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

Deliverable	**D1.7.1 – Data Format Specification**
Work package	WP 1, Requirements and Specifications
Due date	01/08/2013
Submission date	30/07/2013
Revision	2.5
Status of revision	
Responsible partner	DFN-CERT Services GmbH
Contributors	KU Leuven TU Delft Various project partners for the questionnaires
Project Number	CIP-ICT PSP-2012-6 / 325188
Project Acronym	ACDC
Project Title	Advanced Cyber Defence Centre
Start Date of Project	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
1.0	30/05/2013	DFN-CERT	TOC
2.0	12/07/2013	DFN-CERT	Initial version of content
2.1	16/07/2013	DFN-CERT	Internal comments addressed
2.2	16/07/2013	DFN-CERT	Internal comments addressed
2.3	16/07/2013	TU-DELFT	Contents for section 3.4 added
2.4	18/07/2013	DFN-CERT	Added Introduction and Summary
2.5	29/07/2013	DFN-CERT	Layout improvements

Glossary

ACDC	Advanced Cyber Defence Centre
CCH	Centralised Data Clearing House

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 will introduce components and workflows to gather and analyse data originating from technical sensors such as honeypots and IDSs as well as user reports. A central role in ACDC is devoted to the centralised data clearing house (CCH) providing a platform for storage and analysis of gathered 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. Since each user group and technical sensor does have different requirements the choice of applicable data transport formats is a crucial task. In this document the relevant data formats are assessed, and a first advice is given on which formats are expected to be important for the later usage in the context of the ACDC project.

The first necessary step is to assess the available data formats and their properties. This is done by analysing a survey in collaboration with all ACDC partners. The survey is based on a questionnaire that has been distributed to the partners comprising questions about the data formats in use, their use cases, properties, and planned extensions. This survey reveals a list of 15 different formats from which nearly all are based on a textual representation of the data. Because of their formal structure 10 formats allow an automatic processing of the data whereas 7 formats provide a publicly available specification of its syntax and semantics. The responses include some well-known formats such as IODEF, X-ARF, and STIX. Other applied formats are devoted to a specific use case like the sFlow format for transferring NetFlow data and hpfeeds, which is specific for honeypot data.

In a second step we assess a list of technical, organisational, and legal requirements that have been derived from publicly available specifications of data formats as well as a preliminary analysis of the projects' planned evaluation tasks and the legal framework. In addition, a list of use cases and their specific requirements has been compiled that are relevant for the project. It turns out that the available data formats meet these requirements pretty well. However, all data formats lack a fine granular specification for restrictions concerning the usage of included data. This shortcoming could be either compensated by an extension of the data formats itself or by considering this feature in the transport protocol.

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 an extensive sharing of information across borders to improve the early detection of botnets
- provide an extensive set of solutions accessible online for mitigating ongoing attacks
- use the pool of knowledge to create best practices that support organisations in raising their cyber-protection level
- create a European wide network of cyber-defence centres

ACDC will deploy a comprehensive set of national support centres throughout eight Member States interconnected to ACDC's central clearing house (CCH). Through this networked approach, ACDC will also pave the way for a consolidated approach to protect organisations from cyber-threats and support mitigation of on-going attacks through easy access to an increasing pool of solutions.

A central role in ACDC is devoted to the centralised data clearing house. This component 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.

As shown in Fig. 1 the CCH collects data from various sources that comprise technical sensors as well as user reports. 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 anonymisation for any other usage of the data. It is reasonable to assume, that all these requirements cannot be fulfilled by a single data exchange format. Instead, a bunch of formats is needed 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. An additional aim of this document is to identify shortcomings resulting from the project demands not addressed by the data formats already in use and to give advices how to close these gaps.

