement.

(b) Structure or specification

i. Format specification

SDN Malware Detector uses syslog message. Field separators “|”

```
<Datetime>  CEF:<number>|<Vendor name>|<SDN Controller hostname>|<Version>|
<number>|DNS query notification|6|msg=OF Switch ID: <MAC Address> InPort: <port
number> Score: <value>, Tags:<type of domain> dvc=<switch_IP> src=<infected IP>
act=<NOTIFY,DROP,DROP_NOTIFY> dhost=<malware domain>
```

Spam-bot Detector output CSV:
First Line: `ANALYSIS DATE:    <datetime_decimal>`
Each following line fields separator (tab):

- Datetime decimal format when spammer wast first seen.
- Spammer IP address (public or Private)
- Detection trigger (Zero if no detection happen): number of sent mails
- Detection trigger (Zero if no detection happen): DNS Queries
- Detection trigger (Zero if no detection happen): SMTP error response
- Detection trigger (Zero if no detection happen): number of different senders
- Detection trigger (Zero if no detection happen): SMTP sents.
- Network VLANs number
- Network access (landline or mobile)
- Bytes consumed

ii. Availability of specification

No.

iii. Extending the format

Yes can be extended for CSV format as an evolving product in testing phase.

iv. Validate syntax and semantics

Could be done, but there is no available tools at the moment.

v. Representation

Human readable text in all case.

(c) Type of data or threat

SDN Malware Detector generates a atomic syslog alert from a user accessing to a malicious domain. These domains are related with botnet controller, droppers, phising, etc. The data include the user IP, domains accessed and actions done (drop, alert or both).

Spam-bot & DNS-bot Detector output CSV format are designed to collect identification of infected users of spam botnets or generic botnets detected by SMTP and DNS protocols that allow a ISP to remediate his users.

(d) Security aspects

i. Confidentiality and integrity

None mechanism is available at this development stage. There can be delegated to standards protocols mechanism (like FTP, SCP, HTTP and so forth).

ii. Authentication

None mechanism is available at this development stage. Transport can be delegated to standards protocols authentication mechanism (like FTP, HTTP and so forth).

iii. Availability

None mechanism is available at this development stage. Transport can be delegated to standards protocols authentication mechanism (like FTP, HTTP and so forth).

(e) User group

Target user group for Spam-bot & DNS-bot detector are all ISP clients (landline and mobile), mainly Security Operations Centers (SOC) or TAC.

SDN Malware detector target user group is SME and Corporations companies.

(f) Communication infrastructure

i. Peer to peer

ii. Centralised

iii. Closed user group

Centralized communications or Closed user groups is preferred in most case because of aggregation of data needed.

(g) Software components

Spam-bot & DNS-bot detector use DPI proprietary and patented software over general PC architecture hardware, for data capture and aggregation. Python is used in import, export of data information.

Proprietary Switch OS with standard Openflow protocol support and proprietary Java based Controller is used for SDN Malware detector. Linux standard syslog daemon is used for export information.

A.18. Questionnaire Q

1. Format name and version

- Raw Data Events

Raw Data events are un-processed events, and the input that AHPS will accept from other ACDC components.

Raw data collected from the Event Sources is received by AHPS connectors and stored in raw data files. Each raw data event is represented as a single line in a raw data file, in the form of a JSON object with a predefined format.

- AHPS Events

AHPS events are processed events, and the output that AHPS will generate in the context of ACDC.

The Atos High Performance Security (AHPS) Event Format is based on the Distributed Audit Services (XDAS)[1] standard.

2. Use case

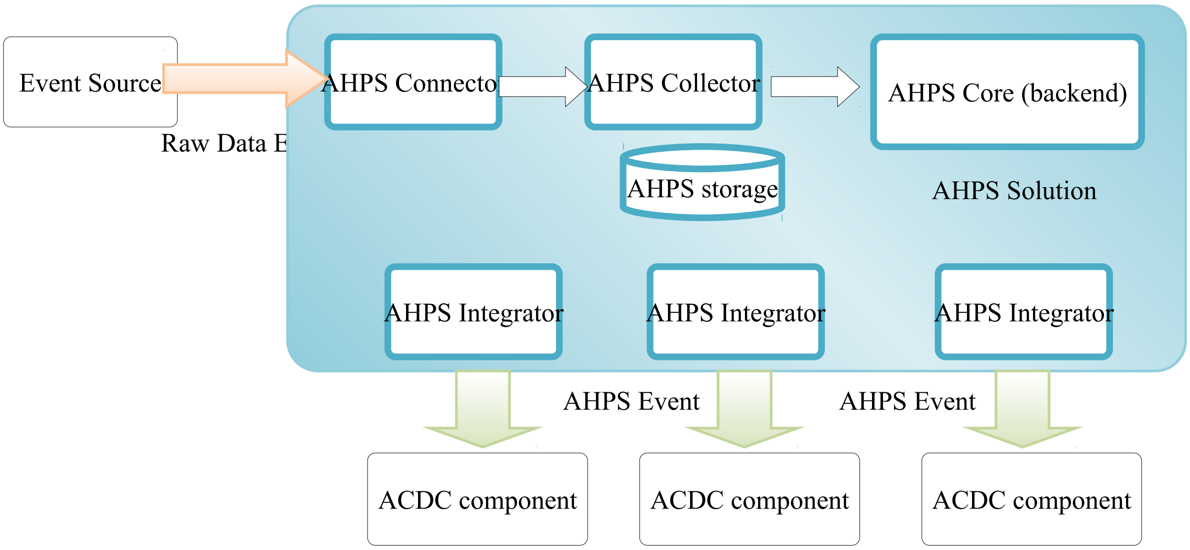
(a) Role and rationale

The AHPS would receive information (Raw data events) from other ACDC components, such as network or device sensor tools. Once we know the source of the information and the format, we will need to develop a Collector component and a Connector for each one, and configure AHPS to use it.

The AHPS is mainly an analysis component that receives information from other sources and normalizes, filters, correlates and analyses the information received to automatically identify inconsistencies in the environment. Based on these inconsistencies, AHPS identifies and alert on anomalous activity or new suspicious trends, alerting of potential threats or attacks.

The information generated by AHPS (AHPS events) could be used as input to other ACDC tools and stored in the CCH for further analysis.

(b) Workflows



The AHPS takes the input from event source "connectors" and converts the raw data into a textual map form consumable by, what is called "collectors". Collectors parse and normalize the textual map and create an AHPS Event, categorizing it according to the AHPS taxonomy of events. The AHPS Event is enriched with additional source-specific data and, depending on the collector, may apply additional contextual metadata such as identity, host, vulnerability, or custom mapped metadata. AHPS events can be sent to other external systems or components by means of "Integrators".

It is possible to export or download the raw data files containing raw data events, collected from event sources and stored in AHPS, in the CSV format.

[1] Distributed Audit Service (XDAS) https://www2.opengroup.org/ogsys/catalog/p441

AHPS Events will be output of AHPS and can be imported and exported from the internal AHPS database through the AHPS web interface.

(c) Software components and interfaces

- Software: AHPS solution
- Interfaces:
  - AHPS web interface: user interface for configuration and interaction with the solution
  - Event Source Connector API: Java API
  - Event Collectors API: JSP API
  - Report generation interface
  - Integrator API

(d) Experiences

(e) Samples

(f) Licenses or patents

Yes, both the AHPS Event Format and Raw data format are Atos proprietary.

3. Format details

(a) Transport protocol

No, it is a matter of configuration. AHPS can collect data from a wide range of event sources, such as intrusion detection systems, firewalls, operating systems, routers, databases, switches, mainframes, antivirus applications, and other applications. The configuration required to integrate a new event source with AHPS varies, depending on the type of event source and the communication method selected. For example, to accept Syslog data from Syslog event sources that send data over TCP (port 1468),UDP(port 1514), or SSL(port 1443). You can also configure AHPS to listen on additional ports.

(b) Structure or specification

i. Format specification

Yes, for both Raw data events and AHPS events

ii. Availability of specification

No, they are not publicly available. The AHPS event format is based on XDAS standard.

iii. Extending the format

The AHPS event format allows for custom extensions by using extension fields.

The Raw data event format is not extensible, but it is possible to create “mappers” to transform from one origin format to the AHPS raw data format, and there is a field that works as a payload, where the original event can be dumped.

iv. Validate syntax and semantics

Yes.

(c) Representation

Textual.

(d) Type of data or threat

AHPS raw data event format is used to describe events collected from the following sources:

- Security Perimeter: Devices and software used to create a security parameter for your environment.
- Operating Systems: Events from the different operating systems running in the network.
- Referential IT Sources: The software used to maintain and track assets, patches, configuration, and vulnerability.
- Application Events: Events generated from the applications installed in the network.
- User Access Control: Events generated from applications or devices that allow users access to company resources.

The AHPS Event model describes event activity generated from integrated devices, services, and applications. The fields of the event may describe a complex resource such as a user account residing in a directory hosted by a particular server, or software module running inside a service hosted by a particular server, and so forth. The AHPS Events are classified according to a taxonomy, which uses the XDAS standard taxonomy (v1), a classification that is intended to group events of similar type together to ease reporting and searching. Rather than use proprietary, app-specific event names (login, authenticated, logged in, etc), all events of a particular type should map to the same taxonomic classification.

AHPS Events are correlated and analyzed by AHPS to provide notifications about incidents and attacks. Also, there is a feature to cross-reference between event data signatures and vulnerability scanner data, generating feeds which contain information about vulnerabilities and threats, and associated remediation information.

(e) Security aspects

i. Confidentiality and integrity

Raw data can be checked for integrity by using the corresponding AHPS UI option. This feature checks integrity by various means, for example:

- Verify the sequence number of JSON records, by using the fields ChainID and ChainSequrence
- Verify the RawDataHash against the RawData

ii. Authentication

Secured data collection is determined by the specific protocols supported by the event source.

Internally, the protocol used for communication between the server and the database is defined by a JDBC Driver. For networked storage locations to store the event data and raw data, it depends on the capabilities of the type of server used. For example, CIFS or NFS servers do not offer data encryption, while local or SAN storage servers do not have the same security vulnerabilities.

AHPS uses several digital, public-key certificates as part of establishing secure TLS/SSL communications.

iii. Availability

(f) User group

Target group: Enterprises

(g) Communication infrastructure

The preferred communication infrastructure would be AHPS to run as a external service for ACDC solution, which will receive events from other AHPS components (e.g. network sensors, vulnerability scanners, etc) and will produce as output AHPS events (representing attacks, incidents, threats, vulnerabilities, etc.) as well as reports and countermeasures.

i. Peer-to-peer

ii. Centralised

iii. Closed user group

(h) Software components

AHPS provides interfaces for import, export, parse, mapping, normalization, correlation data, but all of them are proprietary. However, it is possible to develop components to adapt to the specific formats of external components.

