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

Deliverable	**D1.2.2 Specification of Tool Group "Centralised Data Clearing House"**
Work package	WP1 Requirements & Specifications
Due date	M30
Submission date	31/07/2015
Revision	Final
Status of revision	
Responsible partner	ECO
Contributors	Michael Weirich (eco) Peter Meyer (eco) Thorsten Kraft (eco) Thomas King (DE-CIX) Sascha Bleidner (DE-CIX) Matthias Simonis (eco) Georg Roßrucker (eco) Thomas Berchem (eco) Alexander Zeh (eco)
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
V0.1	03/11/2015	Michael Weirich, Matthias Simonis, Alexander Zeh, Georg Roßrucker	CCH Schemata, examples, Infrastructure, Datasets, XMPP Server
V0.2	04/23/2015	Michael Weirich	DE-CIX infrastructure, Examples, Figures
V0.3	06/08/2015	Michael Weirich	CCH Statistics
V0.4	06/09/2015	Michael Weirich	Workflow - processes
V0.5	06/20/2015 – 07/20/2015	Review –DE-CIX	Sascha Bleidner, Thomas King
V0.6	07/21/2015	Michael Weirich	Changes, resulting from review
V0.7	07/22/2015	Michael Weirich, Thomas Berchem	Formatting
V0.8	07/24/2015	Michael Weirich	X-Arf
V0.9	07/25/2015	Michael Weirich	Metrics, Pentest results
V1.0	07/26/2015	Peter Meyer / Thorsten Kraft	Final Review

Glossary

ACDC	Advanced Cyber Defence Centre
CCH	Centralized (Data) Clearing House
SME	Small-and Medium Enterprises
DoW	Description of Work, Part B Form of the Amendment
NSC	National Support Centre
ISP	Internet Service Provider
DDoS	Distributed Denial of Service
DoS	Denial of Service
IdMS	Identity Management System
TI	Trusted Introducer
STIX	Structured Threat Information Expression
API	Application-Programming-Interface
SotA	State of the Art
HTTP	Hypertext Transmission Protocol
TTL	Time-to-Live
WP	Work Package
DAM	Data Access Management
EU	European Union
CERT	Computer Emergency Response Team

Index

List of Figures

List of Tables

1. Introduction

This deliverable **D.1.2.2 "Specification of the Tool Group Centralised Data Clearing House"** is the second iteration of the specifications for the databases and data storage components within the ACDC project. This document provides the basic specifications and requirements for this key component of the project and supplies the information, how to integrate and interact with other project software components, as described in the **D1.1.2 "Overall Software Architecture".**
This document focuses on initial concept and design, focussing on the data itself, its storage, data formats, access control, user management and security framework. It describes the actual implementation and adaptions that have integrated during this project pilot. It replaces the document **D.1.2.1 "Specification of the Tool Group Centralised Data Clearing House"** as the only recent version.

The Advanced Cyber Defence Centre (ACDC) is a European project aiming to improve prevention, detection and mitigation of botnets. Its goal is to deploy an infrastructure of interconnected support centres across European Member States linked to a central ACDC clearing house.

The Central Data Clearing House with the internal abbreviation **CCH** is the central data storage repository of the ACDC project. The overall concept of the ACDC-project, the ACDC-approach, requires the CCH to limit the boundaries that might prevent other partners or stakeholders from providing and retrieving data. Therefore, the CCH has been designed to provide open standards and a wide flexibility in supporting data formats. This approach is necessary, as the ACDC project is designed as an integration pilot, collecting and processing data from already deployed tools, sensors, sources and further unspecified components.

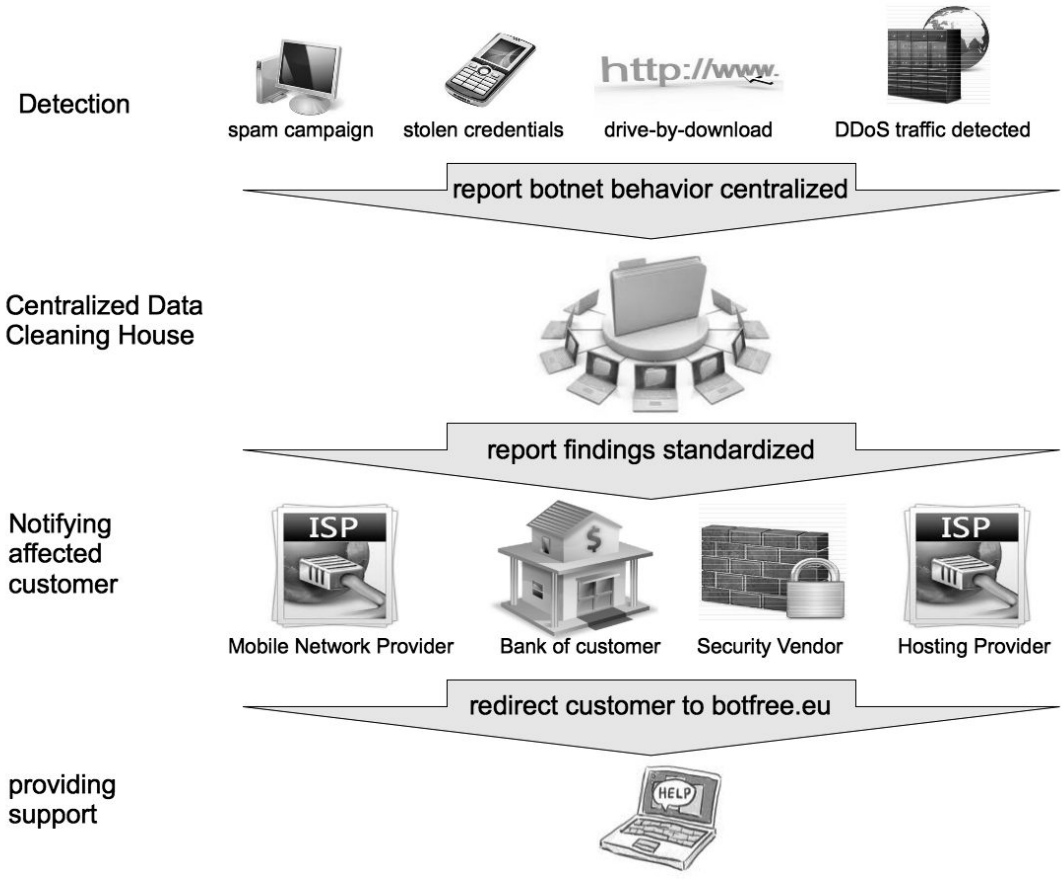

Figure 1: ACDC big picture.

Even though the initial concept defined having no limitations on data (format) submissions, it has been agreed on across the project participants, that a basic standardisation of the submitted data fields and basic requirement on mandatory fields simplifies the data submission and retrieval. These specifications have been defined as the ACDC - "Schemata". These have been outlined and defined in the Deliverables **D1.7.1/2 "Data Formats Specification"**.
Besides the data format specifications, similar agreements had to be made for all other components of the overall software architecture. These have been identified within ***Work Package 1 "Requirements & Specifications"*** (WP1) of the ACDC project.
The following table displays other deliverables that specify the requirements of tools of the overall software architecture:

Deliverable	**Name**
D1.1.1 / D1.1.2	Overall Software Architecture Description
D1.2.1 / D1.2.2	Specification of Tool Group "Centralised Data Clearing House"
D1.3.1 / D1.3.2	Specification of Tool Group "Support Centre"
D1.4.1 / D1.4.2	Specification of Tool Group "Malicious or Vulnerable Websites"
D1.5.1 / D1.5.2	Specification of Tool Group "Network Traffic Sensors"
D1.6.1 / D1.6.2	Specification of Tool Group "End Customer Tools"
D1.7.1 / D1.7.2	Data Formats Specification
D1.8.1 / D1.8.2	Legal requirements

Figure 2: List of Deliverables WP1

Besides the collection and processing of data, the CCH is also designed to distribute data to stakeholders like ISP´s, government agencies, law enforcement, research groups or industry partners. Due to security concerns, but also with the project aiming to support an open community of stakeholders, the access management has been integrated into the Community Website / Community Portal of the project. The end user policy enforcement is part of the CCH.
One approach of ACDC is mutual data sharing across international boarders and an advanced capability of interaction, permissions and restrictions between the involved stakeholders. The Community Portal manages the stakeholder / User database and maintains their relationships. This portal and further settings are described in the deliverable **D6.2.1 ACDC Social Platform.**

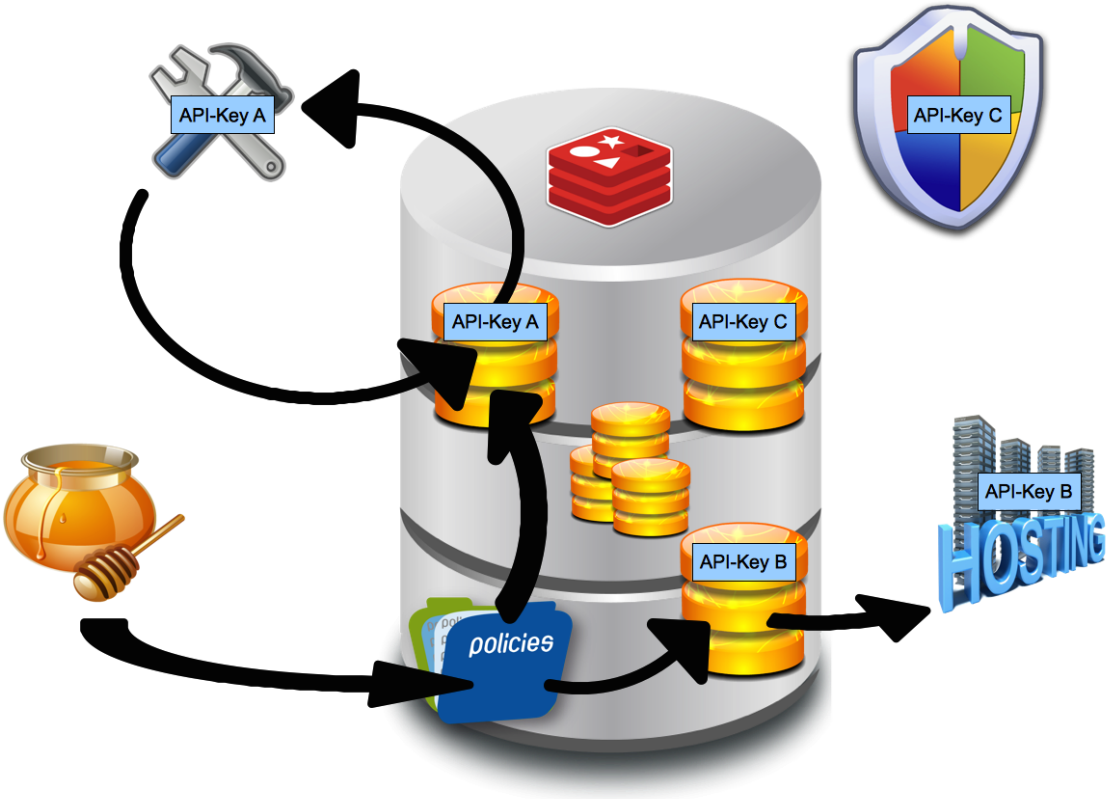

Figure 3: Policy based data exchange through the CCH

The Community Portal manages access policies for the stakeholder and allows each contributor full control about his submitted data. Restrictions could for instance deny data sharing with competitors or based on legal requirements. Access is granted by APIs, ensuring that the CCH cannot be accessed without the proper code that the Community Portal provided just for this particular request or for a given timeframe.

2. Privacy and Legal Requirements

The overall legal concept of ACDC has been addressed in the deliverable **D1.8.2 Legal Requirements**. Conclusions and recommendations from this and the general legal requirements have been incorporated into the development and conception of the CCH.
Generally, all ACDC partners shall follow the following guidelines:
Whenever personal data is processed, various legal bases must be complied with.
As the CCH collects and transfers IP addresses (static and dynamic) and domains to stakeholders of the entire project. The CCH must consider the handling, storage and processing under legal requirements.

2.1. Dynamic IP addresses and personal data

Whether dynamic IP addresses qualify as personal data, is disputable. A clear assignment is not possible, unlike in the case of static IP addresses. The starting point for this differentiation of opinions is the element of "determinability" *according Section 3 I Federal Data Protection Act.*[1]
The assignment of a dynamic IP address to a customer by the Internet access provider is merely temporary. In this way, the anonymity of the Internet user is guaranteed. Even if the IP addresses is stored by the server operator, a long-term association between the address and a user identity is not possible. From the IP address as such there exists no direct relationship to a particular person, so that this first needs to be ascertained.
As the access provider in their own right undertakes the allocation of IP addresses, and because it is relatively uncomplicated for the access provider to ascertain this relationship between an IP address and the user's identity, the above-mentioned cases where other parties, such as the mailbox provider, collect and forward dynamic IP addresses, remain disputable.

2.2. Relativity of the Relationship to the Person

One argument/legal opinion deals with the relativity of the personal reference, and applies according to Section 3 I Federal Data Protection Act for the assessment of the determinability, based on whether the responsible party can ascertain the relationship to a natural person with the means normally available to them and without disproportionate effort. In particular, a differentiation is made on the basis of whether the de-anonymisation is possible with proportionate effort. This should, however, only be possible for the access provider. A third party (here, the mailbox provider) is only able to identify the user behind the IP address with the help of the access provider, who, in turn, is legally not permitted to provide this information to third parties. The theoretical possibility of an identification of the user does not correspond to the aforementioned definition of determinability.