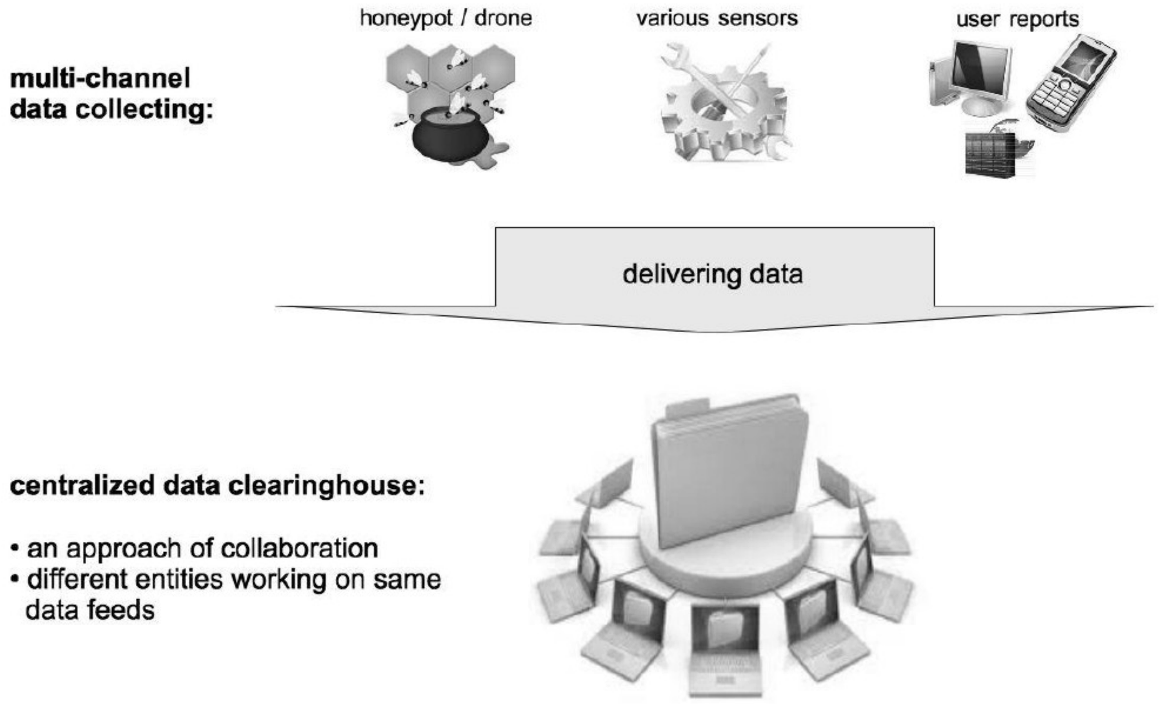

honeypot / drone
various sensors
user reports
multi-channel data collecting:
delivering data
centralized data clearinghouse:
• an approach of collaboration
• different entities working on same data feeds

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

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. In addition, we identify missing demands not addressed by data formats already in use and propose how to fill those gaps. The results of the survey are shown in the Appendix.

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 are listed as presented. There is no particular order of relevance, neither of nor within the categories themselves. Some requirements can also be implemented by the communication protocol used for the data exchange and therefore discarded for the data exchange format. Since there is no specific communication protocol or requirements for the communications protocol to be used these requirements apply to the data exchange in general and might therefore be handled by the data exchange format.

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 tool itself as stated in Section 1.1.6.1 in ACDC's Description of Work on Page 31.

Whereas in other cases users want their personal data be handled for involvement or notification purposes. In this case the appropriation of the provided personal data must be observed. Most likely a user wants an immediate classification or result of the malware's analysis she reported. If this is not feasible she 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 pattern to the CCH. On the other hand the ISP should contact their customer to remove malware, maybe using the ACDC's project web page.

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. On the other hand, this update process 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 c&c 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 c&c 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 —mostly automated—analyse the provided malware or reports and classify them. But they will most likely also perform manual analysis of malware or reports that cannot be classified by software. On the one hand these analysts take data samples from the CCH but 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 academic and business researchers as well as ISEC specialists and employees of Computer Security Incident Response Teams (CSIRT).