B. Report Schema Usage

This appendix lists the schema names together with the partners that can provide data with this schema. This is intended to have a snapshot on the expected usage of schemata as well as list the possible additional data that a partner can provide. It has been used to evaluate whether the schemata should be extended and whether the additional data could provide for other analysis or aggregation if present.

This list is not supposed to be a complete list of data providers but just aggregates the information obtained as feedback to the schema proposal.

Partner	Data source	Available additional data
eu.acdc.attack		
CARNet	Subcategory abuse: Spam mail sent to spamtrap	
	Subcategory compromise: Attacks on web honeypot	
DE-CIX	Subcategory dos: Attacks redirected to a blackhole	• the bandwidth of the attack in kbit/s
TI	Subcategories other and malware: Attacks on Dionaea honeypot	• hostname of the attacker (via reverse lookup) • application_protocol • dst_access_type
	Subcategory login: Attacks on Kippo SSH honeypot	• SSH banner of attacker • credentials used • whether the login was successful • log of the SSH session • dst_access_type
	Subcategories compromise or data: Attacks on Glastopf honeypot	• HTTP request send to the honeypot • HTTP response from the honeypot • Attack type as provided by Glastopf (pattern) • dst_access_type
	Subcategory dos.tcp: Attackers sending large amounts of requests to the honeypots	• dst_access_type
TID	Subcategory abuse: Attackers sending out spam.	• src_access_type if known • number of emails sent by the attacker • number of DNS queries of type MX performed by the attacker • number of SMTP errors the server sent to the attacker • number of distinct domains used in from addresses • number of emails sent to port 587

	Subcategory malware: Attacks on Amun honeypot	• network service that is attacked • vulnerability exploited in the attack • src_access_type if known
	Subcategories compromise or data: Attacks on Glastopf honeypot	• HTTP request send to the honeypot • Attack type as provided by Glastopf (pattern) • src_access_type if known
	Subcategory login: Attacks on Kippo SSH honeypot	• src_access_type if known
eu.acdc.fast_flux		
CARNet	Domains identified with pDNS sensor	
INTECO	Data captured with Flux Detect	
TID	Domains with features suspicious of being fast flux (e.g. low TTL, multiple IPs, spatial distribution)	• src_access_type if known
eu.acdc.malicious_uri		
CARNet	Subcategories malware, phishing, exploit: Data collected by honeypot, spamtrap or NIRC	
INTECO	Subcategory malware: Data collected by Skanna	
TID	Subcategory other: URIs that are suspicious because their domain names look suspicious (e.g. long domain names, domain names containing a lot of numbers)	• src_access_type if known
	Subcategory other: URIs that are suspicious because they were queried in a large batch like a bot contacting DGA generated domain names	• number of DNS queries sent by a particular IP • number of NX errors in above DNS queries • src_access_type if known
eu.acdc.malware		
CARNet	Malware collected from spam attachment or web honeypot	
FKIE	Analyses of suspicious PDF documents	
INTECO	Samples collected by Skanna	
TI	Malware samples uploaded to Dionaea honeypot	• the type of the binary (e.g. "PE32 executable (DLL) (GUI) Intel 80386, for MS Windows")
TID	Malware samples uploaded to Amun or Glastopf honeypot	
eu.acdc.spam_campaign		

CARNet	Campaigns identified with spamtrap	

C. Report Schemata

This appendix lists the complete JSON schemata for the report categories in ACDC and suggested schemata for the fields in the field list. The field schemata may be used in schemata for the additional_data section or to illustrate their intended usage.

C.1. Attack (eu.acdc.attack)

```
{
   "title": "Attack",
   "description": "A host performing an attack.",
   "properties": {
       "report_category": {
           "title": "Report category: attack",
           "description": "The category of the report: an attack on a system.",
           "type": "string",
           "enum": ["eu.acdc.attack"]
       },
       "report_type": {
           "title": "Report type",
           "description": "The type of the report. This is a free text field 
characterising the report that should be used for a human readable description 
rather than for automatic processing. As a rule of thumb, this should not be longer 
than one sentence.",
           "type": "string"
       },
       "timestamp": {
           "title": "Time of the attack observation",
           "description": "The timestamp when the attack took place.",
           "type": "string",
           "format": "date-time"
       },
       "source_key": {
           "title": "Type of the reported object: IP",
           "description": "The type of the reported object: an IP.",
           "type": "string",
           "enum": ["ip"]
       },
       "source_value": {
           "title": "Attacking IP",
           "description": "The IP of the system performing the attack.",
           "type": "string"
       },
       "confidence_level": {
           "title": "Confidence level of the report",
           "description": "The level of confidence put into the accuracy of the 
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being 
verified to be accurate.",
           "type": "number",
           "minimum": 0.0,
           "maximum": 1.0
       },
       "version": {
           "title": "Version of the format: 1",
           "description": "The version number of the data format used for the 
report: Version 1.",
           "type": "integer",
           "enum": [2]
       },
       "report_subcategory": {
           "title": "Attack category",
           "description": "The type of attack performed.",
           "type": "string",
           "enum": ["abuse", "abuse.spam", "compromise", "data", "dos", "dos.dns", 
"dos.http", "dos.tcp", "dos.udp", "login", "malware", "scan", "other"]
```

```
        },
        "ip_protocol_number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number of
the attack connection.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "ip_version": {
            "title": "IP version number",
            "description": "The IP version of the attack connection.",
            "type": "integer",
            "enum": [4, 6]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by the
CCH and is thus overwritten on import.",
            "type": "string"
        },
        "duration": {
            "title": "The duration of the attack",
            "description": "The duration of the attack in seconds.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet responsible for attack",
            "description": "The botnet this attack can be attributed to. This
references a separate eu.acdc.botnet report.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows putting
more specific information into a report on a case by case basis in a structured
manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate format.
This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "src_ip_v4": {
            "title": "Source IPv4 of the attack",
            "description": "The source IPv4 of the attack connection. This is always
the IP of the attacking system (i.e., the one identified by source_value). This
field equals source_value.",
            "type": "string",
            "format": "ipv4"
```

```
       },
       "src_ip_v6": {
           "title": "Source IPv6 of the attack",
           "description": "The source IPv6 of the attack connection. This is always
the IP of the attacking system (i.e., the one identified by source_value). This
field equals source_value.",
           "type": "string",
           "format": "ipv6"
       },
       "src_mode": {
           "title": "Source IP mode",
           "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
           "type": "string",
           "enum": ["plain", "anon", "pseudo"]
       },
       "dst_ip_v4": {
           "title": "Destination IPv4 of the attack",
           "description": "The destination IPv4 of the attack connection. This is
always the IP of the attacked system.",
           "type": "string",
           "format": "ipv4"
       },
       "dst_ip_v6": {
           "title": "Destination IPv6 of the attack",
           "description": "The destination IPv6 of the attack connection. This is
always the IP of the attacked system.",
           "type": "string",
           "format": "ipv6"
       },
       "dst_mode": {
           "title": "Destination IP mode",
           "description": "The mode of the destination IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
           "type": "string",
           "enum": ["plain", "anon", "pseudo"]
       },
       "src_port": {
           "title": "Source port of the attack",
           "description": "The source port of the attack connection. This is always
the port on the attacking system (i.e., the one identified by source_value).",
           "type": "integer"
       },
       "dst_port": {
           "title": "Destination port of the attack",
           "description": "The destination port of the attack connection. This is
always the port on the attacked system.",
           "type": "integer"
       },
       "application_protocol": {
           "title": "Application protocol",
           "description": "The application protocol used for the connection.",
           "type": "string"
       },
       "sample_filename": {
           "title": "Filename of the payload",
           "description": "The filename used for the payload that the attack tried
to install or run on the attacked system. This should only be used if the payload
is uploaded to the attacked system directly. Otherwise, malicious_uri should be
used to link this report to an eu.acdc.malicious_uri report that in turn contains
the SHA256 hash.",
           "type": "string"
       },
       "sample_sha256": {
           "title": "Hash of the payload",
           "description": "The SHA256 hash of the payload that the attack tried to
install or run on the attacked system. This should only be used if the payload is
uploaded to the attacked system directly. Otherwise, malicious_uri should be used
```

```
to link this report to an eu.acdc.malicious_uri report that in turn contains the
SHA256 hash.",
            "type": "string"
        },
        "malicious_uri": {
            "title": "URI of the payload",
            "description": "The URI of the payload in the wild that the attack tried
to install or run on the attacked system. This can for example be the location of a
malware offered as a download or a webshell offered as a remote include during an
attack.",
            "type": "string",
            "format": "uri"
        },
        "subject_text": {
            "title": "Subject of spam email",
            "description": "The subject of an email sent in a report of subcategory
abuse.spam. Varying parts, especially personal information like names or email
addresses, must be replaced with the placeholder . This references a separate
eu.acdc.spam_campaign report.",
            "type": "string"
        },
        "mail_header": {
            "title": "Email header",
            "description": "The header of the email.",
            "type": "string"
        },
        "bit_rate": {
            "title": "Bits per second of traffic",
            "description": "The number of bits per second of traffic transferred,
for example in a DoS attack.",
            "type": "integer"
        },
        "packet_rate": {
            "title": "Packets per second of traffic",
            "description": "The number of packets per second of traffic transferred,
for example in a DoS attack.",
            "type": "integer"
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version", "report_subcategory",
"ip_protocol_number", "ip_version"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"],
        "dst_ip_v4": ["dst_mode"],
        "dst_ip_v6": ["dst_mode"]
    },
    "additionalProperties": false,
    "oneOf": [
        {
            "title": "An IPv4 attack",
            "properties": {
                "ip_version": {
                    "enum": [4]
                }
            },
            "required": ["src_ip_v4", "src_mode"]
        },
        {
            "title": "An IPv6 attack",
            "properties": {
                "ip_version": {
                    "enum": [6]
                }
            },
            "required": ["src_ip_v6", "src_mode"]
        }
    ]
```

```
}
```