2.3. Objectivity of the Relationship to the Person

According to this argument/legal opinion, it is not relevant whether a disproportionate effort is required to de-anonymise the IP address. It is simply sufficient that the theoretical possibility of a link of any form exists between the email address and a natural person. This is regardless of the fact that the determinability of the person in the legal sense is only possible when the person is identified using legal means. In this argument, the Data Protection Act is specifically designed to protect against the misuse of data, meaning that such a limitation of the term determinability would not be justifiable. The objectivity of the relationship to the person is also supported by recital 26 of the *Data Protection Directive 95/46/EG.*[2]. The Art. 29 Working Party also assume the absolute definition of the term. In recital 26 of the EU Data Protection Directive 95/46/EG, it is unambiguously defined that all means that could be used by the party responsible for the processing, or by any other party who

[1] http://www.gesetze-im-internet.de/englisch_bdsg/englisch_bdsg.html
[2] http://www.dataprotection.ie/docs/EU-Directive-95-46-EC-The-Recitals-Page-1/90.htm

could be reasonably considered, are to be taken into account in order to establish whether a person is determinable.
As it cannot be ruled out that third parties possess the additional knowledge required for the ascertainment of the relationship to the person, it depends in actuality on the judgment of the probability of a potential identification. Dynamic IP addresses can also be used by third parties, with the help of log files of the Internet Service Provider's (ISP) individual connections, leading potentially to the identification of the individual person. Therefore, it must be assumed that it is possible to ascertain the relationship to the person for dynamic IP addresses and that the data protection laws are applicable.

2.4. Legality of data processing in Germany

The physical hosting location of the CCH is in Germany. Therefore, additional regulations and law considerations have to be taken into account. In general, personal data processing is legal in Germany, if the prior consent of the data subject is granted or law permits it. The evaluation of national Law related to the objectives of the ACDC project refers to the following requirements.

2.4.1. Prior consent

According to Section 4 of Federal Data Protection Act the collection, processing and use of personal data shall be admissible only if permitted or prescribed by this Act or any other legal provision or if the data subject has consented. Further, the collector of personal data (the CCH) must comply with Section 4 a of Federal Data Protection, in particular the data subjects shall be informed of the purpose of collection, processing or use and, in so far as the circumstances of the individual case dictate or upon request, of the consequences of withholding consent.

2.4.2. Permission by law

According to Section 28 (1) of Federal Data Protection Act the collection, storage, modification or transfer of personal data or their use as a means of fulfilling one's own business purposes shall be admissible when needed to create, carry out or terminate a legal obligation or quasi-legal obligation with the data subject, in so far as this is necessary to safeguard justified interests of the controller of the filing system and there is no reason to assume that the data subject has an overriding legitimate interest in his data being excluded from processing or use, if the data are generally accessible or the controller of the filing system would be entitled to publish them, unless the data subject's legitimate interest in his data being excluded from processing or use clearly outweighs the justified interest of the controller of the filing system.

2.4.3. Justified interests:

The data processing is necessary within the meaning of based on the collection, storage and transmission of strange data, so data subjects can be informed and further investigation can be initiated.
Whether the legitimate interests of the person concerned predominate, must be examined in the context of a balance of interests. Here the informational self-determination must be considered, which results from Article 1 und 2 of Basic Law for the Federal Republic of Germany (Grundgesetz, GG). An argument against an overriding legitimate interest of the person concerned is that only conspicuous data will be stored.
Moreover, it is precisely in the interest of the person concerned, when he gets informed about the conspicuousness regarding his data. The person concerned may take further steps for investigation and can take action to prevent further abuses of its data. As far as the data processing serves the recognition and containment of data misuse and moreover, to avoid massive harm and major disruption to the telecommunications infrastructure, the collection and transmission of personal data is justified.

As a result, the data processing in accordance with the principles set out above is admissible. It is possible either to take action on the data subject's prior consent or a statutory basis.

2.4.4. Granularity of the shared data

Some tools will report finite lists of IP addresses that might represent command and control servers of botnets, others will simply report that TCP/IP connections have been received to IP addresses that have no services and are potentially suspect. A malware sample could be extremely detailed, including file hashes, libraries it utilises, registry entries it created, etc. Therefore, it will be very important that the Data Exchange format used is able to represent data that varies in granularity. It is intended that the correlation capabilities of the CCH will allow data granularity to be increased when required by cross-referencing data fed by multiple tools. This can be vital in order to build a more complete picture of suspicious activities.
ACDC data controllers (CCH and Concentrators) shall put in place mechanisms to enable data subjects to enforce their right of access and related rights (right to confirmation, right of access to one's data, and right to rectification, erasure and blocking). The API key system to be used in the Community Portal, while designed as a data confidentiality and security mechanism, holds the potential to also be used as a tool to facilitate the exercise of data subject rights. Through the API key system, the platform has an overview of which data is sent to the CCH and to whom the data relates. Such control is likely to help the CCH respond to the exercise of data subject rights in a fair and responsible manner.
The CCH is the main element of the ACDC project. Via this platform, data collected through the detection tools will be aggregated and further analysed with the purpose of generating data feeds to interested data controllers. This single EU common report will facilitate information exchange across the Union and enabling appropriate response against malware. The CCH is thus a gathering platform, which combines data collected by sensors and other tools directly placed in public networks and personal devices to be further analysed. The results of the CCH are thus shared with trusted partners that have previously requested to receive the data feed. A party interested in subscribing to a data feed must register via the Community Portal, which will grant access by assigning an API key. However, access to the CCH data feed is limited in light of the specific and legitimate interest a party may hold. The relation between requests and nature of access will be measured and granted by the Community Portal based on the relationship and level of trust of a given party.
Before sharing any data with the platform, countries are expressly requested to pay attention to their national laws. So partners are fully responsible for their network sensors and malware website analysis tools, as the are the controllers of the processing within these tools. Thus said, they hold full responsibility to comply with national legislation before engaging in information sharing with the CCH or other partners. Such duties must be borne by partners contributing data due to the impossibility of having the CCH verifying those compliances.
ECO (Association of the German Internet Industry) is managing and controlling the CCH. This control covers a three stage processing, which includes access to the data shared by data collectors (such as ISPs, end-users, computer security companies, banks, etc.), further processing relating to the analysis and aggregation of the data received, and a final processing concerning the distribution of this data with trusted parties according to the rules of the Community Portal. Is is important to recall, that the automation processes deployed within the CCH minimise human intervention. The information distribution on a need-to-know basis assures that only data pertaining to the IP address space or Autonomous Networks or special legal interest of a specific partner is shared to this partner. Furthermore, only parties holding legitimate interest over personal data will receive redistribution of this data. Finally, not every partner receiving feeds from the CCH will have access to a users' personal data, for the reasons explained above.

2.5. Data privacy considerations

With many types of shared data the sensitivity may depend on if the data is describing the victim of a cyber-incident or the perpetrator. Also objects such as files and e-mails could contain either sensitive

or non-sensitive information. Categorising data to indicate the privacy issues may be required, along with the ability to anonymise information before it is shared. An illustration of the potential types of data objects that could be stored in the CCH as part of the ACDC project, and the potential privacy concerns can be found in Annex: *CybOX*[3] *Cyber Observables and Privacy*.

WP1 Deliverable 1.8.1 defines the legal issues regarding the dissemination of information, and technical solutions delivered by WP2. The technical implementation will be required to conform to the findings of this deliverable.

Article 7(f) requires a proportionality assessment, i.e. the legitimate interest of the data controller (ensuring the security of the network) must be weighted against the need to protect data subjects' fundamental rights (confidentiality of communications). In the case of ACDC, the need to fight botnets (as part of the security of the Internet) should be balanced against the need to protect users' confidentiality of communications. While fighting botnets is clearly legitimate, the proportionality assessment will bear upon the adequacy and necessity of the means used to achieve this goal, thus on the design of the ACDC solution. In this case, two data processing operations should be legitimized: the sharing of personal data with the CCH and its further processing for fighting botnets. This processing will be performed by the CCH and shared with a legitimate interested party, like ISPs, webmasters and hosts.

The Today`s cybercrime has an international character and imposes a significant threat to the Internet. Therefore, there is a strong need for providing a central platform such as the CCH to efficiently provide countermeasures. Providers, webmasters, hosting services, and computer security companies cannot tackle large-scale infections individually. The need for cooperation and information sharing is thus crucial. The automated and non-human operated character of the CCH reduces likelihood of unauthorized or abusive intervention.

With regard to the adequacy of the means used, the CCH environment consists of data processing that have a strong and concrete potential to reduce botnets across the EU. The CCH is thus a suitable and adequate environment to the objectives pursued: the tools deployed at the clearing house are tested and efficient mechanisms capable of promoting knowledge and sharing data of infections. This will facilitate disinfection, as the data feed provided by the CCH will enable partners to alert their infected users and redirect them to the National Support Centre for cleansing. The purpose of the CCH is thus protecting end users and networks from botnets, promoting prevention, detection and disinfection. The information processed at the CCH is not intended to be used against individuals, left alone the possibility of ISPs, webmasters and web hosts to take the necessary legal measure to safeguard their networks against malicious users. To this end, ISPs and all partners are requested to, in accordance with their national law, decide upon the remedies, sanctions and means to be given to users to enable their adequate defence. It is thus a duty of data controllers to ensure that data subjects' rights are fully respected and that the necessary remedies and opportunities for their exercise are put in place.

Finally, the rules of the Community Portal together with the numb and automated character of the CCH are designed to ensure the minimal possible impact on the fundamental rights of data subjects in being taken into account. Removing the processing of personal data would diminish the capability of the CCH to a level where the purpose is no longer attainable. However, input and output data are controlled and ranked by the Community Portal, which also enables sharing preferences and restricts access to data according to partners' legitimate interests. For instance, ISPs receiving data feeds from the CCH only have access to information related to their own range of IP addresses, never to data which is outside of their scope in terms of IP address range. This guarantees sufficient levels of confidentiality of communications and lesser invasive means on user data.

[3] https://cybox.mitre.org/

3. Data Storage

The CCH, same as the overall ACDC software architecture **(Deliverable D1.1.2)** is designed to be flexible to limit the barriers that might limit stakeholders from submitting data. Particular to the CCH, this flexibility refers to the way to receive data from the different tools or aggregators. Though the CCH is able to receive data in a wide range of formats, certain mandatory and basic requirements have been defined and specified. These standard requirements are part of the standard ACDC Data exchange format. These data formats have been defined as part of the work in WP1, which is dedicated to standards and specifications. The deliverables outlining these requirements are **D1.7.1/2 "Data Formats Specification"**

The long-time data storage functionality of the CCH supports data correlation and basic data analysis, providing basic functionalities like:

- High data capacity with easy and seamless storage expansion;
- The ability to store data represented by the ACDC Data Exchange Format with minimal mapping to storage types;
- Support for fast querying of stored data;
- The ability to create triggers and watch lists in order to facilitate automated analysis;
- The ability to restrict views of information based on the permissions of the viewer.

3.1. CCH Infrastructure

All servers of the central infrastructure are physically stored and operated in a ISO-27001-certified Data-Centre owned by DE-CIX. The data centre has an electronic access control system, with a 24/7 camera monitoring of entrances and server rooms secure the servers against unauthorized physical access. Remote access is only possible via SSH with Key Authentication and limited to a very few system administrators, with user rights and the possibility to get "root"-access via "sudo" only. The default root login is disabled by default.

The web servers as entry points for the entire system are completely independent systems to the database server. They run on the latest stable Debian version, access is only possible through SSH via key authentication. The CCH uses an NGINX, configured as a reverse proxy, listening on port 3000 (SSL on). All NGINX proxies connect to a "Thin Webserver" which is used as an API server and it is not reachable from the Internet itself, because it did not bind any TCP sockets by itself.

All other servers (Database and Filestore- Server) are in a virtual network. They are only accessible through the API-providing web server, with a valid API key, and for system administration purpose through SSH via key authentication only.

3.1.1. Hardware used in the project

Centralized Clearing House: Consisting of REST-API, Redis, MySQL-Database, Mongo-DB-Database, one XMPP-Server.

These standardized hosting services are offered by the DE-CIX Management GmbH.

By default, DE-CIX offers Debian based virtual machines at the DE-CIX owned, highly available ESX Cluster. Here numbers and size of the CPU, memory and disk capacity can be provided as needed and justified by the requirements. Besides Debian, other operation systems such as additional Linux distributions or Windows can be provided on demand.