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 as well. They are interested in malware samples as well as the resulting analysis to close vulnerabilities in their products. And also vendors of Operating Systems (OS) are interested in exploitation data too.

On the other hand these vendors might 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. The question whether there are legal requirements or even an obligation to share data with LEAs is covered in more detail in Deliverable D1.8.2 of the ACDC project.

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 to clean and stop it from spreading malware any further when their Content Management System (CMS) is infected. 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. On the other hand data centres might 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 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.

So we assume this category might only be interested in receiving data. On the other hand small and medium enterprises (SMEs) are more dependent on someone else's IT infrastructure or security related services.

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 when other means are not appropriate or the number of people affected by a loss of data is so great that it'd not be efficient or possible to notify each person on its own.

3.2. Technical Requirements

Technical requirements 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 to exchange 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 to extend and adopt the data format for the project's requirements and allows to implement 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 of time. Those might include anonymization or deletion of provided data. There might also be a legal requirement to anonymize or delete data after a period of time 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

Additionally 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 level either by encryption of the communication protocol and 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 initially 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 format 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 [D1.2.1], 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 to remove both sender and destination e-mail addresses from the contents, and/or maybe entirely the content 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 5101]. 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 in an interdisciplinary fashion, 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 mitigation/detection tools 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 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*

Obliges data controller 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 her personal data (notice), having means to obtain further information (right of access) and immediate tools of recourse towards the controller in case she feels her 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 mitigation/detection tools.

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. Whereas not all criteria have to be met. For some there are workarounds by implementing the requirement at 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 the 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 5101] 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, a large number of 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 to track connections that are originated by a compromised system. This is especially important to track botnets because this allows to monitor 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 to track 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. For example, the line

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

of an example IODEF message is comprised of 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 alert 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 cyber threat 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 pattern that allow to characterise it. These pattern are, for example artefacts and/or behaviours of interest within a cyber security 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 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 servers 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 e-mail, 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. A JSON schema exists defining the structure and syntax of the data. This allows to test 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 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 C&C 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 format does 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.

A survey was conducted to collect the currently used data formats and to gather information about the usage of these formats. This survey includes a questionnaire that is partitioned into three blocks (for the complete questionnaire we refer to the Appendix). After the name of 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

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

- **E,F,G:** Proprietary data formats based on JSON

- **H:** Proprietary data formats based on XML

- **I:** Sflow 5.0

- **J:** hpfeeds

22

- **K, L:** IODEF
- **M:** X-ARF
- **N:** IDMEF
- **O:** STIX/TAXII
- **P, Q:** proprietary formats