C.2. Bot (eu.acdc.bot)

```
{
    "title": "A bot",
    "description": "A system infected with a bot.",
    "properties": {
        "report_category": {
            "title": "Report category: a bot",
            "description": "The category of the report: a bot infection.",
            "type": "string",
            "enum": ["eu.acdc.bot"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the bot infection was observed.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: IP",
            "description": "The type of the reported object: an IP address.",
            "type": "string",
            "enum": ["ip"]
        },
        "source_value": {
            "title": "Infected IP",
            "description": "The IP address of the infected system.",
            "type": "string"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        },
        "report_subcategory": {
            "title": "Report subcategory",
            "description": "The type of the bot.",
            "type": "string",
            "enum": ["fast_flux", "other"]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "duration": {
            "title": "Duration of the bot observation",
            "description": "The duration in seconds during which the bot infection
was observed. This can be the timespan during which connections to the C2 server
were observed.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet infection is attributed to",
            "description": "The botnet the bot is attributed to.",
            "type": "string"
        },
        "bot_id": {
            "title": "Identifier of the bot",
            "description": "The identifier the botnet uses for this bot. Not all
botnets have this concept. Since bot IDs are only meaningful in the context of a
botnet, a report containing a bot_id should contain a botnet field as well if
possible.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "ip_version": {
            "title": "IP version number",
            "description": "The IP version of the infected sytem's IP address.",
            "type": "integer",
            "enum": [4, 6]
        },
        "ip_protocol_number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number of
the C2 connection.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "src_ip_v4": {
            "title": "Source IPv4 of the bot",
            "description": "The source IPv4 of the bot infected system. This field
equals source_value.",
            "type": "string",
            "format": "ipv4"
        },
        "src_ip_v6": {
            "title": "Source IPv6 of the bot",
            "description": "The source IPv6 of the bot infected system. This field
equals source_value.",
            "type": "string",
            "format": "ipv6"
        },
        "src_mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"]
        },
        "src_port": {
            "title": "Source port of the C2 connection",
            "description": "The source port of the connection from the bot to the
C2 server. This is always the port on the bot infected system.",
            "type": "integer"
        },
        "c2_ip_v4": {
            "title": "IPv4 of the C2",
            "description": "The IPv4 of the C2 server.",
            "type": "string",
            "format": "ipv4"
        },
        "c2_ip_v6": {
            "title": "IPv6 of the C2",
            "description": "The IPv6 of the C2 server.",
            "type": "string",
            "format": "ipv6"
        },
        "c2_mode": {
            "title": "C2 IP mode",
            "description": "The mode of the C2 server IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"]
        },
        "c2_port": {
            "title": "C2 port of the C2 connection",
            "description": "The port of the C2 connection on the C2 server.",
            "type": "integer"
        },
        "sample_sha256": {
            "title": "Hash of the malware",
            "description": "The SHA256 hash of the malware the system is infected
with. This references a separate eu.acdc.malware report.",
            "type": "string"
        },
        "fast_flux_uri": {
            "title": "URI of the fast flux domain",
            "description": "The URI of the fast flux domain resolving to this
bot.",
            "type": "string",
            "format": "uri"
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version", "report_subcategory"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"],
        "c2_ip_v4": ["c2_mode"],
        "c2_ip_v6": ["c2_mode"]
    },
    "additionalProperties": false,
    "oneOf": [
        {
            "title": "An IPv4 bot",
            "properties": {
                "ip_version": {
                    "enum": [4]
                }
            },
            "required": ["src_ip_v4", "src_mode"]
        },
        {
            "title": "An IPv6 bot",
            "properties": {
                "ip_version": {
                    "enum": [6]
                }
            },
            "required": ["src_ip_v6", "src_mode"]
        }
    ]
}
```

C.3. Botnet (eu.acdc.botnet)

```
{
    "title": "Botnet",
    "description": "A botnet tracked by ACDC.",
    "properties": {
        "report_category": {
            "title": "Report category: botnet",
            "description": "The category of the report: a botnet.",
            "type": "string",
            "enum": ["eu.acdc.botnet"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "source_key": {
            "title": "Type of the reported object: botnet",
            "description": "The type of the reported object: a botnet.",
            "type": "string",
            "enum": ["botnet"]
        },
        "source_value": {
            "title": "Botnet",
            "description": "The identifier of the botnet. This can be the name of a
single type of botnet or a combination of a botnet type and an identifier for a
specific instance of the botnet.",
            "type": "string"
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [1]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "report_subcategory": {
            "title": "Botnet category",
            "description": "The category of the botnet.",
            "type": "string",
            "enum": ["c2", "p2p", "other"]
        }
    },
    "required": ["report_category", "report_type", "source_key", "source_value",
"version", "report_subcategory"],
    "additionalProperties": false
}
```

C.4. C2 server (eu.acdc.c2_server)

```
{
    "title": "A C2 server",
    "description": "A command and control server.",
    "properties": {
        "report_category": {
            "title": "Report category: C2",
            "description": "The category of the report: a C2 server.",
            "type": "string",
            "enum": ["eu.acdc.c2_server"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the C2 server was observed.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: IP",
            "description": "The type of the reported object: an IP address.",
            "type": "string",
            "enum": ["ip"]
        },
        "source_value": {
            "title": "C2 IP",
            "description": "The IP address of the C2 server.",
            "type": "string"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        },
        "report_subcategory": {
            "title": "C2 subcategory",
            "description": "The control channel used by the C2.",
            "type": "string",
            "enum": ["http", "irc", "other"]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "duration": {
            "title": "Duration of the C2 observation",
            "description": "The duration in seconds during which the C2 server was
observed. This can be the timespan during which connections to the C2 server were
successful.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet C2 is attributed to",
            "description": "The botnet the C2 server is attributed to.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "ip_version": {
            "title": "IP version number",
            "description": "The IP version of the C2 server's IP address.",
            "type": "integer",
            "enum": [4, 6]
        },
        "ip_protocol_number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number used
for C2 connections.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "c2_ip_v4": {
            "title": "IPv4 of the C2",
            "description": "The IPv4 of the C2 server.",
            "type": "string",
            "format": "ipv4"
        },
        "c2_ip_v6": {
            "title": "IPv6 of the C2",
            "description": "The IPv6 of the C2 server.",
            "type": "string",
            "format": "ipv6"
        },
        "c2_mode": {
            "title": "C2 IP mode",
            "description": "The mode of the C2 server IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"]
        },
        "c2_port": {
            "title": "C2 Port of C2 connections",
            "description": "The port of C2 connections on the C2 server.",
            "type": "integer"
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version", "report_subcategory"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"],
        "c2_ip_v4": ["c2_mode"],
        "c2_ip_v6": ["c2_mode"]
    },
    "additionalProperties": false,
    "oneOf": [
        {
            "title": "An IPv4 C2",
            "properties": {
                "ip_version": {
                    "enum": [4]
                }
            },
            "required": ["c2_ip_v4", "c2_mode"]
        },
        {
            "title": "An IPv6 C2",
            "properties": {
                "ip_version": {
                    "enum": [6]
                }
            },
            "required": ["c2_ip_v6", "c2_mode"]
        }
    ]
}
```

C.5. Fast Flux Domain (eu.acdc.fast_flux)

```
{
    "title": "Fast Flux Domain",
    "description": "A fast flux domain.",
    "properties": {
        "report_category": {
            "title": "Report category: Fast flux",
            "description": "The category of the report: a fast flux domain.",
            "type": "string",
            "enum": ["eu.acdc.fast_flux"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the fast flux domain was first
observed. If the report contains IPs, this is typically the earliest timestamp of
the IPs that the domain resolves to.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: domain",
            "description": "The type of the reported object: a domain URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "source_value": {
            "title": "Fast flux domain",
            "description": "The fast flux domain URI.",
            "type": "string",
            "format": "uri"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "duration": {
            "title": "Duration of the observation",
            "description": "The duration of the observation in seconds. If the
report contains IPs, this is typically the difference between the earliest and the
latest timestamp of the IPs that the domain resolves to.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet of the fast flux domain",
            "description": "The botnet this fast flux domain can be attributed to.
This references a separate eu.acdc.botnet report.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"]
    },
    "additionalProperties": false
}
```

C.6. Malicious URI (eu.acdc.malicious_uri)

```
{
    "title": "Malicious URI",
    "description": "A URI pointing to malicious content.",
    "properties": {
        "report_category": {
            "title": "Report category: malicious URI",
            "description": "The category of the report: a malicious URI.",
            "type": "string",
            "enum": ["eu.acdc.malicious_uri"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the malicious URI was observed.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: URI",
            "description": "The type of the reported object: a URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "source_value": {
            "title": "Malicious URI",
            "description": "The URI to the malicious content.",
            "type": "string",
            "format": "uri"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format: 1",
            "description": "The version number of the data format used for the
report: Version 1.",
            "type": "integer",
            "enum": [2]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "report_subcategory": {
            "title": "Malicious category",
            "description": "The type of the malicious content at the URI.",
            "type": "string",
            "enum": ["exploit", "malware", "phishing", "other"]
        },
        "duration": {
            "title": "Duration of the reported observation",
            "description": "The duration in seconds during which the malicious URI
was observed.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet of the malicious URI",
            "description": "The botnet this malicious URI can be attributed to.
This references a separate eu.acdc.botnet report.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate 
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "ip_version": {
            "title": "IP version number",
            "description": "The IP version of the IP address belonging to the 
malicious URI.",
            "type": "integer",
            "enum": [4, 6]
        },
        "src_ip_v4": {
            "title": "Source IPv4 of the URI",
            "description": "The source IPv4 associated with the malicious URI.",
            "type": "string",
            "format": "ipv4"
        },
        "src_ip_v6": {
            "title": "Source IPv6 of the URI",
            "description": "The source IPv6 associated with the malicious URI.",
            "type": "string",
            "format": "ipv6"
        },
        "src_mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for 
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"]
        },
        "sample_filename": {
            "title": "Malicious content file name",
            "description": "The file name of the malicious content if applicable.",
            "type": "string"
        },
        "sample_sha256": {
            "title": "Hash of the malicious content",
            "description": "The SHA256 hash of the malicious content if 
applicable.",
            "type": "string"
        },
        "exploits": {
            "title": "Exploits in the URI",
            "description": "Exploits discovered in the analysed URI. This is an 
array of objects, each giving an identifier scheme like CVE and an identifier for 
the actual exploit found.",
            "type": "array",
            "items": {
                "title": "Exploit",
                "description": "Indicator for the type and identifier of the given 
exploit",
                "type": "object",
                "properties": {
                    "type": {
                        "title": "Identifier scheme",
                        "description": "Indicate whether a CVE was matched or a 
heuristic identified the exploit",
                        "type": "string",
                        "enum": ["cve", "other"]
                    },
                    "value": {
                        "title": "Exploit identifier",
                        "description": "An indicator for the specific CVE or 
heuristic that was triggered by the file",
                        "type": "string"
                    }
                }
            }
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version", "report_subcategory"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"]
    },
    "additionalProperties": false,
    "anyOf": [
        {
            "title": "An IPv4 connected URI",
            "properties": {
                "ip_version": {
                    "enum": [4]
                }
            },
            "dependencies": {
                "ip_version": ["src_ip_v4", "src_mode"]
            }
        },
        {
            "title": "An IPv6 connected URI",
            "properties": {
                "ip_version": {
                    "enum": [6]
                }
            },
            "dependencies": {
                "ip_version": ["src_ip_v6", "src_mode"]
            }
        }
    ]
}
```