For the ACDC, following virtual machines have been requested by eco and set up by DE-CIX

- 3x server instances for REST API: 1x DEV server - 2x PROD server
- 2x instances MySQL / Redis server (master / slave) configuration
- 2x instances Mongo DB server
- 1x instance XMPP server
- 1x instance statistic server

Server:
4 cores (at least 2.6 GHz.); 16 GB RAM; 3 TB disk storage
• The DEV system needs less CPU power (2 Cores - 16 GB RAM, 500 GB), but must be easily adaptable for testing purposes to the live systems
• PROD server: The disk storage does not have to be in full capacity at the initial configuration, but must be easily scalable (initial configuration: 4 cores, 8GB RAM, 1TB disk storage)

MySQL / Redis:
each server requires 4 cores (3.3 GHz); 64GB of RAM; 2 TB disk storage
The provided disk capacity can be made highly available (initial configuration: 4 cores, 32GB RAM, 500GB disk storage)

Mongo DB:
each with 8 cores (3.3 GHz); 64GB of RAM; 3 TB disk storage
• The provided disk capacity can be made highly available (initial configuration: 4 cores, 32GB RAM, 500 GB disk storage)

XMPP server:
4 cores (at least 2.6 GHz.); 16 GB RAM, 500 GB disk storage
• Initial configuration: 2 cores, 4GB RAM, 500 GB disk storage (Start configuration: 2 cores, 4GB RAM, 500GB hard drive capacity)

Statistics Server:
4 cores (at least 2.6 GHz.); 16 GB RAM, 360 GB disk storage
• Initial configuration: 2 cores, 4GB RAM, 360 GB disk storage (Start configuration: 2 cores, 4GB RAM, 360GB hard drive capacity)

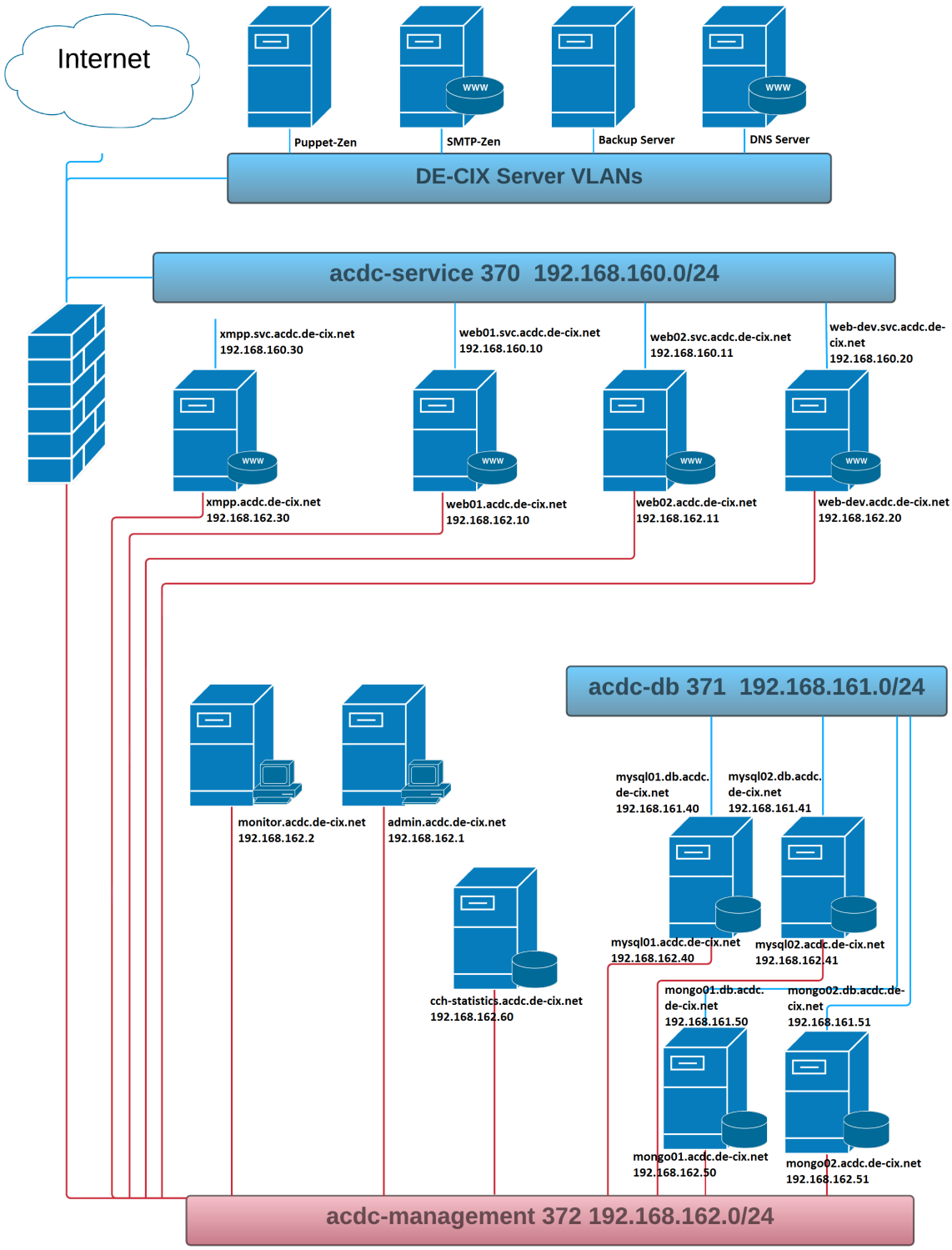

Figure 4: *ACDC infrastructure - hosted by DE-CIX*

3.2. Hosting and IT Security at DE-CIX

As DE-CIX operates a very important part of the Internet infrastructure, IT security is deemed as essential at DE-CIX in order to provide reliable and robust Internet Exchange (IX) services. To achieve a high level of IT security, DE-CIX has established and maintains various activities. A selection of these activities relevant for this technical offer are listed below.

In around 2005, DE-CIX started managing the IT security of its production network following the rules and guidelines of BSI IT-Grundschutz / ISO 27001 (IT-Grundschutz). In 2010 DE-CIX was ready to be audited on full compliance with IT-Grundschutz, which resulted in certification. IT-Grundschutz requires yearly re-certifications, which are based on a renewed detailed security risk analysis, followed by a definition of how security risk will be handled, and the selection and implementation of safeguards. The picture below shows the ISO 270001 certificate issued by the BSI to DE-CIX valid until 14.03.2016.

Figure 5: ISO 27001 certificate issued by BSI to DE-CIX

Two employees at DE-CIX, namely Arnold Nipper and Jan Stumpf, act as Security Officers. Their responsibility is to manage and drive all security related topics. This includes overviewing and reviewing the management of incidents, security protection measures and risk mitigation controls. An example for a very successful risk mitigation control at DE-CIX is the “Security Updates Watching Group”. Since this group has been live, no system hacks have occurred as a result of missed security updates. The group consists of four employees. The task of the group is to read security bulletins from vendors and publically available vulnerability reports (such as the full disclosure mailing list) on a regular basis and extract security information. This security information is evaluated in the context of DE-CIX and the severity is accordingly re-assessed. Based on this new assessment, actions are launched on different levels if needed, to mitigate the risk arising from the bulletin or vulnerability.

3.2.1. Guidelines to Secure DE-CIX Data Centres and Equipment

This section contains the guidelines to secure the DE-CIX data centres and equipment. The documents are party copied from the official DE-CIX guidelines used in the ISO 27001 IT security management system.

3.2.1.1. Data Centers

Functional Specification of the Service

The DE-CIX is hosted inside different data centres that are spread across Frankfurt. The two pictures below are taken from DE-CIX data centres.

Figure 6: DE-CIX data centre in Frankfurt, Germany

Figure 7: DE-CIX equipment in a DE-CIX data centre

As all of them are operated by professional colocation providers DE-CIX gets the following additional benefit (but not limited to):

Security

Access to the DE-CIX facilities is restricted and secured via i.e.

- Local security guards
- Access-lists
- Security cameras
- Key cards
- Mantrap
- Biometric systems
- Alarm systems
- Railings
- Special door and rack locks
- Security guard escort for customers

Monitored 24/7 by Colocation Provider

Power

Full power backup via generators in combination with UPS systems (available for dual-feed)
Available in different configurations, depending on DE-CIX needs (i.e. AC Power 16 or 32 Amps, DC Power 48 Volts...)
Monitored by colocation provider as well as by DE-CIX own monitoring equipment

Temperature control

Redundant cooling equipment
Down floor cooling concept
Room temperature is maintained at 18 to 25 ºC with a humidity level of 40% - 60%
Monitored by Colocation Provider as well as by DE-CIX own monitoring equipment

Fire protection

Various fire and smoke detections units (raised floor, ceiling)
Gas-based fire suppression systems (automatic)

Hand-held fire extinguishing systems
Monitored 24/7 by Colocation Provider

Water protection

Water detection systems inside the raised floor
Monitored 24/7 by colocation provider

On-Site Support

24/7 technical on-site support available

Connectivity

Installation service for various kind of cabling (copper, fibre)
Connections could be made to a wide range of Carriers, ISPs, etc.

3.2.2. Guideline for Equipment Security

This guideline covers all equipment that is directly or indirectly used to operate the services at the DE-CIX:
Server, Router, Switches, Appliances and Optical Fibre Equipment.
This guideline does not cover client systems that are used to get remote access to the DE-CIX infrastructure. It does also not cover systems operated by DE-CIX customers like customer peering routers and customer out of band terminal servers operated by the hosting providers.

3.2.3. General Strategy

There are two kinds of systems at DE-CIX:

a) Systems with normal end user access. Those systems are
   - Terminal Servers
   - Personal Laptops

b) System without normal end user access.

In general, a strict strategy is implemented with respect to authentication and system access, especially when logging in from external networks. Inside a system, a liberal strategy is implemented for services inside a system and a strict strategy is implemented with respect to network services, especially on exposed systems.
Different Services must be implemented on different systems to guarantee proper isolation.
Basically, we differentiate between normal local users and administrators. The latter have the ability to gain root/administrator access without knowing the root/administrator password after login with their normal user credentials if the underlying operating system allows that (e.g. sudo on Unix/Linux, separate admin account or UAC on Windows). If that is not possible, the user directly logs in with administrative rights (e.g. Windows Admin User, Cisco). A strong strategy is implemented when it comes to authentication. Authentication and administrative rights are controlled from a central point. Changes need approval of a security officer.
For DE-CIX equipment, all IT-Grundschutz measures must be met.

Physical Environment

All DE-CIX equipment must be located in secured data centres with Strong physical access control access for a well-defined list of persons. The list of persons is subject to approval by a Security Officer.

1. Storage and Transport of Equipment

In general, all DE-CIX equipment must be stored in secure locations. Secure locations are
DE-CIX office Special storage locations at data centre sites with strong physical access control according to data centre standards.
If equipment must be transported between data centre, storage locations or the office, the following regulations apply:
In general, DE-CIX equipment is to be transported only by DE-CIX staff and with DE-CIX vehicles.

Project managers may decide to use external transport service partners from the following reasons:

- Organizational
- Financial
- Risk Management/Insurance
- Capacity

If DE-CIX equipment is transported by external service partners, it must be ensured, that

- All removable storage media are removed or erased
- All non-removable storage media are erased
- Firmware, Operating Systems and Configurations are installed form scratch after the transport, if a device was transported together with non-removed media.

Login Access

Login to DE-CIX systems is allowed by

- Local console login from within DE-CIX data centres
- Local console login via the server virtualization platform. The server virtualization platform itself is also subject to the login rules defined here.
- Remote console login via installed terminal servers, if such a device exists for the respective system.
- Remote login via a strongly authenticated and encrypted connection from external networks to a central entry point.

A Security Officer grants login access to DE-CIX systems to all DE-CIX administrators after approval. Administrative rights on DE-CIX systems is granted to all DE-CIX administrators after approval by a Security Officer

Documentation

All installations and changes must be documented according to the requirements documented in the change management process.

3.3. Database

The ACDC database consists of the following infrastructure:

- **Redis Database**[4]
- **mongoDB Database**[5]

The memory based Redis Database offers significant better performance compared to database systems writing every request or event into disk.
As the Redis database does not keep data for long time storage purposes, a mongoDB Database has been attached to the Redis Cluster. This mongoDB database stores and provides the data that is required within the project, e.g. for long-time measurements, research or historical data.
Both databases, Redis and MongoDB, support clustering. This gives the system the possibility to increase performance and storage possibilities with no downtime or interruption to applications.

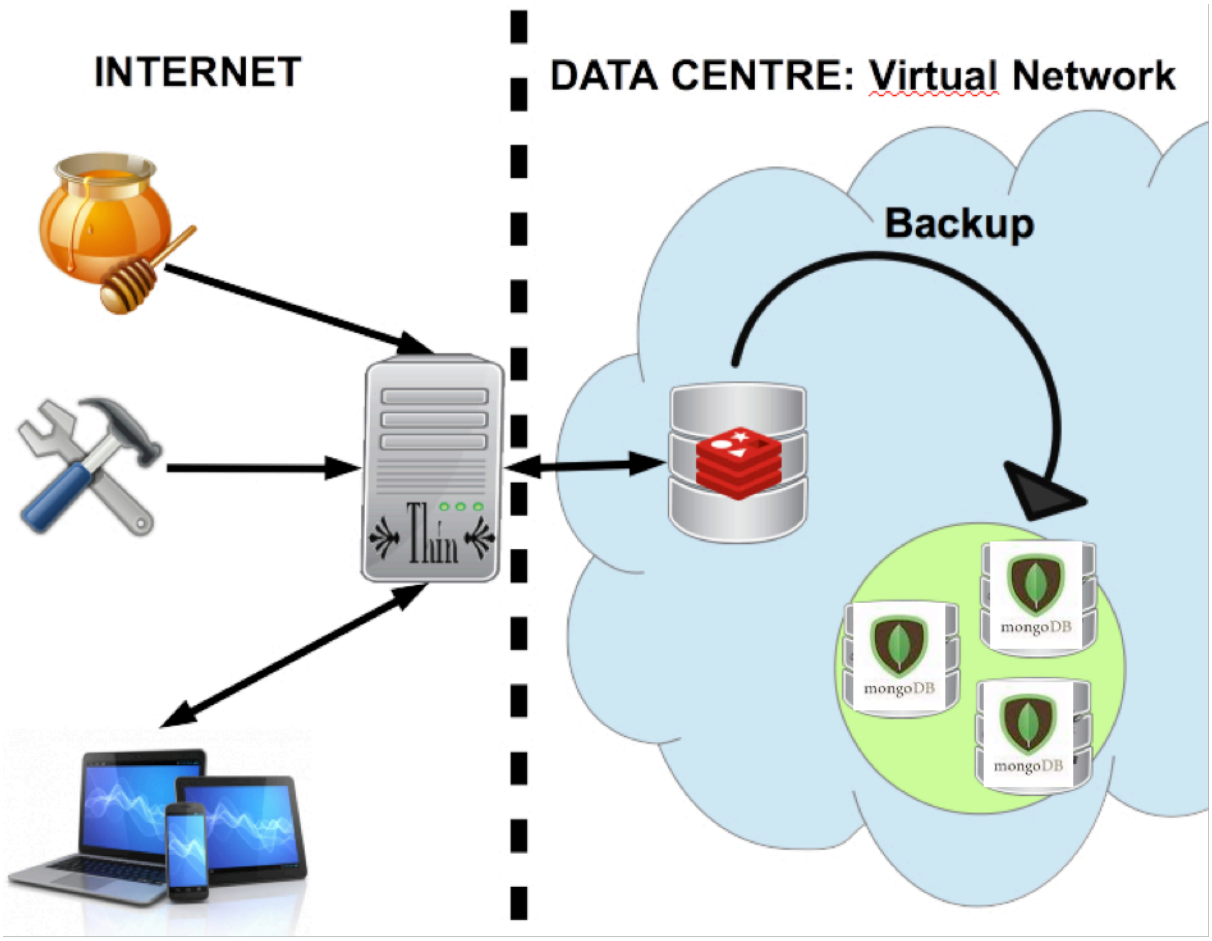

Figure 8: CCH Infrastructure

[4] http://redis.io
[5] http://www.mongodb.org/

3.3.1. Tools, Sensors or Aggregators

Tools, Sensors or Aggregators can only access the CCH through unique API-keys. An API-Key of any external component needs to authenticate itself through the API-Server, which is located on both databases (Redis and MongoDB). The specifications for any external tools have been defined in the following deliverables:

- D1.4.1/2: Tools for Malicious or Vulnerable Websites analysis and detection
- D1.5.1/2: Network Traffic Sensors
- D1.6.1/2: End Customer Tools

External Sensors enrich the database with additional information. This data feed adds new incidents or enriches existing incidents with additional information on possible activities related to botnet. A sensor can deliver incidents or details on a given incident to the CCH, which another sensor can receive and further process. In this case the second sensor should be hooked to a listener, watching the relevant data stream for its task.