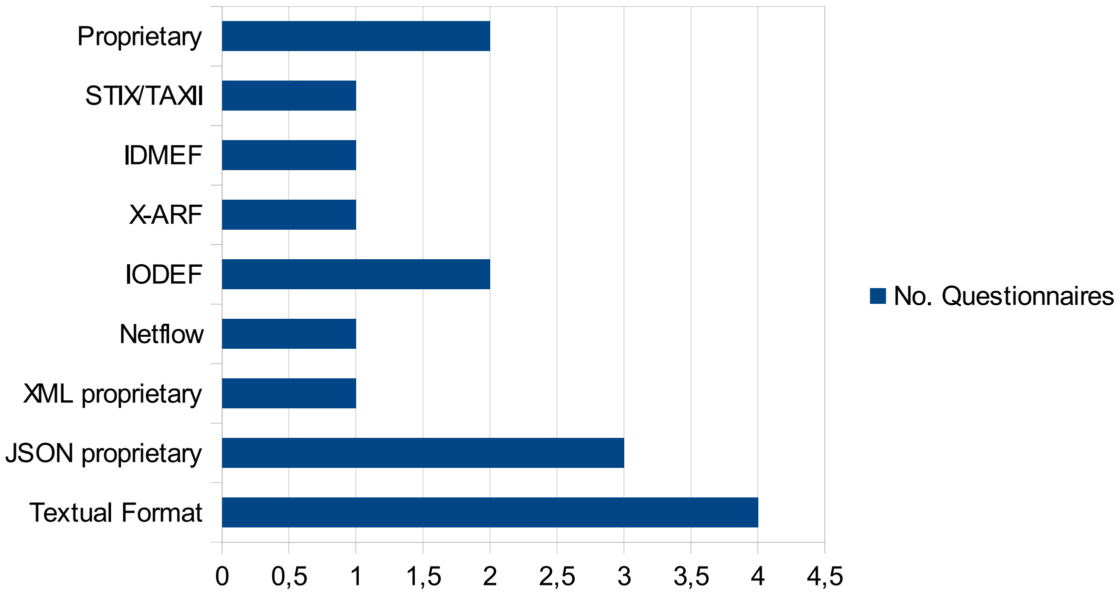


Fig. 2: Overview of the received questionnaires

5.1.2. Overview of the technical properties of the 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: **E,F,G,H,I,J,K,L,M,N**