C.7. Malware Sample (eu.acdc.malware)

```
{
    "title": "Malware Sample",
    "description": "A sample of a malware.",
    "properties": {
        "report_category": {
            "title": "Report category: eu.acdc.malware",
            "description": "The category of the report: a malware sample.",
            "type": "string",
            "enum": ["eu.acdc.malware"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the sample was obtained.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: malware",
            "description": "The type of the reported object: a malware sample.",
            "type": "string",
            "enum": ["malware"]
        },
        "source_value": {
            "title": "Malware SHA256",
            "description": "The SHA256 hash of the malware sample.",
            "type": "string"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "botnet": {
            "title": "Botnet of the sample",
            "description": "The botnet the sample is attributed to. This references
a separate eu.acdc.botnet report.",
            "type": "string"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "sample_b64": {
            "title": "Source of the sample.",
            "description": "The source code of the sample encoded in Base64. Only
to be used for data not inluding personal information.",
            "type": "string"
        },
        "mime_type": {
            "title": "MIME type",
            "description": "The MIME type of the sample.",
            "type": "string"
        },
        "cpe": {
            "title": "CPE of the affected platform",
            "description": "The full or partial CPE name binding of the platform 
affected by the malware.",
            "type": "string"
        },
        "sample_hashes": {
            "title": "Hashes of the sample",
            "description": "An array of objects containing hashes for the sample. 
Each item gives a hash function and the corresponding hash value.",
            "type": "array",
            "items": {
                "title": "Hash function and value",
                "description": "Identifier for the cryptographic hash function and 
the value it generated for the given file",
                "type": "object",
                "properties": {
                    "type": {
                        "title": "Hash function identifier",
                        "type": "string",
                        "enum": ["md5", "sha1", "sha256", "sha512", "sha3-256", 
"sha3-512"]
                    },
                    "value": {
                        "title": "Hash value",
                        "type": "string"
                    }
                }
            }
        },
        "exploits": {
            "title": "Exploits in the sample",
            "description": "Exploits discovered in the analysed sample. This is an 
array of objects, each giving an identifier scheme like CVE and an identifier for 
the actual exploit found.",
            "type": "array",
            "items": {
                "title": "Exploit",
                "description": "Indicator for the type and identifier of the given 
exploit",
                "type": "object",
                "properties": {
                    "type": {
                        "title": "Identifier scheme",
                        "description": "Indicate whether a CVE was matched or a 
heuristic identified the exploit",
                        "type": "string",
                        "enum": ["cve", "other"]
                    },
                    "value": {
                        "title": "Exploit identifier",
                        "description": "An indicator for the specific CVE or 
heuristic that was triggered by the file",
                        "type": "string"
                    }
                }
            }
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key", 
"source_value", "confidence_level", "version"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"]
    },
    "additionalProperties": false
```

```
}
```

C.8. Spam Campaign (eu.acdc.spam_campaign)

```
{
   "title": "Spam Campaign",
   "description": "A spam campaign.",
   "properties": {
       "report_category": {
           "title": "Report category: Spam Campaign",
           "description": "The category of the report: a spam campaign.",
           "type": "string",
           "enum": ["eu.acdc.spam_campaign"]
       },
       "report_type": {
           "title": "Report type",
           "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
           "type": "string"
       },
       "timestamp": {
           "title": "Start of the spam campaign",
           "description": "The timestamp when the spam campaign was first
observed.",
           "type": "string",
           "format": "date-time"
       },
       "source_key": {
           "title": "Type of the reported object: subject",
           "description": "The type of the reported object: an email subject.",
           "type": "string",
           "enum": ["subject"]
       },
       "source_value": {
           "title": "Campaign subject",
           "description": "The common subject of the spam campaign. Varying parts,
especially personal information like names or email addresses, must be replaced
with the placeholder .",
           "type": "string"
       },
       "confidence_level": {
           "title": "Confidence level of the report",
           "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
           "type": "number",
           "minimum": 0.0,
           "maximum": 1.0
       },
       "version": {
           "title": "Version of the format",
           "description": "The version number of the data format used for the
report.",
           "type": "integer",
           "enum": [2]
       },
       "report_subcategory": {
           "title": "Spam category",
           "description": "The type of spam messages sent.",
           "type": "string",
           "enum": ["advance_fee", "malware", "mule", "phishing", "product",
"product.casino", "product.contact", "product.pharmacy", "product.replica",
"stock", "other"]
       },
       "report_id": {
           "title": "Report ID",
           "description": "The ID of the report in the CCH. This will be set by the
CCH and is thus overwritten on import.",
           "type": "string"
       },
       "duration": {
           "title": "Duration of the spam campaign",
           "description": "Duration during which the spam campaign was observed.",
           "type": "integer",
           "minimum": 0
       },
       "reported_at": {
           "title": "Time of the report's submission",
           "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
           "type": "string",
           "format": "date-time"
       },
       "botnet": {
           "title": "Botnet responsible for campaign",
           "description": "The botnet the spam campaign can be attributed to. This
references a separate eu.acdc.botnet report.",
           "type": "string"
       },
       "additional_data": {
           "title": "Additional data",
           "description": "Additional data for the observation. This allows putting
more specific information into a report on a case by case basis in a structured
manner. The usage of this field is at the data providers discretion.",
           "type": "object"
       },
       "alternate_format_type": {
           "title": "Type of the alternate format",
           "description": "The type of the alternate format description of the
observation.",
           "type": "string",
           "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
       },
       "alternate_format": {
           "title": "Alternate format description of the observation",
           "description": "A description of the observation in an alternate format.
This is used to submit complex structured formats like IDMEF to the CCH.",
           "type": "string"
       },
       "mail_body": {
           "title": "Email body",
           "description": "The body of the email. Varying parts, especially
personal information like names or email addresses, must be replaced with the
placeholder '{}'.",
           "type": "string"
       },
       "sample_filename": {
           "title": "Malicious attachment file name",
           "description": "The file name of the malicious attachment used in this
campaign.",
           "type": "string"
       },
       "sample_sha256": {
           "title": "Campaign malware",
           "description": "The SHA256 of the malware distributed with this
campaign. This references a separate eu.acdc.malware report.",
           "type": "string"
       },
       "malicious_uri": {
           "title": "Campaign URI",
           "description": "The URI advertised with this campaign. This references a
separate eu.acdc.malicious_uri report.",
           "type": "string",
           "format": "uri"
       }
   },
   "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version"],
   "dependencies": {
       "alternate_format": ["alternate_format_type"]
   },
   "anyOf": [
       {
           "title": "A malware distributing campaign",
           "required": ["sample_sha256"]
       },
       {
           "title": "A campaign linking to a malicious URI",
           "required": ["malicious_uri"]
       }],
   "additionalProperties": false
}
```

C.9. Vulnerable URI (eu.acdc.vulnerable_uri)

```
{
    "title": "Vulnerable URI",
    "description": "A URI pointing to a vulnerable resource.",
    "properties": {
        "report_category": {
            "title": "Report category: vulnerable URI",
            "description": "The category of the report: a vulnerable URI",
            "type": "string",
            "enum": ["eu.acdc.vulnerable_uri"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb, this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when vulnerable URI was observed.",
            "type": "string",
            "format": "date-time"
        },
        "source_key": {
            "title": "Type of the reported object: URI",
            "description": "The type of the reported object: a URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "source_value": {
            "title": "Vulnerable URI",
            "description": "The URI to the vulnerable resource.",
            "type": "string",
            "format": "uri"
        },
        "confidence_level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        },
        "src_ip_v4": {
            "title": "Source IPv4 of the URI",
            "description": "The source IPv4 associated with the malicious URI.",
            "type": "string",
            "format": "ipv4"
        },
        "src_ip_v6": {
            "title": "Source IPv6 of the URI",
            "description": "The source IPv6 associated with the malicious URI.",
            "type": "string",
            "format": "ipv6"
        },
        "src_mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"]
        },
        "vulnerabilities": {
            "title": "Vulnerabilities discovered at the URI",
            "description": "An array of objects describing vulnerabilities
discovered at the vulnerable URI.",
            "type": "array",
            "items": {
                "title": "Vulnerability",
                "description": "Indicator for the type and identifier of the given
vulnerability",
                "type": "object",
                "properties": {
                    "type": {
                        "title": "Identifier scheme",
                        "description": "Indicate whether a CVE or CWE was matched
or a heuristic identified the vulnerability",
                        "type": "string",
                        "enum": ["cve", "cwe", "other"]
                    },
                    "value": {
                        "title": "Vulnerability identifier",
                        "description": "An indicator for the specific CVE, CWE, or
heuristic that was triggered by the URI",
                        "type": "string"
                    }
                }
            }
        },
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "duration": {
            "title": "Duration of the reported observation",
            "description": "The duration in seconds during which the
vulnerabilities at the URI were observed.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "additional_data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object"
        },
        "alternate_format_type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"]
        },
        "alternate_format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format. This is used to submit complex structured formats like IDMEF to the CCH.",
            "type": "string"
        },
        "ip_version": {
            "title": "IP version number",
            "description": "The IP version of the IP address belonging to the
vulnerable URI.",
            "type": "integer",
            "enum": [4, 6]
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_key",
"source_value", "confidence_level", "version", "vulnerabilities"],
    "dependencies": {
        "alternate_format": ["alternate_format_type"]
    },
    "additionalProperties": false,
    "anyOf": [
        {
            "title": "An IPv4 connected URI",
            "properties": {
                "ip_version": {
                    "enum": [4]
                }
            },
            "dependencies": {
                "ip_version": ["src_ip_v4", "src_mode"]
            }
        },
        {
            "title": "An IPv6 connected URI",
            "properties": {
                "ip_version": {
                    "enum": [6]
                }
            },
            "dependencies": {
                "ip_version": ["src_ip_v6", "src_mode"]
            }
        }
    ]
}
```