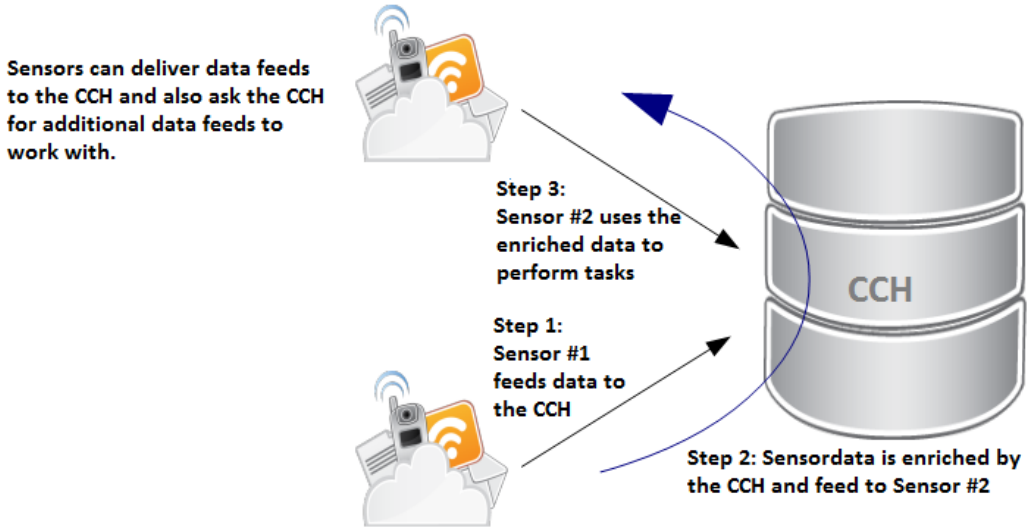

Figure 9: Sensors feeding data into the CCH

Each sensor has its own unique API-Key and the API-Key also includes the information about the schema for automated processing. Schemata are further explained in chapter 6.1 of this document and in deliverable **D1.7.2 Data formats.**

3.3.2. Data Aggregation

The CCH has a minimal set of operations, which can be performed on the gathered data. This covers the functionality of increasing counters for a certain event, grouping of same-type incidents, grouping by file-hash, domain, IP or ASN, addition of a TTL or basic information like first-seen, last-seen.

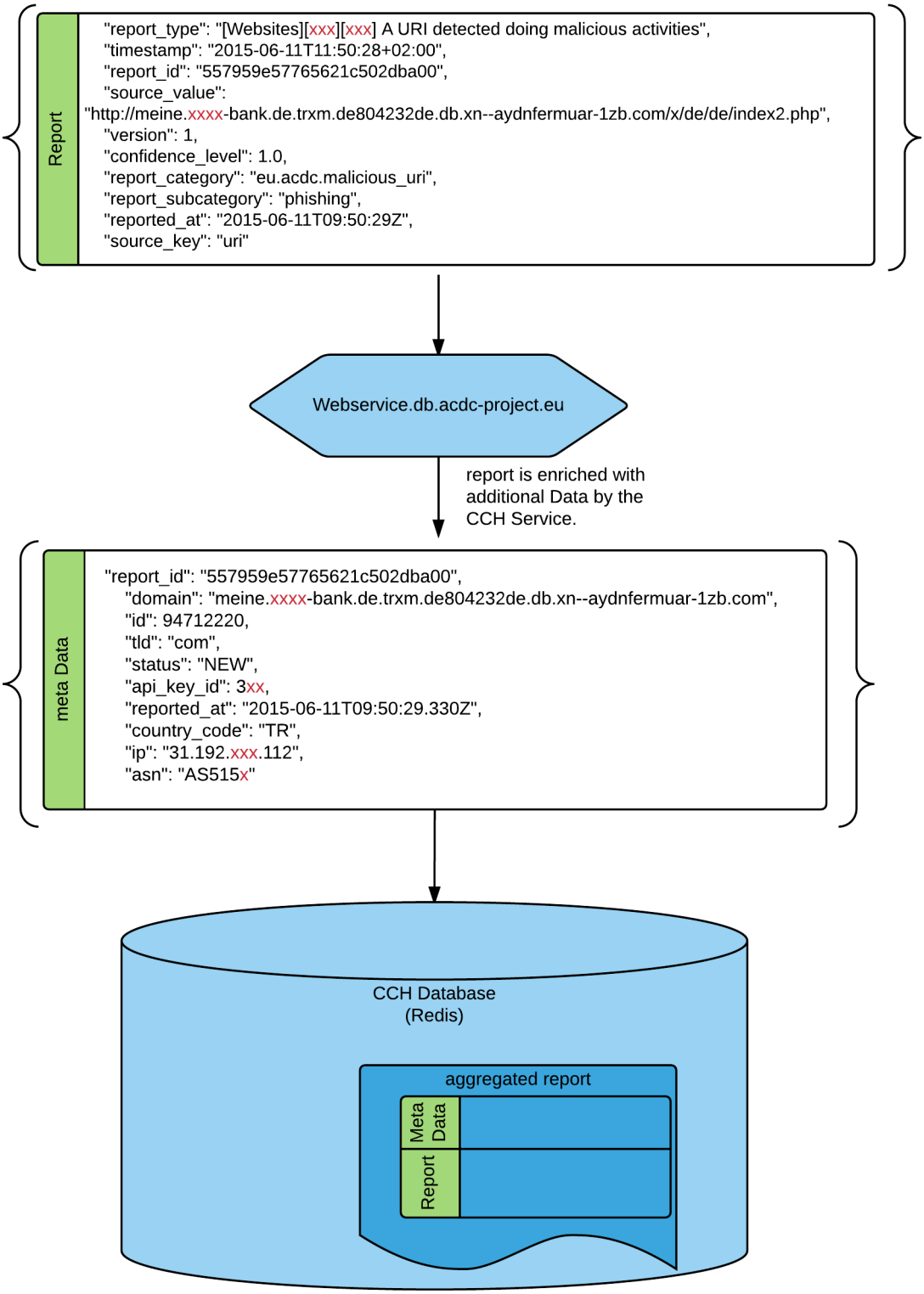

Figure 10 : Data aggregation example

3.4. Data Anonymisation

IPv4 and IPv6 addresses included in CCH reports are anonymised before saved to the persistent data storage (MongoDB and MySQL). This affects the following fields:

- If source_key equals “ip”, the source_value field will be anonymised
- If exists, the dst_ip_v4 field will be anonymised
- If exists, the src_ip_v4 field will be anonymised
- If exists, the dst_ip_v6 field will be anonymised
- If exists, the src_ip_v6 field will be anonymised

Anonymisation will be conducted as follows:

- IPv4 Addresses are transcript to IPv4-localhost: “127.0.0.1”
- IPv6 Addresses are transcript to IPv6-localhost: “::0”

Only reports that are published via the XMPP channels contain the full, unanonymised data sets for further processing. All later queries (e.g. API calls) will only return anonymised data.

3.5. Data Output

The CCH feeds a XMPP Server, where all data is streamed to channels.
We have chosen Prosody IM[6] as it is well maintained, documented and open-source, under the MIT[7] license. Therefore, the source code is freely available to download and use. The source code may be modified and used in closed-source applications.

Every (read) API Key connects to his own channel where the relevant data – the datasets this key is allowed to see – is streamed into in real time.
Datasets from Keys, which this Channels owner is allowed to see, are also streamed into the XMPP channel.

The data is streamed in the XMPP channel by following rules:

- If an organization has declared IPs, a IP range or a ASN Range in the Community Portal, then every incident that falls within that IP Range is automatically sent to the XMPP channel.
- Your read-key will get all reports from write-keys that are connected to yours e.g. that have accepted to share data with your key.

To identify which Key sent the dataset and to include additional Information, collected by the CCH to a given report, first a set of “Meta Data” is send. After the Metadata, the original report is streamed. The report is delivered in the schema it was submitted to the CCH.

The XMPP Server address:

XMPP Server: xmpp.001.eco.dedicateservices.com
Port: 5222

Login to the XMPP Server:

Logon Name / Jabber ID: user.API_ID@xmpp.001.eco.dedicateservices.com
Logon Password: API_Key

[6] https://prosody.im
[7] https://prosody.im/source/mit

The API_ID is returned by the creation of an API Key in within the Community Portal.
Example output for the CP, while creating a new API Key:

{
"id":1111,
"access_token":"b1335e2dc580c361563a81d74951ac1d",
…..
}

A dataset in the XMPP channel is divided into 2 fields:

- “report” The original report that was sent to the CCH
- “meta data” additional information on the ASN / IP / Country-code etc. that was enhanced by the CCH.

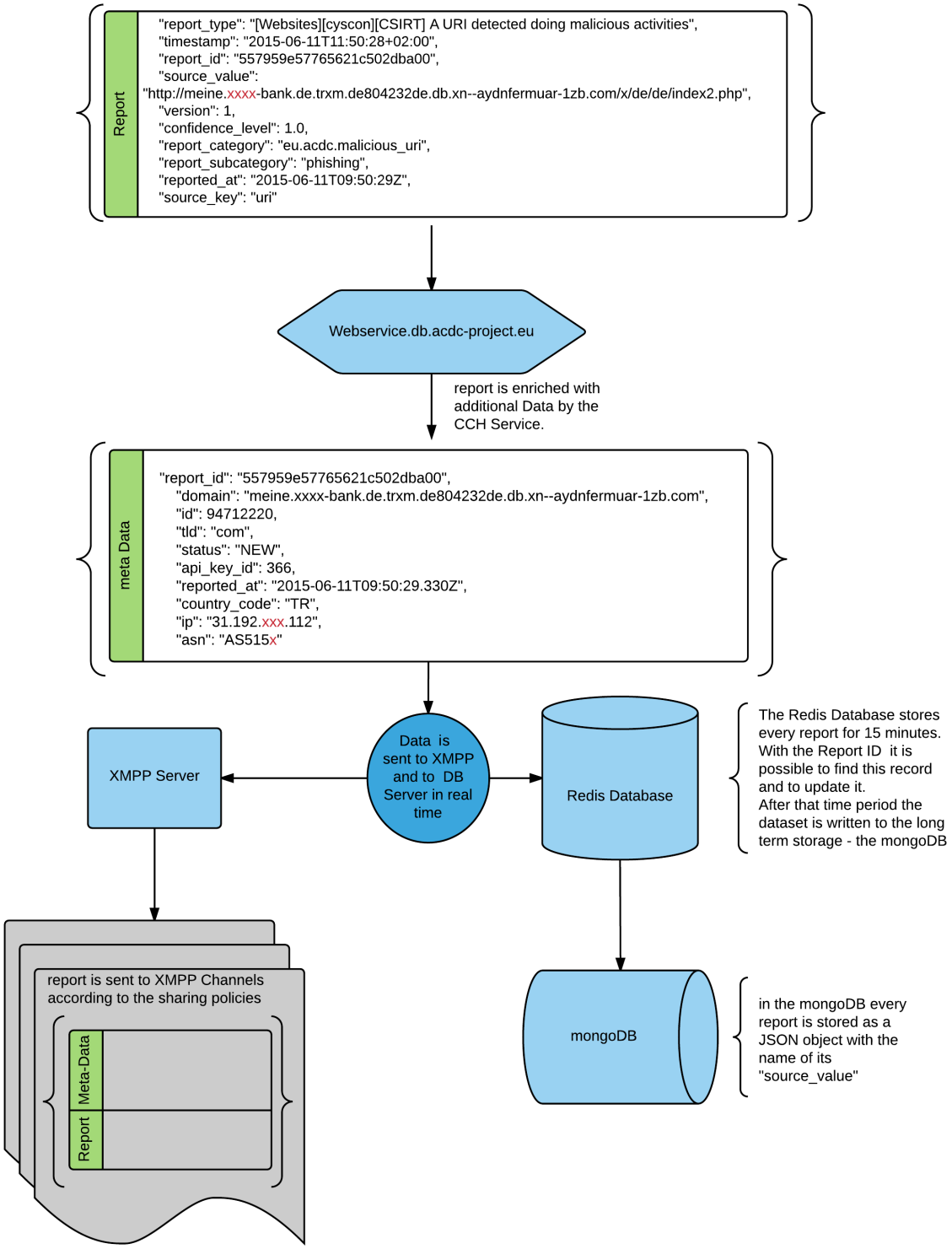

Figure 11 : Data sent to the CCH, distributed to XMPP and to DB Servers

example :
incident streamed in the XMPP channel of read key 657 by one of the ACDC Project members


```
2015-06-16 07:25:49,656 - xmpp657.py - DEBUG - {
 "report": {
  "timestamp": "2015-06-12T22:10:01+02:00",
  "c2_ip_v4": "98.126.xxx.18",
  "source_value": "98.126.xxx.18",
  "report_subcategory": "http",
  "report_category": "eu.acdc.c2_server",
  "source_key": "ip",
  "confidence_level": 0.9,
  "c2_mode": "plain",
  "report_id": "557b59e577656235f5ff1100",
  "reported_at": "2015-06-12T22:15:01Z",
  "version": 1,
  "report_type": "[DDoS][DDoS_Monitoring_Tool][IF-IS]"
 },
 "meta_data": {
  "id": 95398026,
  "api_key_id": 508,
  "report_id": "557b59e577656235f5ff1100",
  "status": "NEW",
  "ip": "98.126.xxx.18",
  "reported_at": "2015-06-12T22:15:01.469Z",
  "asn": "AS359xx",
  "tld": "com",
  "domain": "98.126.xxx.18.static.krypt.com",
  "country_code": "US"
 }
}
```