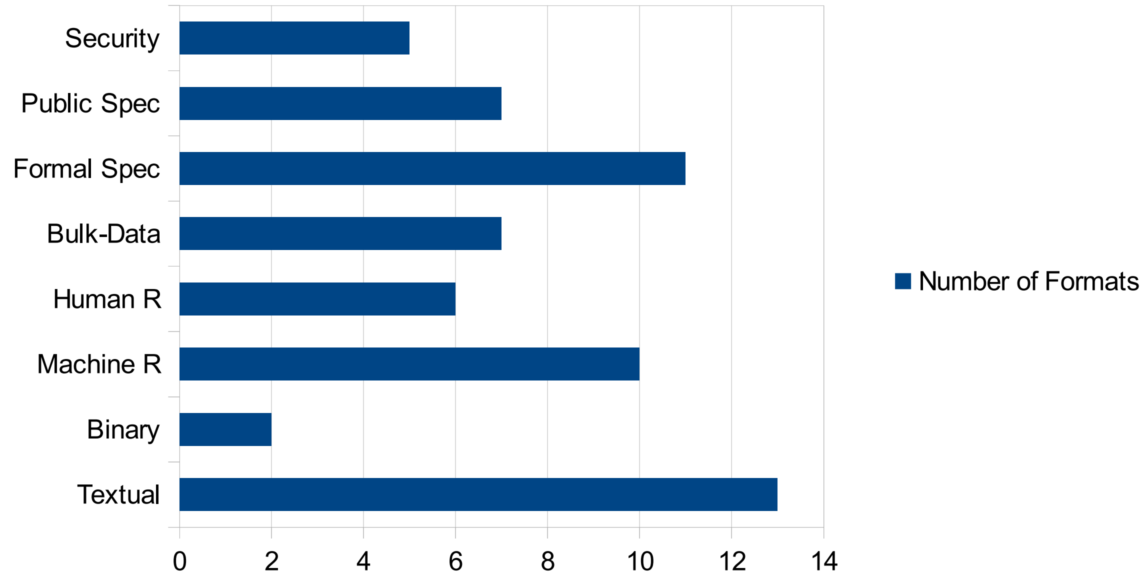


Fig. 3: Properties of data formats

5.1.3. User groups and their requirements

- 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

24

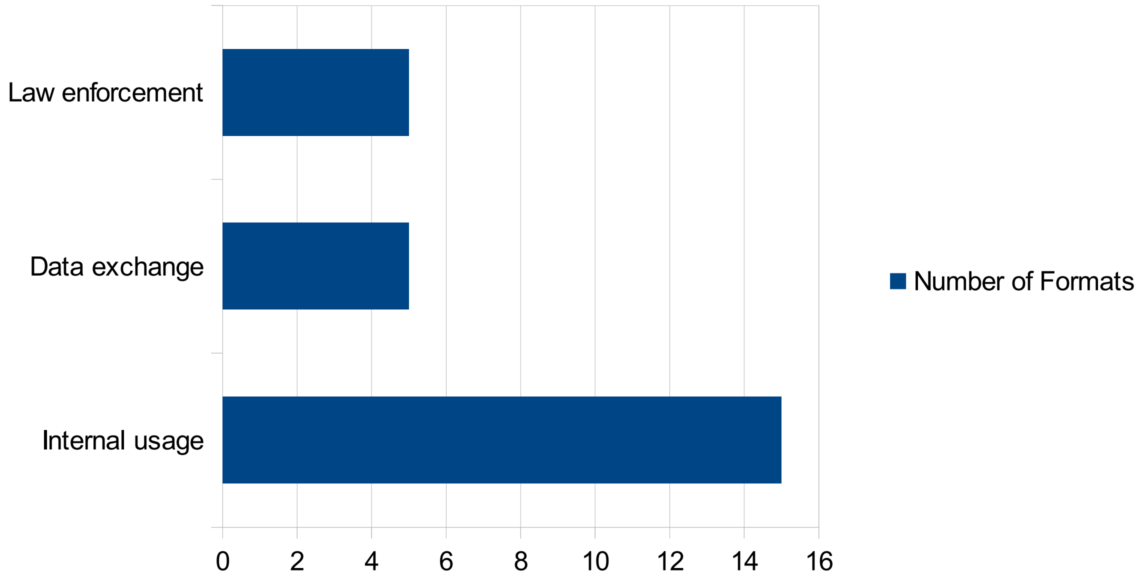


Fig. 4: Number of formats supporting a specific user group

5.1.4. Overview of data formats and supported sources

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

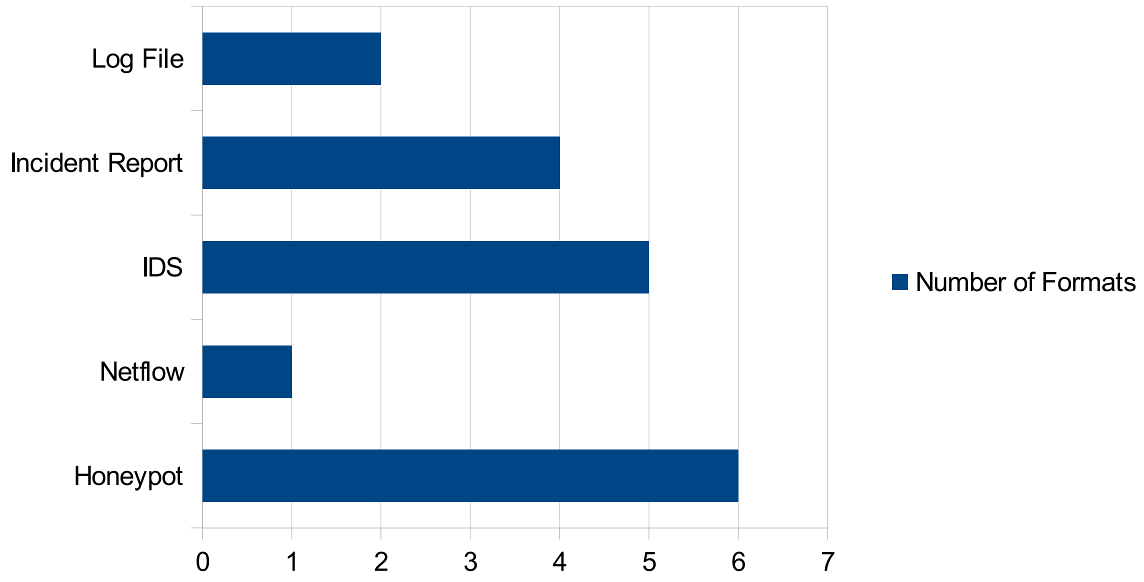


Fig. 5: Number of data formats supporting a specific user group

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 the 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 export from a sensor (e.g. honeypot)
- 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 or filter the data

  – 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 or filter the data

  – Support of the relevant information

- Reporting to end users: **M**

  – Human readable

  – Public specification/description available

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 sensor 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.