C.10.Field Schemata

```
{
"additional_data": {
   "title": "Additional data",
   "description": "Additional data for the observation. This allows putting more
specific information into a report on a case by case basis in a structured manner.
The usage of this field is at the data providers discretion.",
   "type": "object"
},
"alternate_format": {
   "title": "Alternate format description of the observation",
   "description": "A description of the observation in an alternate format. This is
used to submit complex structured formats like IDMEF to the CCH.",
   "type": "string"
},
"alternate_format_type": {
   "title": "Type of the alternate format",
   "description": "The type of the alternate format description of the
observation.",
   "type": "string",
   "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9", "OpenIOC",
"sFlow", "STIX"]
},
"application_protocol": {
   "title": "Application protocol",
   "description": "The application protocol used for the connection.",
   "type": "string"
},
"bit_rate": {
   "title": "Bits per second of traffic",
   "description": "The number of bits per second of traffic transferred.",
   "type": "integer"
},
"bot_id": {
   "title": "Identifier of the bot",
   "description": "The identifier the botnet uses for this bot. Not all botnets
have this concept. Since bot IDs are only meaningful in the context of a botnet, a
report containing a bot_id should contain a botnet field as well if possible.",
   "type": "string"
},
"botnet": {
   "title": "Botnet observation is attributed to",
   "description": "The botnet the observation is attributed to. This can for
example be the botnet a malware joins, the botnet that sends a spam campaign, or
the botnet a bot belongs to.",
   "type": "string"
},
"c2_ip_v4": {
   "title": "IPv4 of the C2",
   "description": "The IPv4 of the C2 server.",
   "type": "string",
   "format": "ipv4"
},
"c2_ip_v6": {
   "title": "IPv6 of the C2",
   "description": "The IPv6 of the C2 server.",
   "type": "string",
   "format": "ipv6"
},
"c2_mode": {
   "title": "C2 IP mode",
   "description": "The mode of the C2 server IP. This can be plain for unaltered
IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
   "type": "string",
   "enum": ["plain", "anon", "pseudo"]
},
"c2_port": {
   "title": "Port of the C2 connection",
   "description": "The port of the connection to the C2 server.",
   "type": "integer"
},
"confidence_level": {
   "title": "Confidence level of the report",
   "description": "The level of confidence put into the accuracy of the report. A
number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be
accurate.",
   "type": "number",
   "minimum": 0.0,
   "maximum": 1.0
},
"cpe": {
   "title": "CPE of the affected platform",
   "description": "The full or partial CPE name binding of the platform affected by
the report.",
   "type": "string"
},
"credentials": {
   "title": "Credentials",
   "description": "The credentials used for example in an attack to brute force a
login. This is a list of pairs. Each pair consists of a user name and a password as
strings.",
   "type": "array",
   "items": {
       "title": "Pair of user name and password",
       "type": "array",
       "items": {
           "type": "string"
       },
       "minItems": 2,
       "maxItems": 2
   }
},
"dst_access_type": {
   "title": "Access type of the destination IP",
   "description": "The type of access network used by the destination IP. Mobile
signifies address spaces assigned to mobile access technologies like 3G or 4G.
Fixed signifies address spaces assigned to fixed access technologies like xDSL or
FTTH.",
   "type": "string",
   "enum": ["mobile", "fixed"]
},
"dst_ip_v4": {
   "title": "Destination IPv4 of the connection",
   "description": "The destination IPv4 of the connection. This is always the
remote IP from the perspective of the reported system (i.e., the one identified by
source_value). It can for example be the IP of a honeypot that was contacted to
infect it.",
   "type": "string",
   "format": "ipv4"
},
"dst_ip_v6": {
   "title": "Destination IPv6 of the connection",
   "description": "The destination IPv6 of the connection. This is always the
remote IP from the perspective of the reported system (i.e., the one identified by
source_value). It can for example be the IP of a honeypot that was contacted to
infect it.",
   "type": "string",
   "format": "ipv6"
},
"dst_mode": {
   "title": "Destination IP mode",
   "description": "The mode of the destination IP. This can be plain for unaltered
IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
   "type": "string",
   "enum": ["plain", "anon", "pseudo"]
},
"dst_port": {
   "title": "Destination port of the connection",
   "description": "The destination port of the connection. This is always the 
remote port from the perspective of the reported system (i.e., the one identified 
by source_value). It can for example be the port of a honeypot that was contacted 
to infect it.",
   "type": "integer"
},
"duration": {
   "title": "Duration of the observation",
   "description": "The duration of the observation in seconds. This can for example 
be the duration of DoS attack.",
   "type": "integer",
   "minimum": 0
},
"exploits": {
   "title": "Exploits in the sample",
   "description": "Exploits discovered in the analysed sample. This is an array of 
objects, each giving an identifier scheme like CVE and an identifier for the actual 
exploit found.",
   "type": "array",
   "items": {
       "title": "Exploit",
       "description": "Indicator for the type and identifier of the given exploit",
       "type": "object",
       "properties": {
           "type": {
               "title": "Identifier scheme",
               "description": "Indicate whether a CVE was matched or a heuristic 
identified the exploit",
               "type": "string",
               "enum": ["cve", "other"]
           },
           "value": {
               "title": "Exploit identifier",
               "description": "An indicator for the specific CVE or heuristic that 
was triggered by the file",
               "type": "string"
           }
       }
   }
},
"fast_flux_uri": {
   "title": "URI of the fast flux domain",
   "description": "The URI of the fast flux domain connected to this report.",
   "type": "string",
   "format": "uri"
},
"http_request": {
   "title": "HTTP request",
   "description": "The HTTP request observed. This can be for example a request 
some attacking machine sent to a sensor or a request a bot sent to a C2 server to 
query new commands.",
   "type": "string"
},
"ip_protocol_number": {
   "title": "IP protocol number",
   "description": "The IANA assigned decimal internet protocol number of the 
connection.",
   "type": "integer",
   "minimum": 0,
   "maximum": 255
},
"ip_version": {
   "title": "IP version number",
   "description": "The IP version of the connection.",
   "type": "integer",
   "enum": [4, 6]
```

```
},
"mail_body": {
   "title": "Email body",
   "description": "The body of the email. Varying parts, especially personal
information like names or email addresses, must be replaced with the placeholder
'{}'.",
   "type": "string"
},
"mail_header": {
   "title": "Email header",
   "description": "The header of the email.",
   "type": "string"
},
"mime_type": {
   "title": "MIME type",
   "description": "The MIME type of an object. This can for example be the MIME
type of a malware sample.",
   "type": "string"
},
"malicious_uri": {
   "title": "URI of malicious content",
   "description": "The URI where the malicious content can be found in the wild
like the location of a malware supposed to be downloaded as part of an attack.",
   "type": "string",
   "format": "uri"
},
"packet_rate": {
   "title": "Packets per second of traffic",
   "description": "The number of packets per second of traffic transferred.",
   "type": "integer"
},
"report_category": {
   "title": "Report category",
   "description": "The category of the report. This links the report to one of
ACDC's schemata. A report category has the format .",
   "type": "string"
},
"report_id": {
   "title": "Report ID",
   "description": "The ID of the report in the CCH. This will be set by the CCH and
is thus overwritten on import.",
   "type": "string"
},
"report_subcategory": {
   "title": "Report subcategory",
   "description": "The subcategory of the report. This is used to categorise
different types of similar reports that have mostly the same fields. It is defined
as an enum in the schema of the report category.",
   "type": "string"
},
"report_type": {
   "title": "Report type",
   "description": "The type of the report. This is a free text field characterising
the report that should be used for a human readable description rather than for
automatic processing. As a rule of thumb, this should not be longer than one
sentence.",
   "type": "string"
},
"reported_at": {
   "title": "Time of the report's submission",
   "description": "The timestamp when the report was submitted to the CCH. This
will be set by the CCH and is thus overwritten on import.",
   "type": "string",
   "format": "date-time"
},
"sample_b64": {
   "title": "Source of the sample",
   "description": "The source code of the sample encoded in Base64. Only to be used
for data not including personal information.",
   "type": "string"
},
"sample_filename": {
   "title": "Filename of the sample",
   "description": "The filename used for the sample like the name of an attachment
to an email or an upload to a honeypot.",
   "type": "string"
},
"sample_hashes": {
   "title": "Hashes of the sample",
   "description": "A list of hashed for the sample, each giving a hash function and
the corresponding hash value for the sample. This is used as information on a
specific sample.",
   "type": "array",
   "items": {
       "title": "Hash function and value",
       "description": "Identifier for the cryptographic hash function and the value
it generated for the given file",
       "type": "object",
       "properties": {
           "type": {
               "title": "Hash function identifier",
               "type": "string",
               "enum": ["md5", "sha1", "sha256", "sha512", "sha3-256", "sha3-512"]
           },
           "value": {
               "title": "Hash value",
               "type": "string"
           }
       }
   }
},
"sample_sha256": {
   "title": "SHA256 of the sample",
   "description": "The SHA256 hash of the sample. This is used to reference a
specific sample.",
   "type": "string"
},
"source_key": {
   "title": "Type of the reported object",
   "description": "The type of the reported object.",
   "type": "string",
   "enum": ["botnet", "ip", "malware", "subject", "uri"]
},
"source_value": {
   "title": "Identifier of the reported object",
   "description": "The identifier of the reported object like its IP address or
URI.",
   "type": "string"
},
"src_access_type": {
   "title": "Access type of the source IP",
   "description": "The type of access network used by the source IP. Mobile
signifies address spaces assigned to mobile access technologies like 3G or 4G.
Fixed signifies address spaces assigned to fixed access technologies like xDSL or
FTTH.",
   "type": "string",
   "enum": ["mobile", "fixed"]
},
"src_ip_v4": {
   "title": "Source IPv4 of the connection",
   "description": "The source IPv4 of the connection. This is always the IP of the
reported system (i.e., the one identified by source_value).",
   "type": "string",
   "format": "ipv4"
},
"src_ip_v4s": {
   "title": "Source IPv4s and timestamps",
   "description": "A list of IPv4 source addresses together with timestamps
associated to the observation. This can for example be the IPs a fast flux domain
resolves to.",
   "type": "array",
   "items": {
       "type": "object",
       "properties": {
           "src_ip_v4": {
               "title": "Fast flux IPv4",
               "type": "string",
               "format": "ipv4"
           },
           "timestamp": {
               "title": "Timestamp of domain resolving to IP",
               "type": "string",
               "format": "date-time"
           }
       }
   }
},
"src_ip_v6": {
   "title": "Source IPv6 of the connection",
   "description": "The source IPv6 of the connection. This is always the IP of the
reported system (i.e., the one identified by source_value).",
   "type": "string",
   "format": "ipv6"
},
"src_ip_v6s": {
   "title": "Source IPv6s and timestamps",
   "description": "A list of IPv6 source addresses together with timestamps
associated to the observation. This can for example be the IPs a fast flux domain
resolves to.",
   "type": "array",
   "items": {
       "type": "object",
       "properties": {
           "src_ip_v6": {
               "title": "Fast flux IPv6",
               "type": "string",
               "format": "ipv6"
           },
           "timestamp": {
               "title": "Timestamp of domain resolving to IP",
               "type": "string",
               "format": "date-time"
           }
       }
   }
},
"src_mode": {
   "title": "Source IP mode",
   "description": "The mode of the source IP. This can be plain for unaltered IPs,
anon for anonymised IPs, or pseudo for pseudonymised IPs.",
   "type": "string",
   "enum": ["plain", "anon", "pseudo"]
},
"src_port": {
   "title": "Source port of the connection",
   "description": "The source port of the connection. This is always the port on
the reported system (i.e., the one identified by source_value).",
   "type": "integer"
},
"subject_text": {
   "title": "Subject of the email",
   "description": "The subject text of the email. Varying parts, especially
personal information like names or email addresses, must be replaced with the
placeholder .",
   "type": "string"
},
"timestamp": {
   "title": "Time of the reported observation",
   "description": "The timestamp when the reported observation took place. This can
for example be when an attack occurred, when a malware hosting was observed, or
when a compromise took place according to log files.",
   "type": "string",
   "format": "date-time"
},
"version": {
   "title": "Version of the format",
   "description": "The version number of the data format used for the report.",
   "type": "integer",
   "minimum": 1
},
"vulnerabilities": {
   "title": "Vulnerabilities in a system",
   "description": "Vulnerabilities discovered in an analysed system. This is an
array of objects, each giving an identifier scheme like CWE and an identifier for
the actual vulnerability found.",
   "type": "array",
   "items": {
       "title": "Vulnerability",
       "description": "Indicator for the type and identifier of the given
vulnerability",
       "type": "object",
       "properties": {
           "type": {
               "title": "Identifier scheme",
               "description": "Indicate whether a CVE or CWE was matched or a
heuristic identified the vulnerability",
               "type": "string",
               "enum": ["cve", "cwe", "other"]
           },
           "value": {
               "title": "Vulnerability identifier",
               "description": "An indicator for the specific CVE, CWE, or heuristic
that was triggered by the URI",
               "type": "string"
           }
       }
   }
}
}
```