3.5.1. Message Storage and size.

XMPP messages are stored in the XMPP queue, so a service break or a temporary disconnect between the CCH and a subscribed data receiver will not cause data losses.
Due to security and stability reasons, however, the CCH will not store these messages for a longer period of time.
Offline messages will be deleted after one (1) day of provision in the offline storage. Additionally, all offline data, which exceed the file size of 100MB, will be deleted permanently. Full report information cannot be recovered! Subscribers are requested to fetch their messages in a timely manner.

3.6. User Rights Management

The Centralized Data Clearing House collects and stores data from different sources. This data, along with additional information enriched through ACDC processing, will be distributed to either the data provider itself, or to a third party with the privilege to receive such data.
Every data contributor will gain full control about his submitted data. This policy has been introduced, as either legal boundary restricts data sharing with stakeholders outside a certain country or region, others may have concerns sharing their data with Law Enforcement agencies, and other partners simply do not want to share their knowledge with competitors, like Anti-Virus companies or rivalling research organisations.
The Philosophy of ACDC follows the directive that data sharing follows the mutual approach within a community. An organisation not willing to share data with their competitor will not receive access to that organisations' data either. Such policies are handled through the Community Portal, the front-end of the entire project. A user requires applying for being registered the ACDC Community Portal (https://communityportal.acdc-project.eu), and the portal manages the permissions and stakeholder relationships.
The portal itself allows a participating organisation to specify with whom they are going to share data and with whom not. The specifications can be set based on different stakeholder groups, regional settings or directly to a certain organisation. The model and concept, same as the different user groups, are handled in the deliverable **D6.2.1 ACDC Social Platform**, namely in chapter 8, "The ACDC Portal Applications".

3.6.1. Security Specifications

The Community Portal maintains the access management of the CCH, and the various data sources are graded depending on their level of trust. Moreover, data feeds are provided on-demand and limited in light of national laws and special interests of stakeholders. To the extent that the CCH receives and releases non-personal data, no legal obstacles can be raised.
The Community Portal user right management has been designed as a mechanism with respect to ensure the fundamental rights of data subjects. However, removing the processing of personal data from this context (e.g. IP addresses, domain names, email addresses) would diminish the capability of the CCH to a level where the purpose is no longer attainable. Input and output data are controlled and ranked by the Community Portal, which also enables sharing preferences and restricts access to data according to partners legitimate interests.
Therefore as a recap, ISPs receiving data feeds from the CCH can only have access to information related to their own range of IP addresses, not to any other unrelated information. This guarantees sufficient levels of confidentiality of communications and lesser invasive means on user data. The only machine that is visible to the Internet is located at *webserver.002.eco.redacted*.
The real-time access to the database is handled through so-called Channels. Each owner of an API-Key receives access to the real-time database, enabling immediate notifications about events he is legally allowed to receive only through his dedicated channel. The setting and registration of an ASN owner is handled through the Community Portal, additionally secured by an API-Key and authenticated through the implementation of the *Trusted Introducer* [8] approach, which was implemented by DFN-CERT.

[8] https://www.trusted-introducer.org/

3.6.2. Reputation & Trust

The operator of the CCH, namely ECO, currently determines the reputation and trust of data submitter manually. Their judgements are based on the extended experience in the IT-Security, following guidelines like

- Data quality of previous submissions, determined from other projects, initiatives and partnerships.
- Reputation of the data provider in the industry, 2nd opinions.
- Attribute of the submission (Unfiltered data from an end-customer tool like a report button to submit Spam)
- Pre-verification, e.g. in an aggregator
- Type of organisation (e.g. an AV-Company is supposed to submit verified data)
- Trust by others, 3rd party recommendations (e.g. at Virus Total)
- Known vs. unknown data source
- Frequency of a data feed (Single, continuous or periodic feed or data provider)
- Error rate, false positives
- Etc.

In this stage of the project, no detailed process for determining the reputation and trust has been defined yet. The partners simply have to trust each other and their expertise in this area, same as to trust all other partners to provide quality data. Following the community approach, each partner still has to consider using own quality assurance measurements, before including data from the CCH into his own environment or products. In business, a Quality Assurance department would handle such responsibility, but as ACDC is a pilot project, the objectives just require a proof that the detection of botnets and malware are feasible with this current implementation.
Long-term plans, based on a sustainability and exploitation plan, will have the implementation of such QA measurements on the "product roadmap". The vision is to combine such intelligence with a self-learning and community-supported approach.

3.6.3. Trusted Introducer

The implementation of the Trusted Introducer[9] (TI) approach as a model for authentication has been added to the ACDC roadmap. This service established by the European CERT community was identified as a suitable approach to safeguard a trusted environment within the ACDC-Community. Following their concept, the TI-adaption within ACDC should ensure that community members especially involved in data sharing or decisions-and policy-making meet a defined level of trust, credibility, ethics or standards. In case of the participating CERTs within ACDC, a direct implementation of TI has been considered and set to a stage of evaluation.
The identification of the stakeholder is done by the Trusted Introducer (TI) body, which provides this service to its accredited as well as certified members. TI provides identification and authorisation methods for CERTs as well as enforcing a high degree of data quality. All its member teams are required to regularly update their member data to keep it up-to-date within four months.
Any CERT following this workflow MAY receive data for its constituency, which is defined by IP networks and ASNs. Domain- or TLD-based constituencies are not yet possible due to the ownership of data belonging to these not yet being decided inside ACDC.
The stakeholder assures towards TI that it is eligible to receive data for the constituency scope as defined inside its member data sets and will then receive X-ARF data by email. The Format definition can be found on Deliverable 1.7.2, section 9 and in this document in section 3.11.4

[9] https://www.trusted-introducer.org/

3.6.4. API-User Types

The API concept is based on a hierarchic model of a three-stage API-Key distribution (CCH-Manager/CCH-Key-Manager/CCH-Key-User). Besides CCH-Manager-Keys, the community-portal is able to assign individual keys to subscribers of the ACDC community. With such a Key, each partner (CCH Key Manager) is capable of creating and assigning individual API-keys (CCH Key User) to own sensors, nodes or even sensors running on end-customer devices through the Community Portal, where each key gets registered.
Based on the settings in the Community Portal, a limitation of the provided keys per organisations can be set, same as expiration date or a revoke of the key. At this stage, all distributed keys have been limited until the end of the project duration. The CCH Key-Manager keys are not visible to an organisation and just used internally.

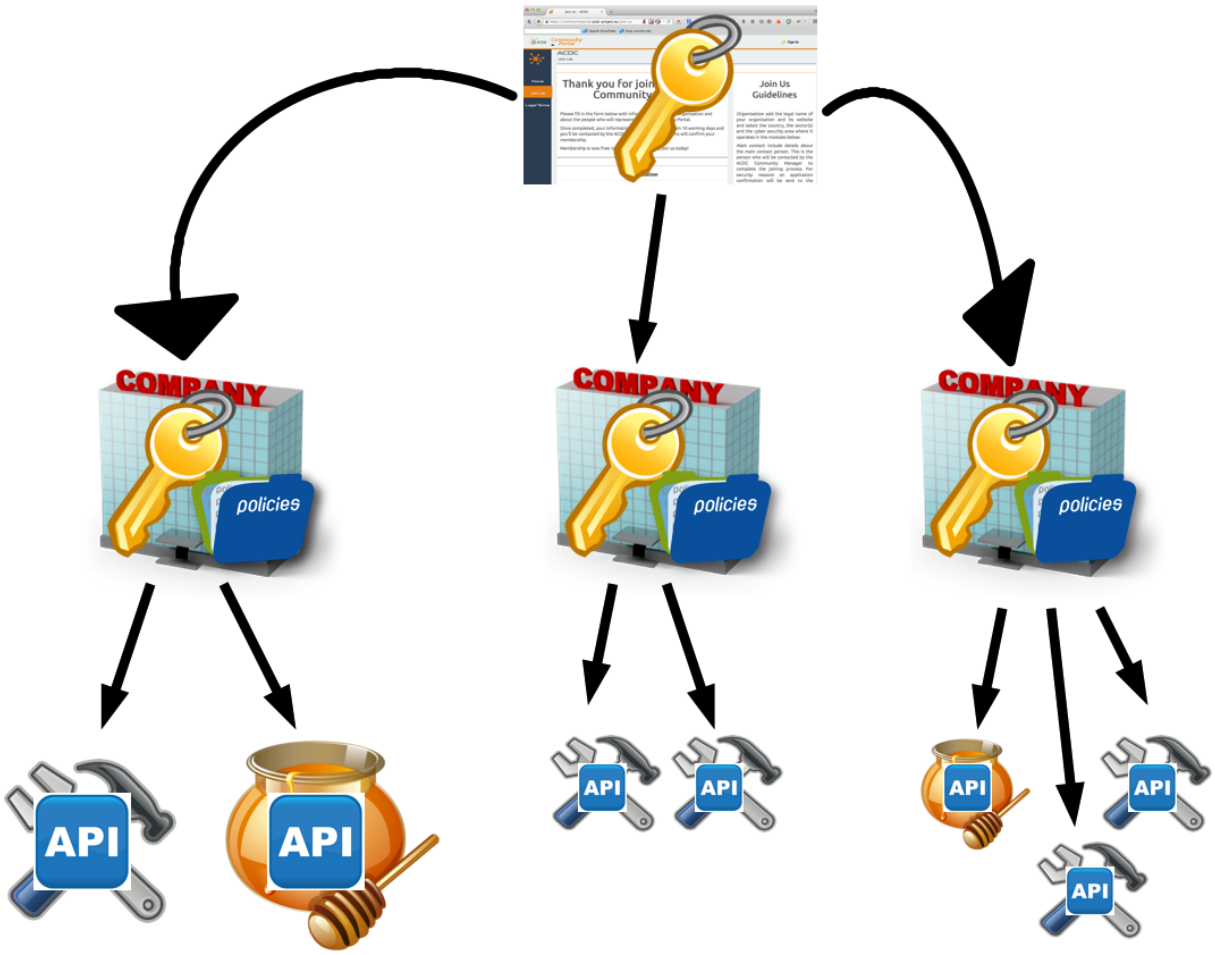

Figure 12: How API Keys are generated

Further information on the Key Management is part of deliverable **D6.2.1 ACDC Social Platform** as part of chapter 8.4

3.6.4.1. CCH Manager

The CCH Manager Role is assigned with the highest permissions within the key management groups, their keys are flagged in the CCH with the "super_user" flag set to true (see 3.7.1 API Key creation). The permission of these "super-user" Keys includes:

- Manage group parameters
- Add organizations to the groups / distribute CCH Key Manager Keys
- Any of the actions of other key owner groups.