- 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. Conclusion

This document aims at proposing data formats that should be adopted by the ACDC project. To assess relevant data formats a survey has been conducted using a questionnaire distributed to all ACDC partners. The resulting list 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.

In the first part of this document a list 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 can be expected to be relevant for the project. For example, such a use case is the submission of IDS sensor data to a central repository. Furthermore, the basic requirements that are crucial for these use cases have been stated. From this, a preliminary proposal has been given which data formats are expected to be relevant for the individual use case. It points 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 with IRTs. 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. Therefore, this format may be advantageous to store structured threat information.

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. Fortunately, IODEF as well as X-ARF and STIX satisfy this requirement. A shortcoming of all data formats is the lack of access control. Although IODEF and TAXII include some form of access control the granularity is very limited and does not, for example, support different privileges for multiple user groups. However, it is important to note that access control is often implemented by the transport protocol.

7. References

D1.8.1: KU Leuven, ACDC Deliverable D1.8.1: Legal Requirements, to be published
D6.1.1: Engineering Ingegneria Informatica, ACDC Deliverable D6.1.1: User profiles and categorization, 2013
D1.2.1: ECO, ACDC Deliverable D1.2.1: Specification of Tool Group "Centralised Data Clearing House", to be published
RFC 5101: B. Claise, Ed., Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, 2008
DoW: ECO, ACDC Description of Work, 2013
RFC 4765: H. Debar, D. Curry, B. Feinstein, The Intrusion Detection Message Exchange Format (IDMEF), 2007

8. Appendix

For reference we first list the complete questionnaire that was sent to the project consortium and afterwards the filled questionnaires with abbreviated questions.

8.1. *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?

8.2. *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.

8.3. *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.

37

8.4. *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>
```

8.5. *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.

8.6. 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.

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

```
{
  "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&d
ownloadId=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",
  "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 (

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.

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

[2] Same output as "file" or "magic" command

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

8.7. *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":
"4679cd287cd2ebd62d56e510ac20d88ae04b02966dfad21fcb0090b8421ddc075fac3ca769296
b70bd1fd4f5569a61be5532116ded0fc196508d4e305a58f258",
                                "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"
}
```

8.8. *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",
            }]
    }
 ]
}
```

8.9. *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:

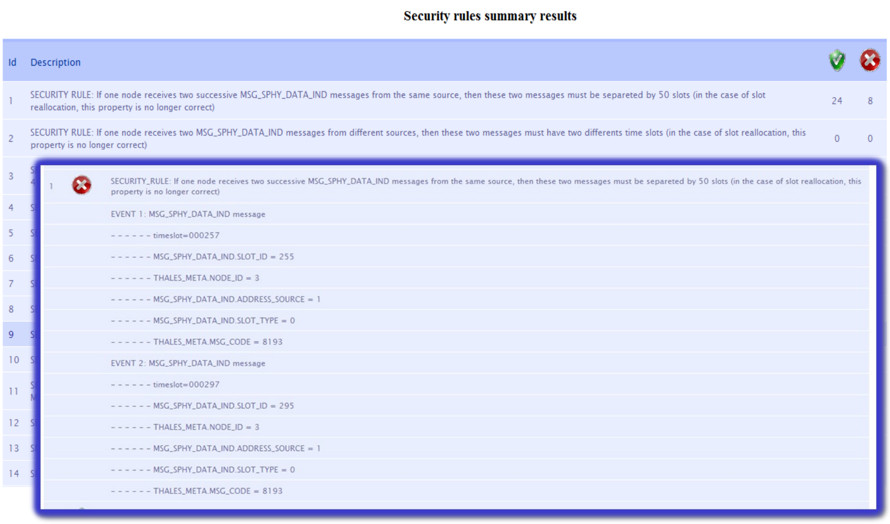
- TDMA, Time Division Multiple Access, protocol traces from OSI layer 1+2 generated by a Omnet++ simulator in ASCII+HEXA format, as for instance:

```
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			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

```
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.