D. Aggregated Schemata

This appendix lists the complete JSON schemata for the aggregated data categories in ACDC.

D.1. *Anonymized or Pseudonymized Data (eu.acdc.anonymized)*

```
{
    "title": "ACDC dataset for anonymised or pseudonymised data",
    "description": "This is the schema for aggregated data that is intended to be
used by the research workflow and workflows devoted to WP4. It is important to
note, that data format must not contain any data that is directly related to a
person. This specific schema contains reports that may be referenced by an
aggregation/correlation report",
    "properties": {
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "report_category": {
            "title": "Report category",
            "description": "The category of the report: anonymized or pseudonymized
data.",
            "type": "string",
            "enum": ["eu.acdc.anonymized"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the reported observation took place.
This can for example be when an attack occurred, when a malware hosting was
observed, or when a compromise took place according to log files.",
            "type": "string",
            "format": "date-time"
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "src_mode": {
            "title": "Type of the reported object",
            "description": "The type of the reported object: source replaced by
hash, pseudonymised (prefix conserving), or anonymised (last 2 Bytes erased)",
            "type": "string",
            "enum": ["anon", "pseudo"]
        },
        "source_key": {
            "title": "The type of the reported object.",
            "description": "The type of the reported object.",
            "type": "string",
            "enum": ["ip", "malware", "subject", "uri"]
        },
        "source_value": {
            "title": "Identifier of the reported object",
            "description": "The identifier of the reported object like its
anonymised or pseudonymised IP address or URI.",
            "type": "string"
        },
        "src_asn": {
            "title": "ASN of source",
            "description": "ASN of source",
            "type": "string"
        },
        "src_country": {
            "title": "Country of source",
            "description": "Country, the source belongs to (According to GeoIP
Information)",
            "type": "string"
        },
        "src_city": {
            "title": "City of source",
            "description": "City, the source belongs to (According to GeoIP
Information)",
            "type": "string"
        },
        "src_organisation": {
            "title": "Organisation of source",
            "description": "Organisation, the source belongs to",
            "type": "string"
        },
        "ip_protocol_number": {
            "title": "IP protocol number",
            "description": "The RFC 790 decimal internet protocol number of the
connection.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "src_port": {
            "title": "Source port of the connection",
            "description": "The source port of the connection. This is always the
port on the reported system (i.e., the one identified by source_value).",
            "type": "integer"
        },
        "dst_port": {
            "title": "Destination port of the connection",
            "description": "The destination port of the connection. This is always
the remote port from the perspective of the reported system (i.e., the one
identified by src_value). It can for example be the port of a honeypot that was
contacted to infect it.",
            "type": "integer"
        },
        "src_connection_type": {
            "title": "Type of connection",
            "description": "The type of the connection according to Maxmind DB
(e.g. DSL)",
            "type": "string"
        },
        "src_tl_domain": {
            "title": "Toplevel domain",
            "description": "Top-level domain according to reverse DNS data",
            "type": "string"
        },
        "src_sl_domain": {
            "title": "Toplevel domain",
            "description": "Second-level domain according to reverse DNS data",
            "type": "string"
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        }
    },
    "required": ["report_category", "report_type", "timestamp", "source_value",
"source_key", "src_asn", "src_mode", "version"]
}
```

D.2. *Aggregated or Correlated Data (eu.acdc.correlated)*

```
{
    "title": "ACDC dataset for aggregated or correlated data",
    "description": "This is the family of schematas for aggregated data that is
intended to be used by the research workflow and workflows devoted to WP4. This
schema provides a means for correlated or aggregated data. This schema defines a
root object where other related reports are referenced.",
    "properties": {
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "report_category": {
            "title": "Report category",
            "description": "The category of the report: an aggregation or
correlation.",
            "type": "string",
            "enum": ["eu.acdc.correlated"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Starting date of the agregation window",
            "description": "The timestamp details the starting date of the
aggregation windows. All reports whose original timestamp (this can for example be
when an attack occurred, when a malware hosting was observed, or when a compromise
took place according to log files) falls into the period of the measurement window
(timestamp, timestamp + duration) are covered by the report.",
            "type": "string",
            "format": "date-time"
        },
        "duration": {
            "title": "The duration of the aggregation window",
            "description": "The duration of the aggregation window in seconds.",
            "type": "integer",
            "minimum": 0
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "related_reports": {
            "title": "List of related reports",
            "description": "The fiels contains a list of reports referred by
report_id that belong to a common attack (aggregation or correlation)",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "aggregation_criterion": {
            "title": "Aggregation_criterion",
            "description": "The criterion used to aggregate the data, e.g. common
ASN or class-c network.",
            "type": "string"
        },
        "incident_description": {
            "title": "Description of aggregation criterion comprising aggregated
reports.",
            "description": "This field is used to detail the type of incident. It
is intended to complement the report_type for a longer description.",
            "type": "string"
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        }
    },
    "required": ["report_category", "report_type", "timestamp", "duration",
"aggregation_criterion", "related_reports", "incident_description", "version"]
}
```

D.3. Statistical Data (eu.acdc.statistic)

```
{
   "title": "ACDC dataset for aggregated or correlatied data",
   "description": "This is the schema for aggregated data that is intended to be
used by the research workflow and workflows devoted to WP4. It is important to
note, that data format must not contain any data that is related to a person.",
   "properties": {
       "report_id": {
           "title": "Report ID",
           "description": "The ID of the report in the CCH. This will be set by the
CCH and is thus overwritten on import.",
           "type": "string"
       },
       "report_category": {
           "title": "Report category",
           "description": "The category of the report: statistical data.",
           "type": "string",
           "enum": ["eu.acdc.statistic"]
       },
       "report_type": {
           "title": "Report type",
           "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb this should not be longer
than one sentence.",
           "type": "string"
       },
       "timestamp": {
           "title": "Starting date of the measurement window",
           "description": "The timestamp details the starting date of the
measurement windows. All reports whose original timestamp (This can for example be
when an attack occurred, when a malware hosting was observed, or when a compromise
took place according to log files.) falls into the period of the measurement window
(timestamp, timestamp + measurement_window) are covered by the report.",
           "type": "string",
           "format": "date-time"
       },
       "reported_at": {
           "title": "Time of the report's submission",
           "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
           "type": "string",
           "format": "date-time"
       },
       "measurement_window": {
           "title": "Time frame of measurement",
           "description": "Time frame of measurement in seconds",
           "type": "integer",
           "minimum" : 0
       },
       "statistic_field": {
           "title": "Field used for aggregation",
           "description": "The Field that is used to aggregate the data for
statistical means. For example, states that all reports that have the same value in
this field are enumerated in the measurement window.",
           "type": "string"
       },
       "statistic_criterion": {
           "title": "Aggregation criterion",
           "description": "The value (criterion) that has been used to aggregate
the data. E.g. AS680 to enumerate all reports within the measurement window
originating from that ASN.",
           "type": "string"
       },
       "time_series": {
           "title": "Time Series",
           "description": "This field contains an array comprising the values of
the statistic. The length of the array corresponds to the number of lags",
           "type": "array",
           "items": {
               "type": "integer"
           }
       },
       "statistic_description": {
           "title": "Description of statistics",
           "description": "Textual description of statistics applied to the data.
It is intended to complement the report_type for a longer description.",
           "type": "string"
       },
       "version": {
           "title": "Version of the format",
           "description": "The version number of the data format used for the
report.",
           "type": "integer",
           "enum": [2]
       }
   },
   "required": ["report_category", "report_type", "timestamp",
"measurement_window", "statistic_field", "statistic_criterion", "time_series",
"version"]
}
```

D.4. *Results of Metrics (eu.acdc.metric)*

```
{
    "title": "ACDC dataset for aggregated or correlatied data",
    "description": "This is the schema for aggregated data that is intended to be
used by the research workflow and workflows devoted to WP4. It is important to
note, that data format must not contain any data that is directly related to a
person.",
    "properties": {
        "report_id": {
            "title": "Report ID",
            "description": "The ID of the report in the CCH. This will be set by
the CCH and is thus overwritten on import.",
            "type": "string"
        },
        "report_category": {
            "title": "Report category",
            "description": "The category of the report: metric evaluation of
reports.",
            "type": "string",
            "enum": ["eu.acdc.metric"]
        },
        "report_subcategory": {
            "title": "Report subcategory",
            "description": "The subcategory of the report. This is used to
categorise different types of similar reports that have mostly the same fields. It
is defined as an enum in the schema of the report category.",
            "type": "string",
            "enum": ["quality_metric", "ip_based_metric", "event_based_metric",
"other_metric"]
        },
        "report_type": {
            "title": "Report type",
            "description": "The type of the report. This is a free text field
characterising the report that should be used for a human readable description
rather than for automatic processing. As a rule of thumb this should not be longer
than one sentence.",
            "type": "string"
        },
        "timestamp": {
            "title": "Starting date of the measurement window",
            "description": "The timestamp details the starting date of the
measurement windows. All reports whose original timestamp (This can for example be
when an attack occurred, when a malware hosting was observed, or when a compromise
took place according to log files.) falls into the period of the measurement window
(timestamp, timestamp + measurement_window) are covered by the report.",
            "type": "string",
            "format": "date-time"
        },
        "reported_at": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the CCH.
This will be set by the CCH and is thus overwritten on import.",
            "type": "string",
            "format": "date-time"
        },
        "measurement_window": {
            "title": "Time frame of measurement",
            "description": "Time frame of measurement in seconds",
            "type": "integer",
            "minimum" : 0
        },
        "metric_id": {
            "title": "ID of Metric",
            "description": "ID of Metric, all metrics are summarised and specified
in an external document.",
            "type": "integer"
        },
        "metric_result": {
            "title": "Result of Metric",
            "description": "Resulting data (unstructured) of application of
metric",
            "type": "object"
        },
        "metric_description": {
            "title": "Description of the metric.",
            "description": "Detailed description of metrics. This field complements
the report_type if a more specififc or additional decription is intended.",
            "type": "string"
        },
        "version": {
            "title": "Version of the format",
            "description": "The version number of the data format used for the
report.",
            "type": "integer",
            "enum": [2]
        }
    },
    "required": ["report_category", "report_subcategory", "report_type",
"timestamp", "measurement_window", "metric_id", "metric_result",
"metric_description", "version"]
}
```

E. X-ARF Schemata

This appendix lists the schemata for ACDC reports converted to X-ARF for end customer notifications.