3.6.4.2. CCH Key Manager

This organizational role allows the user to manage keys belonging to its organization. CCH Key managers can perform following actions:

- Get list of CCH API keys of their organization
- Set/unset CCH Key User role to other users of the organization
- Invalidate API keys of the CCH Key-User in the organization CCH

3.6.4.3. CCH Key User

A CCH Key User can perform:

- Create a new API-key for his organisation.
- Specifying, if a key is in read (data retriever) or write (data provider) modus.
- Get list of created keys grouped by read or write mode.
- Update key
- Invalidate key
- Get Key (from a recovery process)
- Manage sharing policies:
- View pending request from and to Organizations
- Approve/ Deny request of data sharing
- View list of sharing policies in place and revoke

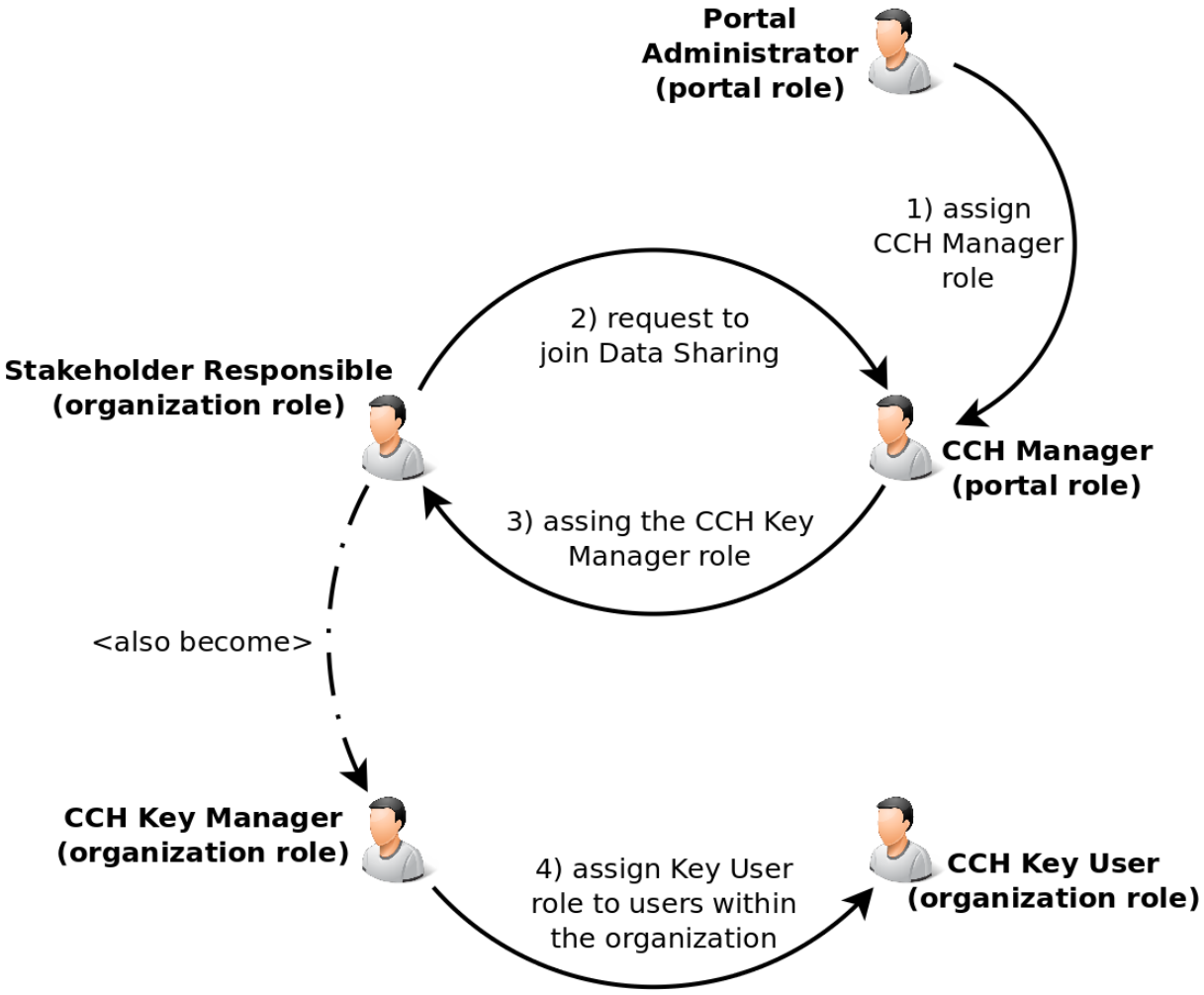

Figure 13: Key Assignment process

3.6.5. Verification Processes for network owners

The processes of the Community Portal required for managing new applicants are described in this section. If an organisation wants to join the Community Portal it an be verified by an already existing member, which takes the role of a "mentor", introducing the new organisation in the Community Portal. This organization can identify and introduce the new member and the CP Manager can enable the new account manually.
All these processes are triggered and run in the Community Portal, but in all cases the CCH is affected. The read keys of the stakeholders are associated here with the ASN and IP lists and are stored in the CCH databases.

3.6.5.1. Manual verification by the CCH-Manager

Until automation of these processes is implemented, manual supervision is needed to ensure that a stakeholder will only have access to IP or ASNs that are belonging to his organisation.

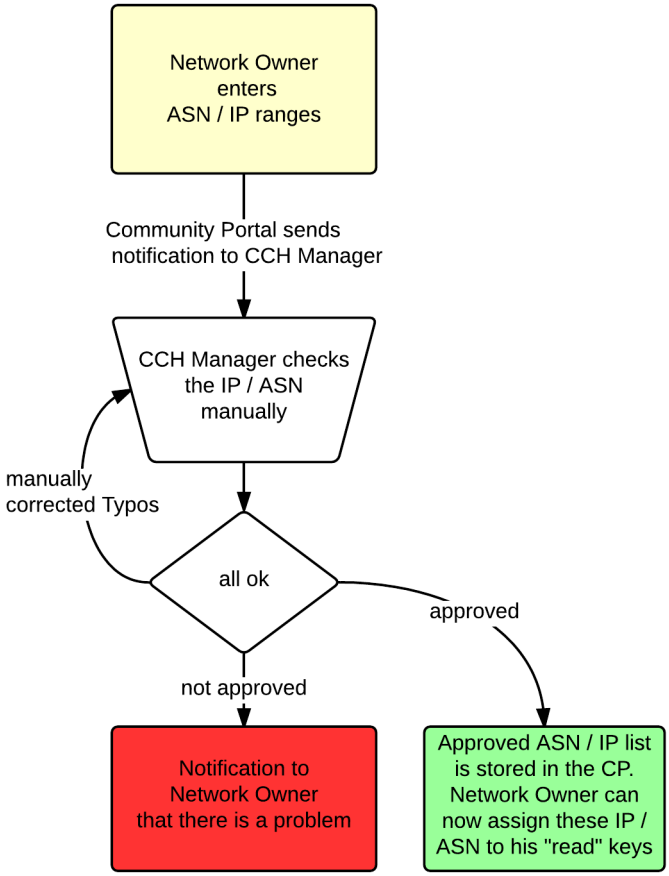

Figure 14: Manual approval of IP/ASN ranges of a network owner

3.6.5.2. Automated Check

For every given organisation, which runs their own networks or owns an IP address range, there has to be an abuse contact stated at the ripe database.
Therefore, the Community Portal can check at the signup of a new member for this abuse contact to verify the organisation and grant access to the Community Portal, and thus enabling an organisation role as CCH Key Manager.

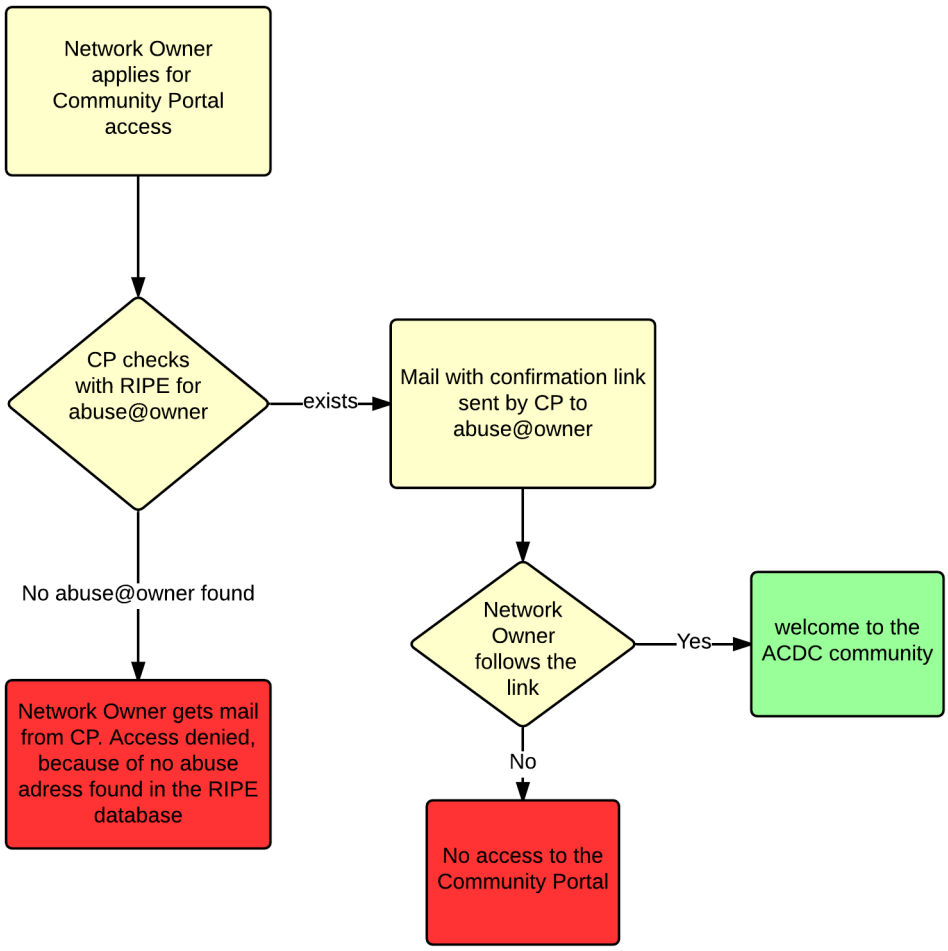

Figure 15: Automated Check for Abuse Contact

If the automated check fails, a CCH Manager can still contact the applicant to verify him manually or contact him to explain the need of an abuse contact.

3.6.5.3. ASN / IP verification of existing Community Portal members

After an organisation joined the Community Portal, they can create write and read API Keys to access the CCH. While a write key is not limited to ASN / IPs it must be ensured for read keys that only the IP or ASN belonging to the organization, will be accepted by the Community Portal.
As ASN and IP ranges can change over time, there is the need to edit them or to update the list of ANSs IP Ranges for a stakeholder. The Community Portal can periodically check for such changes and either give the stakeholder a message to update his read keys or update the read keys automatically in the CCH.

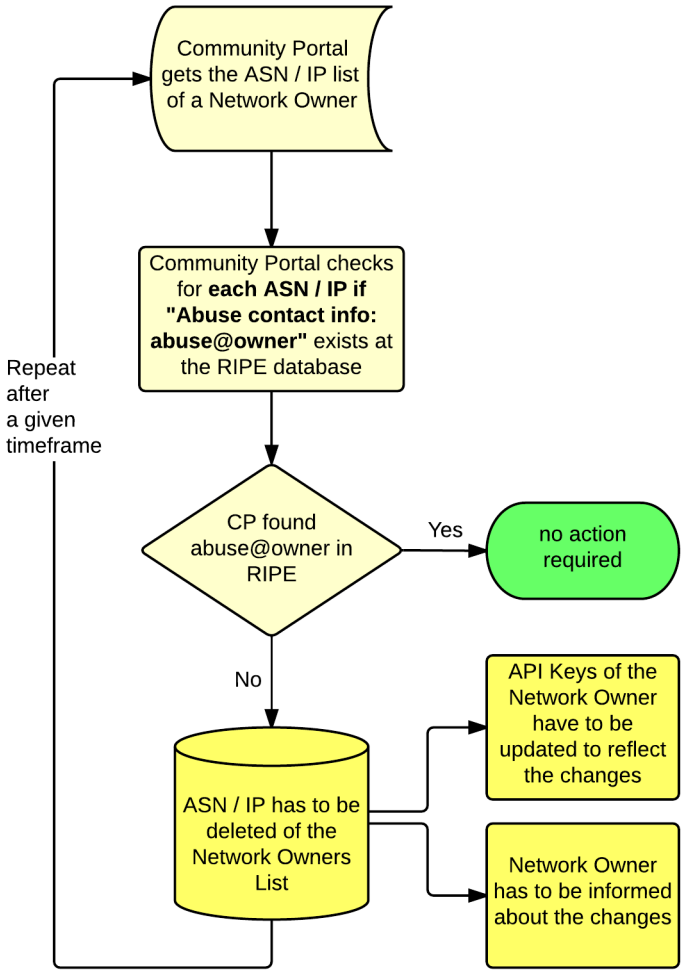

Figure 16: ASN and IP lists are periodically checked by the CP

3.7. Examples for API-Key Management

3.7.1. Create New API Key

CreateNewApiKey - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/api_keys/
Headers	Authorization: Token token="Community_Portal_Key"
Method	POST
Content Type	application/json
Query String	-
URL	https://HOSTNAME:3000/api/v2/api_keys/
Path Params	-
data : {.....}	{ "email":" email@community-portal.it ", "group_id":"[Groups]", "key_type":"[Key Types]", "description":"Testuser_Alpha" "data_schema_url": "http://www.schema.com/schema1" "super_user": [true/false], } Groups (group_id or group name, case sensitive!): /1 (or /Unverified) /2 (or /ISP) /3 (or /CERT) /4 (or /Antivirus) Key Types: - if "read" set key queries to max_queries of the group - if "write" set key queries to 0
Response	{ "id": 123, "key_type": "write", "group_id": "1", "ttl": 346, "description": "My Key Description", "email": "email@community-portal.it", "created_at": "2014-11-12T12:09:13.377Z", "updated_at": "2014-11-12T12:09:13.377Z", "data_schema_url": "http://www.schema.com/schema1" "access_token": "YOUR_API_KEY", "super_user", false } **see:** curl -XPOST -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/api_keys -d'{ "email":"email@community-portal.it", "group_id":1, "key_type":"write", "description":"Testuser_Alpha", "data_schema_url": "http://www.schema.com/schema1", "super_user":false }'

Table 1: CreateNewApiKey – Service

3.7.2. Update API-Key

UpdateApiKey - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/api_keys/
Headers	Authorization: Token token="Community_Portal_Key"
Method	PUT
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/api_keys/
Path Params	/ID_KEY
data : {.....}	{ "email":" email@community-portal.it ", "description":"My Key Description 1", "ttl": [if == 0 invalidate, if >0 valid] }
Response	{ "description":"My Key Description 1", "email": " email@community-portal.it ", "ttl": 0, "updated_at":"2014-12-15T12:09:13.377Z" } see: curl -XPUT -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/api_keys/147 -d'{ "email":"email2@community-portal.it", "description":"Testuser_Alpha 1"}'

Table 2: UpdateApiKey – Service

NOTE: ttl indicates the desired number of seconds the key is still valid from the time of this request. It is set to 0 to invalidate the key.
In any case, the new expiration time of the key (calculated as current time plus new ttl) cannot exceed the max expiration time of the key (calculated as key creation time plus max_ttl of the group the key belongs to). If exceeding, the new ttl will be lowered to that value.