8.10. 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...

8.11. 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.

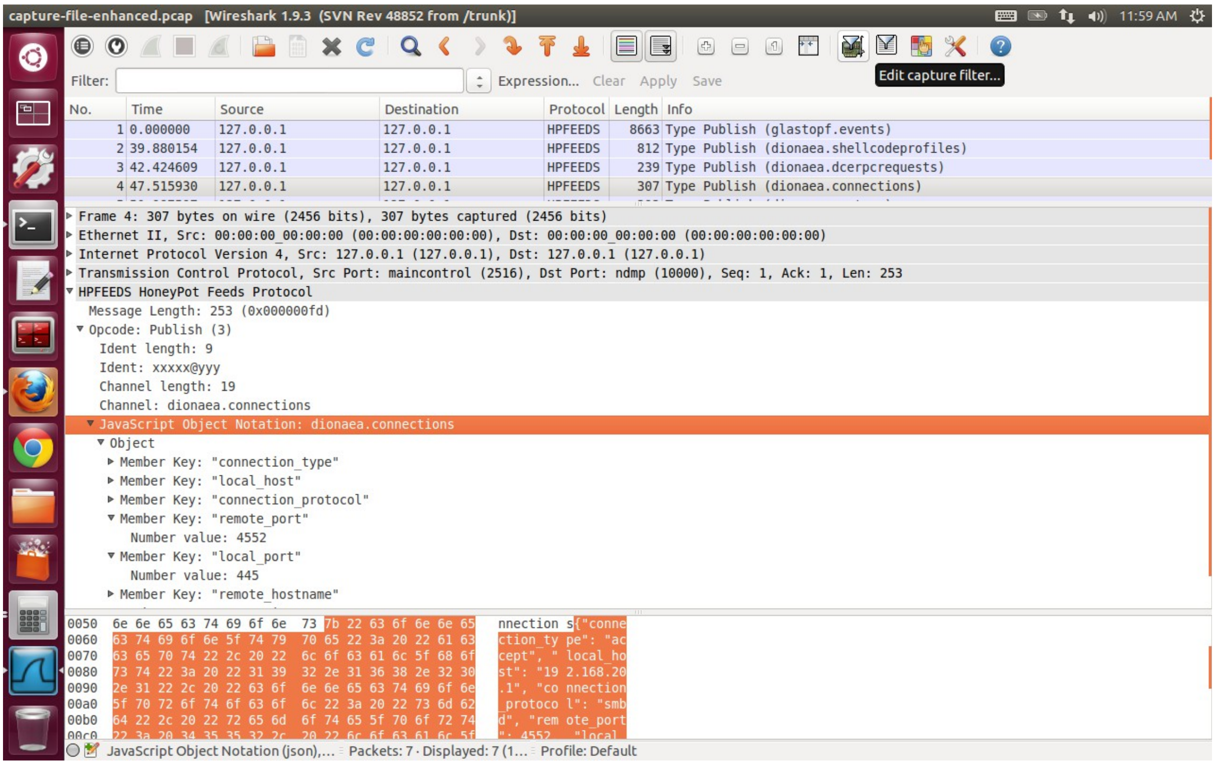
(d) Experiences

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
00c0  22 3a 20 34 35 35 32 2c  20 22 6c 6f 63 61 6c 5f   ": 4552,  "local_
```

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.

72

8.12. 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.

8.13. 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.

8.14. 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--
```

8.15. 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.