E.1. *Attack (eu.acdc.attack_2.0.0)*

```
{
    "title": "Attack",
    "description": "A host performing an attack.",
    "properties": {
        "Report-Type": {
            "title": "Report type: attack",
            "description": "The type of the report: an attack on a system.",
            "type": "string",
            "enum": ["eu.acdc.attack"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the attack observation",
            "description": "The timestamp when the attack took place.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: IP",
            "description": "The type of the reported object: an IP.",
            "type": "string",
            "enum": ["ipv4", "ipv6"]
        },
        "Source": {
            "title": "Attacking IP",
            "description": "The IP of the system performing the attack.",
            "type": "string"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Report-Subcategory": {
            "title": "Attack category",
            "description": "The type of attack performed.",
            "type": "string",
            "enum": ["abuse", "abuse.spam", "compromise", "data", "dos", "dos.dns",
"dos.http", "dos.tcp", "dos.udp", "login", "malware", "scan", "other"]
        },
        "Ip-Protocol-Number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number of
the attack connection.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "Ip-Version": {
            "title": "IP version number",
            "description": "The IP version of the attack connection.",
            "type": "integer",
            "enum": [4, 6]
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Duration": {
            "title": "The duration of the attack",
            "description": "The duration of the attack in seconds.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Botnet": {
            "title": "Botnet responsible for attack",
            "description": "The botnet this attack can be attributed to.",
            "type": "string",
            "optional": true
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "optional": true,
            "requires": "Alternate-Format-Type"
        },
        "Src-Ip-V4": {
            "title": "Source IPv4 of the attack",
            "description": "The source IPv4 of the attack connection. This is
always the IP of the attacking system (i.e., the one identified by Source). If set,
this field equals Source.",
            "type": "string",
            "format": "ipv4",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Ip-V6": {
            "title": "Source IPv6 of the attack",
            "description": "The source IPv6 of the attack connection. This is 
always the IP of the attacking system (i.e., the one identified by Source). If set, 
this field equals Source.",
            "type": "string",
            "format": "ipv6",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for 
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "Dst-Ip-V4": {
            "title": "Destination IPv4 of the attack",
            "description": "The destination IPv4 of the attack connection. This is 
always the IP of the attacked system.",
            "type": "string",
            "format": "ipv4",
            "optional": true,
            "requires": "Dst-Mode"
        },
        "Dst-Ip-V6": {
            "title": "Destination IPv6 of the attack",
            "description": "The destination IPv6 of the attack connection. This is 
always the IP of the attacked system.",
            "type": "string",
            "format": "ipv6",
            "optional": true,
            "requires": "Dst-Mode"
        },
        "Dst-Mode": {
            "title": "Destination IP mode",
            "description": "The mode of the destination IP. This can be plain for 
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "Src-Port": {
            "title": "Source port of the attack",
            "description": "The source port of the attack connection. This is 
always the port on the attacking system (i.e., the one identified by Source).",
            "type": "integer",
            "optional": true
        },
        "Dst-Port": {
            "title": "Destination port of the attack",
            "description": "The destination port of the attack connection. This is 
always the port on the attacked system.",
            "type": "integer",
            "optional": true
        },
        "Application-Protocol": {
            "title": "Application protocol",
            "description": "The application protocol used for the connection.",
            "type": "string",
            "optional": true
        },
        "Sample-Filename": {
            "title": "Filename of the payload",
            "description": "The filename used for the payload that the attack tried 
to install or run on the attacked system. This should only be used if the payload 
is uploaded to the attacked system directly. Otherwise, Malicious-Uri should be
```

```
used to link this report to an eu.acdc.malicious_uri report that in turn contains
the SHA256 hash.",
            "type": "string",
            "optional": true
        },
        "Sample-Sha256": {
            "title": "Hash of the payload",
            "description": "The SHA256 hash of the payload that the attack tried to
install or run on the attacked system.",
            "type": "string",
            "optional": true
        },
        "Malicious-Uri": {
            "title": "URI of the payload",
            "description": "The URI of the payload in the wild that the attack
tried to install or run on the attacked system. This can for example be the
location of a malware offered as a download or a webshell offered as a remote
include during an attack.",
            "type": "string",
            "format": "uri",
            "optional": true
        },
        "Subject-Text": {
            "title": "Subject of spam email",
            "description": "The subject of an email sent in a report of subcategory
abuse.spam. Varying parts, especially personal information like names or email
addresses, are replaced with the placeholder '{}'",
            "type": "string",
            "optional": true
        },
        "Mail-Header": {
            "title": "Email header",
            "description": "The header of the email.",
            "type": "string",
            "optional": true
        },
        "Bit-Rate": {
            "title": "Bits per second of traffic",
            "description": "The number of bits per second of traffic transferred,
for example in a DoS attack.",
            "type": "integer",
            "optional": true
        },
        "Packet-Rate": {
            "title": "Packets per second of traffic",
            "description": "The number of packets per second of traffic
transferred, for example in a DoS attack.",
            "type": "integer",
            "optional": true
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

E.2. Bot (eu.acdc.bot_2.0.0)

```
{
    "title": "A bot",
    "description": "A system infected with a bot.",
    "properties": {
        "Report-Type": {
            "title": "Report type: a bot",
            "description": "The type of the report: a bot infection.",
            "type": "string",
            "enum": ["eu.acdc.bot"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the bot infection was observed.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: IP",
            "description": "The type of the reported object: an IP address.",
            "type": "string",
            "enum": ["ipv4", "ipv6"]
        },
        "Source": {
            "title": "Infected IP",
            "description": "The IP address of the infected system.",
            "type": "string"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Report-Subcategory": {
            "title": "Report subcategory",
            "description": "The type of the bot.",
            "type": "string",
            "enum": ["fast_flux", "other"]
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Duration": {
            "title": "Duration of the bot observation",
            "description": "The duration in seconds during which the bot infection
was observed. This can be the timespan during which connections to the C2 server
were observed.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Botnet": {
            "title": "Botnet infection is attributed to",
            "description": "The botnet the bot is attributed to.",
            "type": "string",
            "optional": true
        },
        "Bot-ID": {
            "title": "Identifier of the bot",
            "description": "The identifier the botnet uses for this bot if
available.",
            "type": "string",
            "optional": true
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "requires": "Alternate-Format-Type",
            "optional": true
        },
        "Ip-Version": {
            "title": "IP version number",
            "description": "The IP version of the infected sytem's IP address.",
            "type": "integer",
            "enum": [4, 6],
            "optional": true
        },
        "Ip-Protocol-Number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number of
the C2 connection.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255,
            "optional": true
        },
        "Src-Ip-V4": {
            "title": "Source IPv4 of the bot",
            "description": "The source IPv4 of the bot infected system. If set,
this field equals Source.",
            "type": "string",
            "format": "ipv4",
            "requires": "Src-Mode",
            "optional": true
        },
        "Src-Ip-V6": {
            "title": "Source IPv6 of the bot",
            "description": "The source IPv6 of the bot infected system. If set,
this field equals Source.",
            "type": "string",
            "format": "ipv6",
            "requires": "Src-Mode",
            "optional": true
        },
        "Src-Mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "Src-Port": {
            "title": "Source port of the C2 connection",
            "description": "The source port of the connection from the bot to the
C2 server. This is always the port on the bot infected system.",
            "type": "integer",
            "optional": true
        },
        "C2-Ip-V4": {
            "title": "IPv4 of the C2",
            "description": "The IPv4 of the C2 server.",
            "type": "string",
            "format": "ipv4",
            "requires": "C2-Mode",
            "optional": true
        },
        "C2-Ip-V6": {
            "title": "IPv6 of the C2",
            "description": "The IPv6 of the C2 server.",
            "type": "string",
            "format": "ipv6",
            "requires": "C2-Mode",
            "optional": true
        },
        "C2-Mode": {
            "title": "C2 IP mode",
            "description": "The mode of the C2 server IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "C2-Port": {
            "title": "C2 port of the C2 connection",
            "description": "The port of the C2 connection on the C2 server.",
            "type": "integer",
            "optional": true
        },
        "Sample-Sha256": {
            "title": "Hash of the malware",
            "description": "The SHA256 hash of the malware the system is infected
with.",
            "type": "string",
            "optional": true
        },
        "Fast-Flux-Uri": {
            "title": "URI of the fast flux domain",
            "description": "The URI of the fast flux domain resolving to this
bot.",
            "type": "string",
            "format": "uri",
            "optional": true
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