3.7.3. Get API Key

GetApiKey Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/api_keys/
Headers	Authorization: Token token="Community_Portal_Key"

Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/api_keys/
Path Params	/ID_KEY
Response	JSON (sample): { "access_token":"YOUR_API_KEY", "id":123, "key_type" : "write", "group_id" : 1, "ttl": 0, description" : "My Key Description 1", "email": " email@community-portal.it ", "created_at":"2014-11-12T12:09:13.377Z", "updated_at":"2014-11-12T12:09:13.377Z", "data_schema": "http://www.schema.com/schema1" } see: curl -XGET -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/api_keys/147

Table 3: GetApiKey Service

3.7.4. Replace Key

ReplaceKey Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/api_keys/replace/
Headers	Authorization: Token token="Community_Portal_Key"
Method	POST
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/api_keys/replace/
Path Params	/ID_KEY
data: {.....}	-
Response	JSON (sample): { "access_token":"YOUR_API_KEY", "id":124, "key_type" : "write", "group_id" : 1, "ttl": 0, "description" : "My Key Description 1", "email": " email@community-portal.it ", "created_at":"2014-11-12T12:09:13.377Z", "updated_at":"2014-13-12T12:09:13.377Z", "data_schema": "http://www.schema.com/schema1" } see: curl -XPOST -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/api_keys/replace/147

Table 4: ReplaceKey Service

3.8. Data Access Management (DAM)

The authentication mechanism of the CCH combines the use of a solid state-of-the-art Identity Management System (IdMS), including a verification process to approve an applicant. For this reason Trusted Introducer (TI) and the Community Portal to centralize the authentication is very useful in this context. Therefore any ACDC components connecting to the CCH must support to work in this type of IdMS environment.

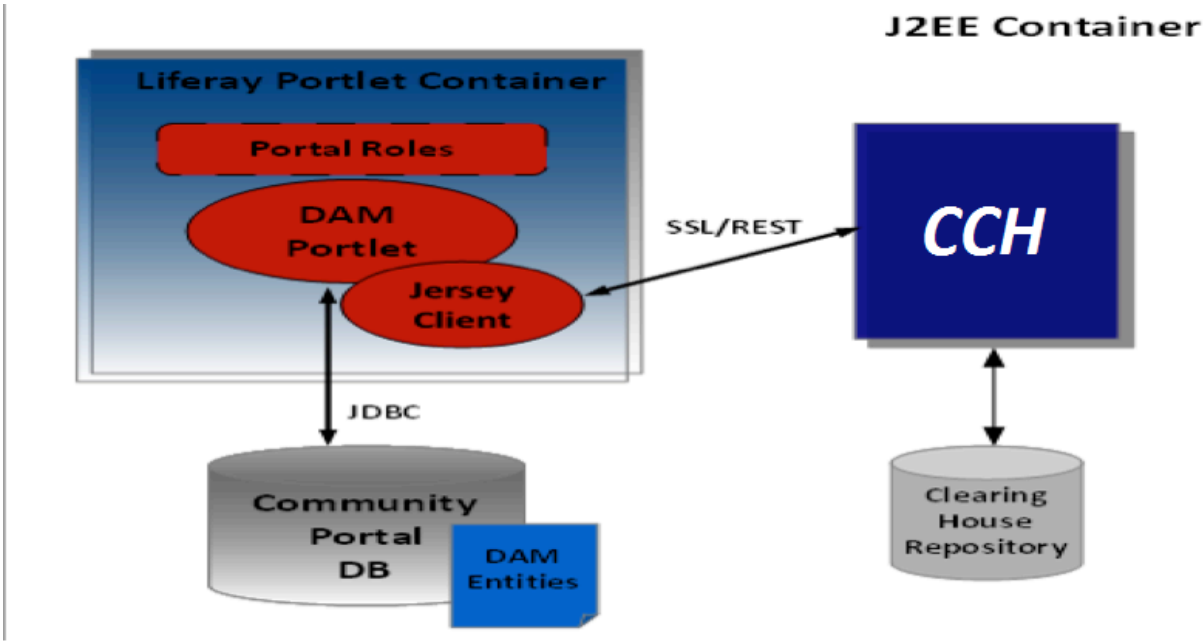

Figure 17: Community Portal and DAM

Restricted access will be handled via the Community Portal, where API keys are generated for every data feed request. The workflows for specific stakeholders are being defined within the work related to **D1.7.2 – Data Format Specification**. The different workflows and related restrictions are covered within chapter 10 of that document. The specifications will be implemented into the CCH. The draft of **D1.7.2** currently covers CERTs, ISPs and research. Other groups and settings will be added within further progress of the project and be based on real-time usage through internal and external ACDC stakeholders.
The *Data Access Management (DAM)* is owned and managed by the Community Portal. Further information on DAM can be retrieved from **D6.2.1 ACDC Social Platform** as part of chapter 8.4

3.8.1. Group Management

This service "group Management" retrieves the list of groups for the API-Keys. Each API-Key belongs to only one group, which imposes constraints on the usage of the key itself. Each group has an ID, identifying the group and its label. Currently, four groups have been foreseen for interaction with the CCH: Unverified, ISP, CERT, Antivirus.
Each organization using CCH services is associated to a group in the DAM, and API-Keys are associated to groups in the CCH.
Current constraints on the key usage associated to each group are:

- max_ttl – the maximum time to live the key can have, when expired the key must be replaced. This constraint is taken into account by the DAM when sending key creation requests to the CCH, and cross checked by the CCH on received requests. Max_ttl is set in seconds, if 0, then the Key in invalidated.
- max_queries – the maximum number of "read" requests that can be performed using the key. Some groups like Antivirus have the possibility to re-increment the number by

performing "write queries" with other keys of their organizations. This constraint is enforced by the CCH on each "read" request.

- max_keys – The maximum number of active (i.e. not expired) keys the organization can create.
- data_types – The type of data that can be retrieved by using the key.

Key Management is handled by the Community Portal. We differentiate between “”super-user” Keys and “normal” Keys. A “super-user” key is a role key that can create other Keys and is not restricted to only read or write, but can do both, read and write.
“Super-user” Keys are used for the Community Portal in its role to Manage Keys and Groups and are used for each organisation to manage the organisational Keys.
A “Super-user” Key is not distributed to the organisation but is stored in the Community Portal and only acts for managing the user keys of the affected organisation.

3.8.2. Examples Group Management

3.8.2.1. Get Groups

GetGroups - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/groups/
Headers	Authorization: Token token="Community_Portal_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/groups/
Path Params	-
Response	{"groups": [1, 2, 3, 4]} see: curl -XGET -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' https://HOSTNAME:3000/api/v2/groups

Table 5: GetGroups Service

3.8.2.2. GetGroups By ID – Service

GetGroups By ID - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/groups/
Headers	Authorization: Token token="Community_Portal_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/groups/
Path Params	Group id (or group name, case sensitive!): /1 (or /Unverified) /2 (or /ISP) /3 (or /CERT) /4 (or /Antivirus)
Response	{ "group_id": 1, "label": "Unverified", "max_ttl":0, "max_queries": 10, "max_keys": 10, "data_types": [{ "data_type_id": 1, "label": "IP Address" }] } see: curl -XGET -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/groups/1

Table 6: GetGroups by ID

3.8.2.3. Update Group – Service

This service updates the parameters associated to a given group. Only the set of parameters included in the PUT request will be updated for the group, other parameters will be kept as they are. Usage of this service must be restricted to super-users of the CCH only. The table below contains parameters for the implementation of the service.

UpdateGroup - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/

Service Name	/groups/
Headers	Authorization: Token token="Community_Portal_Key"
Method	PUT
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/groups/ISP
Path Params	Group id (or group name, case sensitive!): /1 (or /Unverified) /2 (or /ISP) /3 (or /CERT) /4 (or /Antivirus) ...
data : {.....}	JSON: { "max_ttl": 0, "max_queries": 50, "max_keys": 10, }
Response	JSON: { "label": "ISP", "max_ttl": 0, “max_queries": 50, "max_keys": 10, "data_types": "[{"data_type_id": 1, "label": "IP Address"}, ...]" } see: curl -XPUT -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' –k https://HOSTNAME:3000/api/v2/groups/Unverified -d '{"max_queries": 50}'

Table 7: Update Group Service

3.8.2.4. CreateGroup - Service

This service creates a new group. All parameters for the group are mandatory. The CCH returns error in case a group with the same label already exists. Upon creation, CCH returns the ID of the new group created, and parameters of the new group. Usage of this service must be restricted to CCH-Key-User of the CCH only. The table below contains parameters for the implementation of the service.

CreateGroup - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/groups/
Headers	Authorization: Token token="Community_Portal_Key"
Method	POST
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/groups/ISP
Path Params	-
data : {.....}	{

	"label": "New Group", "max_ttl": 0, "max_queries": 10, "max_keys": 10, "data_types": [1,2,3] <- no brackets arround an array!! }
Response	{ "group_id": 5, "label": "New Group", "max_ttl": 0, "max_queries": 10, "max_keys": 10, "data_types": "[{"data_type_id": 1, "label": "IP Address"}, ...]" } see: curl -XPOST -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/groups -d '{ "label": "New Group", "max_ttl": 0, "max_queries": 10, "max_keys": 10, "data_types": [1,2,3] }'

Table 8: CreateGroup - Service

3.8.2.5. Delete Group

This service deletes an existing group. When a group is deleted, all the API-Keys associated to that group in the CCH will be associated to the unverified group, which cannot be deleted or modified. Usage of this service will be restricted to CCH Key-user of the CCH only. The table below contains parameters for the implementation of the service.

DeleteGroup - Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/groups/
Headers	Authorization: Token token="Community_Portal_Key"
Method	DELETE
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/groups/ISP
Path Params	Group id (or group name, case sensitive!): /2 (or /ISP) /3 (or /CERT) /4 (or /Antivirus) ...
Response	JSON: { "result":"[true|false]" } see:

```
curl -XDELETE -H 'Authorization: Token token="Community_Portal_Key"' -H
'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/groups/6
```

Table 9: DeleteGroup – Service

3.8.3. Data Sharing Policies

The “Sharing Policies” module of the Data Access Management (DAM) in the Community Portal will use sharing policies management services exposed by the CCH. These services enable the management of data sharing policies between organizations through the CCH services.
API Keys can be created in the Community Portal, we distinguish between “read” Keys, which can only receive data from the CCH and “write” keys, which can only send data to the CCH and check the CCH for their own data contributions of the last 15 minutes.

Data sharing is done via an XMPP server, where the CCH will send allowed datasets to different XMPP channels.
Every “read” key has it´s own channel, where the data that the key is allowed to see is streamed.
Access to the XMPP Channels are restricted to sign in with the unique API_Key_ID and the API_Key_ID_Number of an API key.

We have 2 sets of sharing policies:

a) sharing by ASN / IP Ranges
   Some companies have IP or ASN Ranges that they own and thus they are allowed to see every incident that falls within these IP or ASN Ranges. So a member organization can setup their ASN and IP ranges in the Community Portal, these are validated and approved by the Portal and once they are approved, these ASN and IP ranges can be applied to a “read” key. Every incident that is submitted to the CCH and falls within this given range will be automatically streamed into the associated “read” keys channel, informing the owner of a new incident.
   Example:
   CERTS are responsible to monitor a given ASN range or IP Range.
   ISPs have their own ASN and IP Ranges; they are legally allowed to see all incidents in the CCH.

This first policy is automatically performed in the CCH.

b) Organisations can establish a mutual exchange, sharing data between a “read” and a ”write” key.
   This is a one to one relationship, a contract between two parties in which the CCH occurs only as a data switch.
   A “read” key asks to be linked with another “write” key and once this is approved by the owner of the “write” key, all data that the “write” key submits to the CCH will be streamed into the “read” keys XMPP channel, regardless of any ASN or IP settings which may apply to the “read” key.
   “read” / “write” sharing policies can be requested in both ways – by the owner of the “read” key, which asks for data and by the owner of a “write” key, who wants to offer data to another party.

3.8.3.1. Add Sharing Policy

This service adds a new sharing policy between two API-Keys. One key has read-only access, while the other has both (read/write). Both keys can be used to setup the sharing policy. The CCH service will detect the channel direction based on the key_type properties of the two keys identifiers.
The table below contains parameters for the implementation of the service.

AddSharingPolicy Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/sharing_policies/
Headers	Authorization: Token token="Community_Portal_Key"
Method	POST
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/sharing_policies/
Path Params	/{int: ID_KEY}
data: {.....}	JSON (sample): { "other_key_id": 234 }
Response	JSON (sample): { "result":"[true|false]" } see: curl -XPOST -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/sharing_policies/147 -d '{"other_key_id":145}'

Table 10: AddSharingPolicy Service

3.8.3.2. Delete Sharing Policy

This service deletes an existing sharing policy. The request can be sent by two keys, one key has read-only access, while the other has both (read/write). In both cases the effect is the immediate stop of the data flow between the read and the write keys.
The table below contains parameters for the implementation of the service.