8.16. 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.co
m/foo/malicious2.html,http://example.com/foo/malicious3.html</URIObject:Value>
                    </cybox:Properties>
                </cybox:Object>
            </indicator:Observable>
        </stix:Indicator>
    </stix:Indicators>
</stix:STIX_Package>
```

8.17. 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

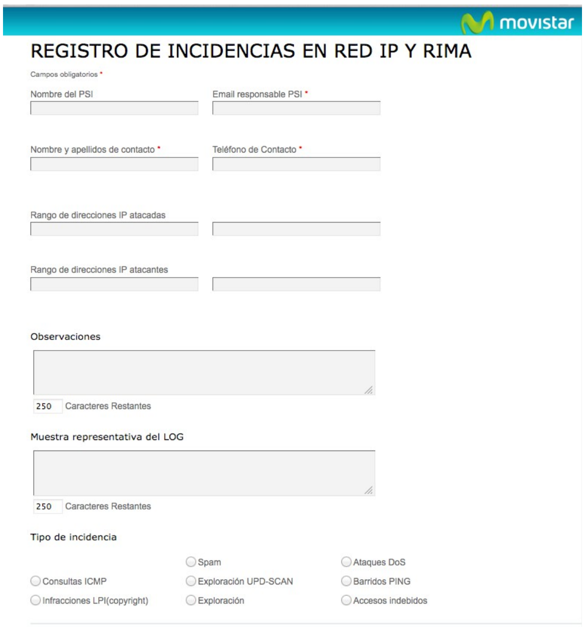
  - IP origin of attack

  - 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.

8.18. 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

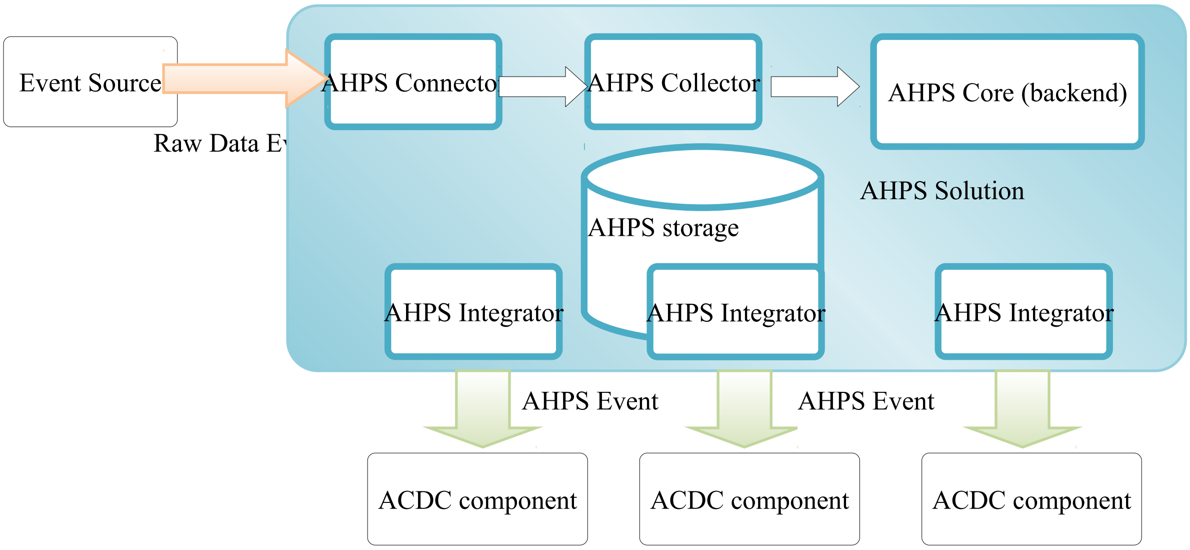
(a) Role and rationale

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



[1] Distributed Audit Service (XDAS) https://www2.opengroup.org/ogsys/catalog/p441

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.

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.

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.