E.3. C2 server (eu.acdc.c2_server_2.0.0)

```
{
    "title": "A C2 server",
    "description": "A command and control server.",
    "properties": {
        "Report-Type": {
            "title": "Report type: C2",
            "description": "The type of the report: a C2 server.",
            "type": "string",
            "enum": ["eu.acdc.c2_server"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the C2 server was observed.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: IP",
            "description": "The type of the reported object: an IP address.",
            "type": "string",
            "enum": ["ipv4", "ipv6"]
        },
        "Source": {
            "title": "C2 IP",
            "description": "The IP address of the C2 server.",
            "type": "string"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Report-Subcategory": {
            "title": "C2 subcategory",
            "description": "The control channel used by the C2.",
            "type": "string",
            "enum": ["http", "irc", "other"]
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Duration": {
            "title": "Duration of the C2 observation",
            "description": "The duration in seconds during which the C2 server was
observed. This can be the timespan during which connections to the C2 server were
successful.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Botnet": {
            "title": "Botnet C2 is attributed to",
            "description": "The botnet the C2 server is attributed to.",
            "type": "string",
            "optional": true
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "requires": "Alternate-Format-Type",
            "optional": true
        },
        "Ip-Version": {
            "title": "IP version number",
            "description": "The IP version of the C2 server's IP address.",
            "type": "integer",
            "enum": [4, 6],
            "optional": true
        },
        "Ip-Protocol-Number": {
            "title": "IP protocol number",
            "description": "The IANA assigned decimal internet protocol number used
for C2 connections.",
            "type": "integer",
            "minimum": 0,
            "maximum": 255,
            "optional": true
        },
        "C2-Ip-V4": {
            "title": "IPv4 of the C2",
            "description": "The IPv4 of the C2 server. If set, this field equals
Source.",
            "type": "string",
            "format": "ipv4",
            "requires": "C2-Mode",
            "optional": true
        },
        "C2-Ip-V6": {
            "title": "IPv6 of the C2",
            "description": "The IPv6 of the C2 server. If set, this field equals
Source.",
            "type": "string",
            "format": "ipv6",
            "requires": "C2-Mode",
            "optional": true
        },
        "C2-Mode": {
            "title": "C2 IP mode",
            "description": "The mode of the C2 server IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "C2-Port": {
            "title": "C2 Port of C2 connections",
            "description": "The port of C2 connections on the C2 server.",
            "type": "integer",
            "optional": true
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

E.4. Fast Flux Domain (eu.acdc.fast_flux_2.0.0)

```
{
    "title": "Fast Flux Domain",
    "description": "A fast flux domain.",
    "properties": {
        "Report-Type": {
            "title": "Report type: Fast flux",
            "description": "The type of the report: a fast flux domain.",
            "type": "string",
            "enum": ["eu.acdc.fast_flux"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the fast flux domain was first
observed. If the report contains IPs, this is typically the earliest timestamp of
the IPs that the domain resolves to.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: domain",
            "description": "The type of the reported object: a domain URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "Source": {
            "title": "Fast flux domain",
            "description": "The fast flux domain URI.",
            "type": "string",
            "format": "uri"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Duration": {
            "title": "Duration of the observation",
            "description": "The duration of the observation in seconds. If the
report contains IPs, this is typically the difference between the earliest and the
latest timestamp of the IPs that the domain resolves to.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Botnet": {
            "title": "Botnet of the fast flux domain",
            "description": "The botnet this fast flux domain can be attributed
to.",
            "type": "string",
            "optional": true
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "optional": true,
            "requires": "Alternate-Format-Type"
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

E.5. *Malicious URI (eu.acdc.malicious_uri_2.0.0)*

```
{
    "title": "Malicious URI",
    "description": "A URI pointing to malicious content.",
    "properties": {
        "Report-Type": {
            "title": "Report type: malicious URI",
            "description": "The type of the report: a malicious URI.",
            "type": "string",
            "enum": ["eu.acdc.malicious_uri"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the reported observation",
            "description": "The timestamp when the malicious URI was observed.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: URI",
            "description": "The type of the reported object: a URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "Source": {
            "title": "Malicious URI",
            "description": "The URI to the malicious content.",
            "type": "string",
            "format": "uri"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Report-Subcategory": {
            "title": "Malicious category",
            "description": "The type of the malicious content at the URI.",
            "type": "string",
            "enum": ["exploit", "malware", "phishing", "other"]
        },
        "Duration": {
            "title": "Duration of the reported observation",
            "description": "The duration in seconds during which the malicious URI
was observed.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Botnet": {
            "title": "Botnet of the malicious URI",
            "description": "The botnet this malicious URI can be attributed to.",
            "type": "string",
            "optional": true
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "optional": true,
            "requires": "Alternate-Format-Type"
        },
        "Ip-Version": {
            "title": "IP version number",
            "description": "The IP version of the IP address belonging to the
malicious URI.",
            "type": "integer",
            "enum": [4, 6],
            "optional": true
        },
        "Src-Ip-V4": {
            "title": "Source IPv4 of the URI",
            "description": "The source IPv4 associated with the malicious URI.",
            "type": "string",
            "format": "ipv4",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Ip-V6": {
            "title": "Source IPv6 of the URI",
            "description": "The source IPv6 associated with the malicious URI.",
            "type": "string",
            "format": "ipv6",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "Sample-Filename": {
            "title": "Malicious content file name",
            "description": "The file name of the malicious content if applicable.",
            "type": "string",
            "optional": true
        },
        "Sample-Sha256": {
            "title": "Hash of the malicious content",
            "description": "The SHA256 hash of the malicious content if
applicable.",
            "type": "string",
            "optional": true
        },
        "Exploits": {
            "title": "Exploits in the URI",
            "description": "Exploits discovered in the analysed URI. This is an
array of objects, each giving an identifier scheme like CVE and an identifier for
the actual exploit found.",
            "type": "array",
            "items": {
                "title": "Exploit",
                "description": "Indicator for the type and identifier of the given
exploit",
                "type": "object",
                "properties": {
                    "Type": {
                        "title": "Identifier scheme",
                        "description": "Indicate whether a CVE was matched or a
heuristic identified the exploit",
                        "type": "string",
                        "enum": ["cve", "other"]
                    },
                    "Value": {
                        "title": "Exploit identifier",
                        "description": "An indicator for the specific CVE or
heuristic that was triggered by the file",
                        "type": "string"
                    }
                }
            },
            "optional": true
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

E.6. *Vulnerable URI (eu.acdc.vulnerable_uri_2.0.0)*

```
{
    "title": "Vulnerable URI",
    "description": "A URI pointing to a vulnerable resource.",
    "properties": {
        "Report-Type": {
            "title": "Report type: vulnerable URI",
            "description": "The type of the report: a vulnerable URI",
            "type": "string",
            "enum": ["eu.acdc.vulnerable_uri"]
        },
        "Report-Description": {
            "title": "Report description",
            "description": "The description of the report. This is a free text
field characterising the report that should be used for a human readable
description rather than for automatic processing. As a rule of thumb, this should
not be longer than one sentence.",
            "type": "string"
        },
        "Date": {
            "title": "Time of the reported observation",
            "description": "The timestamp when vulnerable URI was observed.",
            "type": "string",
            "format": "date-time"
        },
        "Source-Type": {
            "title": "Type of the reported object: URI",
            "description": "The type of the reported object: a URI.",
            "type": "string",
            "enum": ["uri"]
        },
        "Source": {
            "title": "Vulnerable URI",
            "description": "The URI to the vulnerable resource.",
            "type": "string",
            "format": "uri"
        },
        "Confidence-Level": {
            "title": "Confidence level of the report",
            "description": "The level of confidence put into the accuracy of the
report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being
verified to be accurate.",
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0
        },
        "Src-Ip-V4": {
            "title": "Source IPv4 of the URI",
            "description": "The source IPv4 associated with the malicious URI.",
            "type": "string",
            "format": "ipv4",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Ip-V6": {
            "title": "Source IPv6 of the URI",
            "description": "The source IPv6 associated with the malicious URI.",
            "type": "string",
            "format": "ipv6",
            "optional": true,
            "requires": "Src-Mode"
        },
        "Src-Mode": {
            "title": "Source IP mode",
            "description": "The mode of the source IP. This can be plain for
unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.",
            "type": "string",
            "enum": ["plain", "anon", "pseudo"],
            "optional": true
        },
        "Vulnerabilities": {
            "title": "Vulnerabilities discovered at the URI",
            "description": "An array of objects describing vulnerabilities
discovered at the vulnerable URI.",
            "type": "array",
            "items": {
                "title": "Vulnerability",
                "description": "Indicator for the type and identifier of the given
vulnerability",
                "type": "object",
                "properties": {
                    "Type": {
                        "title": "Identifier scheme",
                        "description": "Indicate whether a CVE or CWE was matched
or a heuristic identified the vulnerability",
                        "type": "string",
                        "enum": ["cve", "cwe", "other"]
                    },
                    "Value": {
                        "title": "Vulnerability identifier",
                        "description": "An indicator for the specific CVE, CWE, or
heuristic that was triggered by the URI",
                        "type": "string"
                    }
                }
            }
        },
        "Report-ID": {
            "title": "Report ID",
            "description": "The ID of the report. This is the report ID in the CCH
with the suffix @acdc-project.eu.",
            "type": "string"
        },
        "Duration": {
            "title": "Duration of the reported observation",
            "description": "The duration in seconds during which the
vulnerabilities at the URI were observed.",
            "type": "integer",
            "minimum": 0,
            "optional": true
        },
        "Reported-At": {
            "title": "Time of the report's submission",
            "description": "The timestamp when the report was submitted to the
CCH.",
            "type": "string",
            "format": "date-time"
        },
        "Additional-Data": {
            "title": "Additional data",
            "description": "Additional data for the observation. This allows
putting more specific information into a report on a case by case basis in a
structured manner. The usage of this field is at the data providers discretion.",
            "type": "object",
            "optional": true
        },
        "Alternate-Format-Type": {
            "title": "Type of the alternate format",
            "description": "The type of the alternate format description of the
observation.",
            "type": "string",
            "enum": ["CybOX", "hpfeeds", "IDMEF", "IODEF", "IPFIX", "NetFlow v9",
"OpenIOC", "sFlow", "STIX"],
            "optional": true
        },
        "Alternate-Format": {
            "title": "Alternate format description of the observation",
            "description": "A description of the observation in an alternate
format.",
            "type": "string",
            "optional": true,
            "requires": "Alternate-Format-Type"
        },
        "Ip-Version": {
            "title": "IP version number",
            "description": "The IP version of the IP address belonging to the
vulnerable URI.",
            "type": "integer",
            "enum": [4, 6],
            "optional": true
        },
        "Reported-From": {
            "title": "Sending email address",
            "type": "string"
        },
        "Category": {
            "title": "The X-ARF category",
            "type": "string",
            "enum": ["abuse"]
        },
        "User-Agent": {
            "title": "Name and version of the generating software",
            "type": "string"
        },
        "Attachment": {
            "title": "Attachment present",
            "type": "string"
        },
        "Schema-URL": {
            "title": "URI to the JSON-schema",
            "type": "string",
            "format": "uri"
        },
        "Version": {
            "title": "Version of the X-ARF specification: 0.2",
            "type": "number",
            "enum": [0.2],
            "optional": true
        },
        "Occurences": {
            "title": "Number of attacks",
            "type": "integer",
            "optional": true
        },
        "TLP": {
            "title": "Sensitivity of the report in TLP",
            "type": "string",
            "optional": true
        }
    },
    "additionalProperties": false
}
```

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.