DeleteSharingPolicy Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/sharing_policies/
Headers	Authorization: Token token="Community_Portal_Key"
Method	DELETE
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/sharing_policies/
Path Params	/{int: ID_KEY}
data: {.....}	JSON (sample): { "other_key_id": 234 }
Response	JSON (sample): { "result":"[true|false]" } see: curl -XDELETE -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/sharing_policies/147 -d '{"other_key_id":145}'

Table 11: DeleteSharingPolicy Service

3.8.3.3. Get Sharing Policies

This service allows the user to retrieve the list of sharing policies attached to the keys.
The table below contains parameters for the implementation of the service.

GetSharingPolicy Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/sharing_policies/
Headers	Authorization: Token token="Community_Portal_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/sharing_policies/
Path Params	-
data: {.....}	JSON (sample): { "email": " email@community-portal.it " }
Response	JSON (sample): { "channels": [{"myKey": 255, "stackholders_key" : 322}, {"myKey": 266, "stackholders_key" : 456}, {"myKey": 278, "stackholders_key" : 521},] } see: curl -XGET -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/sharing_policies/ -d '{"email":" email@community-portal.it "}'

Table 12: GetSharing Policy - Service

3.9. Data & Data Formats

The definition and specification on data formats and their retrieval and standards is part of deliverable **D1.7.2 Data Formats.** The lecture of this document is recommended as an addition to this chapter.

1.1. Schemata

As part of the pre-work and assessment by the project partners, it has been determined that a basic standardisation of the submitted data fields and basic requirement on mandatory fields simplifies the data submission and retrieval. These specifications have been defined as the ACDC - "Schemata". These have been outlined and defined in the Deliverables **D1.7.1/2 Data Formats Specification**. Schemata are being worked out in **D1.7.2 Data Formats** and will be managed by the Community Portal. The Community Portal provides storage for the different schemata and will distribute them to every interested ACDC Partner according to the sharing policies that define if the partner is legible to receive the data..
The current process to submit a schemata proposal is handled manually. A partner involved in WP2 makes a suggestion for schemata, which gets verified and approved by the partners of WP1. The schema is then stored in the Community Portal and implemented by the CCH for further use.
The ACDC roadmap includes an automated schemata upload through the Community Portal, which includes a verification and automated adaption into the CCH.

3.9.1. Minimal Dataset

The CCH accepts a basic set of commands with which external sources can report security incidents.

ACDC Minimal dataset		
This is the minimal schema a Dataset submitted to or received from ACDC's CCH must conform to.		
Required fields		
report_category	string	The category of the report. This links the report to one of ACDC's schemata. A report category has the format 'eu.acdc.<identifier>'.
report_type	string	The type of the report. This is a free text field characterising the report that should be used for a human readable description rather than for automatic processing. As a rule of thumb this should not be longer than one sentence.
timestamp	string format: date-time	The timestamp when the reported observation took place. This can for example be when an attack occurred, when a malware hosting was observed, or when a compromise took place according to log files.
source_key	string enum: ip, malware, subject, uri	The type of the reported object.
source_value	string	The identifier of the reported object like its IP address or URI.
confidence_level	number minimum: 0.0 maximum: 1.0	The level of confidence put into the accuracy of the report. A number between 0.0 and 1.0 with 0.0 being unreliable and 1.0 being verified to be accurate.
version	integer enum: 1	The version number of the data format used for the report.

Optional fields		
report_id	integer	The ID of the report in the CCH. This will be set by the CCH and is thus overwritten on import.
reported_at	string format: date-time	The timestamp when the report was submitted to the CCH. This will be set by the CCH and is thus overwritten on import.
botnet	string	The botnet the observation is attributed to. This can for example be the botnet a malware joins, the botnet that sends a spam campaign, or the botnet a bot belongs to.

Figure 18 : The ACDC Minimal Dataset

Additional requirements and advanced schemes have been defined by WP1. It is documented as the "Schemes Proposal" in **Deliverable 1.7.2 Data Format** in chapter 6 and following.

Example for the Minimal Dataset:

ACDC Minimal dataset	
"report_category":"eu.acdc.minimal",	ADCD Minimal Dataset
"report_type":"[Experiment] [ECO][Test] Minimal",	[Experiment Name] [Company] [Sensor] description
"timestamp":"2015-05-22T14:44:33Z",	Timestamp date-time
"source_key":"uri",	Type of reported object – URI / IP / malware / subject
"source_value":"http://example-url.de",	Identifier of reported object – domain example-url.de
"confidence_level":0.11,	Accuracy of report 0 lowest 1 highest
"version":1	Version number of data format

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{"report_category":"eu.acdc.minimal", "report_type":"[Experiment] [ECO][Test] Minimal", "timestamp":"2015-05-22T14:44:33Z", "source_key":"uri", "source_value":"http://example-url.de", "confidence_level":0.11, "version":1}'

output:
{"confidence_level":0.11,"report_category":"eu.acdc.minimal","report_type":"[Experiment] [ECO][Test] Minimal","reported_at":"2015-05-22T14:03:24Z","source_key":"uri","source_value":"http://example-url.de","timestamp":"2015-05-22T14:44:33Z","version":1,"report_id":"545b7fac7765625862050000"}

3.9.2. eu.acdc.attack

Attack – eu.acdc.attack	
"report_category": "eu.acdc.attack"	The category of the report.
"report_type": "[ATTACK] [ECO][Test] TCP SYN Flood by eco"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "ip"	Type of reported object – URI / IP / malware / subject
"source_value": "192.0.0.1"	The IP of the system performing the attack.
"confidence_level": 0.11	Accuracy of report – 0 lowest 1 highest
"version": 1	Version number of data format
"report_subcategory": "dos"	The type of attack performed. string enum: abuse, compromise, data, dos, dos.dns, dos.http, dos.tcp, dos.udp, login, malware,
"ip_protocol_number": 6	
"ip_version": 4	The IP version of the attack connection
"src_ip_v4": "192.0.0.1"	Dependency – IPv4
"src_mode": "plain"	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.

Example for a eu.acdc.attack – minmal Dataset one has to provide.

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.attack", "report_type": "[ATTACK] [ECO][Test] TCP SYN Flood by eco", "timestamp": "2015-05-22T13:33:33Z", "source_key": "ip", "source_value": "192.0.0.1", "confidence_level": 0.11, "version": 1, "report_subcategory": "dos","ip_protocol_number": 6, "ip_version": 4, "src_ip_v4": "192.0.0.1", "src_mode": "plain"}'

output:
{"confidence_level":0.11,"ip_protocol_number":6,"ip_version":4,"report_category":"eu.acdc.attack", "report_subcategory":"dos","report_type":"[ATTACK] [ECO][Test] TCP SYN Flood by eco","reported_at":"2014-11-07T16:12:20Z","source_key":"ip","source_value":"192.0.0.1","src_ip_v4":"192.0.0.1","src_mode":"plain","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545cef647765625852100300"}

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.attack", "report_subcategory": "dos", "report_type": "[Experiment] [ECO][Test] TCP SYN Flood by eco", "timestamp": "2015-05-22T13:33:33Z", "source_key": "ip", "source_value": "192.0.0.1", "ip_protocol_number": 6, "ip_version": 4, "src_ip_v4": "192.0.0.1", "src_mode": "plain", "dst_ip_v4": "192.0.0.111", "dst_mode": "anon", "dst_port": 80, "confidence_level": 0.11, "version": 1 }'

output:

{"confidence_level":0.11,"dst_ip_v4":"192.0.0.111","dst_mode":"anon","dst_port":80,"ip_protocol_number":6,"ip_version":4,"report_category":"eu.acdc.attack","report_subcategory":"dos","report_type":"[Experiment] [ECO][Test] TCP SYN Flood by eco","reported_at":"2015-05-22T13:44:36Z","source_key":"ip","source_value":"192.0.0.1","src_ip_v4":"192.0.0.1","src_mode":"plain","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7b447765625852030000"}

The Minimal Dataset is not intended to be used in productive phase. It was introduced to quickly test data submissions and to give new stakeholders a dataset to test with and to adopt their sensors to the JSON schemata the CCH accepts.

3.9.3. eu.acdc.bot

Bot – eu.acdc.bot	
"report_category": "eu.acdc.bot"	The category of the report.
"report_type": "[Experiment] [ECO][Test] Connection to Zeus C2 powered by eco"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "ip"	Type of reported object – URI / IP / malware / subject
"source_value": "121.154.32.23"	The IP of the system performing the attack.
"confidence_level": 1.0	Accuracy of report
"version": 1	Version number of data format
"report_subcategory": "other"	The type of bot: fast_flux, other
"ip_version": 4	The IP version of the C2 connection.
"src_ip_v4": "121.154.32.23"	Dependency – IPv4
"src_mode": "plain"	The mode of the source IP. This can be plain for unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.
"c2_ip_v4": "10.1.3.12"	The IPv4 of the C2 server.
"c2_mode": "anon"	The mode of the C2 server IP. This can be plain for unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.

a) Bot

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.bot", "report_type": "[Experiment] [ECO][Test] Connection to Zeus C2, powered by eco", "timestamp": "2015-05-22T13:33:33Z", "source_key": "ip", "source_value": "121.154.32.23", "confidence_level": 1.0, "version": 1, "report_subcategory": "other", "ip_version": 4, "src_ip_v4": "121.154.32.23", "src_mode": "plain", "c2_ip_v4": "10.1.3.12", "c2_mode": "anon" }'

output:

{"c2_ip_v4":"10.1.3.12","c2_mode":"anon","confidence_level":1.0,"ip_version":4,"report_category":"eu.acdc.bot","report_subcategory":"other","report_type":"[Experiment] [ECO][Test] Connection to Zeus C2, powered by eco","reported_at":"2015-05-22T13:45:09Z","source_key":"ip","source_value":"121.154.32.23","src_ip_v4":"121.154.32.23","src_mode":"plain","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7b657765625852040000"}

b) Fast flux bot

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.bot", "report_type": "[Experiment] [ECO][Test] Fast flux XYZ bot, powered by eco", "timestamp": "2015-05-22T13:33:33Z", "source_key": "ip", "source_value": "121.154.32.23", "confidence_level": 1.0, "version": 1, "report_subcategory": "fast_flux", "ip_version": 4, "src_ip_v4": "121.154.32.23", "src_mode": "plain", "fast_flux_uri": "dns:fast_flux.example.com" }'

output:

{"confidence_level":1.0,"fast_flux_uri":"dns:fast_flux.example.com","ip_version":4,"report_category":"eu.acdc.bot","report_subcategory":"fast_flux","report_type":"[Experiment] [ECO][Test] Fast flux XYZ bot, powered by eco","reported_at":"2015-05-22T13:45:33Z","source_key":"ip","source_value":"121.154.32.23","src_ip_v4":"121.154.32.23","src_mode":"plain","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7b7d7765625852050000"}

3.9.4. eu.acdc.botnet

Botnet - eu.acdc.botnet	
"report_category": "eu.acdc.botnet"	The category of the report.
"report_type": "[Experiment] [ECO][Test] ZeuS botnet 42 provieded by eco"	[Experiment Name] [Company] [Sensor] description
"source_key": "botnet"	The type of the reported object: a botnet.
"source_value": "ZeuS-42"	The identifier of the botnet. This can be the name of a single type of botnet or a combination of a botnet type and an identifier for a specific instance of the botnet.
"version": 1	The version number of the data format used for the report.
"report_subcategory": "p2p"	The category of the botnet. c2, p2p, other

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.botnet", "report_type": "[Experiment] [ECO][Test] ZeuS botnet 42 provieded by eco", "source_key": "botnet", "source_value": "ZeuS-42", "version": 1, "report_subcategory": "p2p" }'

output:

{"report_category":"eu.acdc.botnet","report_subcategory":"p2p","report_type":"[Experiment] [ECO][Test] ZeuS botnet 42 provieded by eco","reported_at":"2015-05-22T13:46:06Z","source_key":"botnet","source_value":"ZeuS-42","version":1,"report_id":"545b7b9e7765625852060000"}

3.9.5. eu.acdc.c2_server

C2-Server - eu.acdc.c2_server	
"report_category": "eu.acdc.c2_server"	The category of the report.
"report_type": "[Experiment] [ECO][Test] Zeus C2 provided by eco"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "ip"	The type of the reported object: IP address.
"source_value": "121.154.32.23"	The IP address of the C2 server.
"confidence_level": 1.0	
"version": 1	The version number of the data format used for the report.
"report_subcategory": "http"	The control channel used by the C2. http, irc, other
"ip_version": 4	The IP version of the C2 server's IP address. Int 4,6
"c2_ip_v4": "121.154.32.23"	The IPv4 of the C2 server.
"c2_mode": "plain"	The mode of the C2 server IP. This can be plain for unaltered IPs, anon for anonymised IPs, or pseudo for pseudonymised IPs.

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.c2_server", "report_type": "[Experiment] [ECO][Test] Zeus C2 provided by eco", "timestamp": "2015-05-22T13:33:33Z", "source_key": "ip", "source_value": "121.154.32.23", "confidence_level": 1.0, "version": 1, "report_subcategory": "http", "ip_version": 4, "c2_ip_v4": "121.154.32.23", "c2_mode": "plain" }'

output:

{"c2_ip_v4":"121.154.32.23","c2_mode":"plain","confidence_level":1.0,"ip_version":4,"report_category":"eu.acdc.c2_server","report_subcategory":"http","report_type":"[Experiment] [ECO][Test] Zeus C2 provided by eco","reported_at":"2015-05-22T13:46:29Z","source_key":"ip","source_value":"121.154.32.23","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7bb57765625852070000"}

3.9.6. eu.acdc.fast_flux

FastFlux Domains - eu.acdc.fast_flux	
"report_category": "eu.acdc.fast_flux"	The category of the report
"report_type": "[Experiment] [ECO][Test] Fast flux domain of the XYZ-eco bot"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "uri"	The type of the reported object: a domain URI.
"source_value": "dns:fast_flux.example.com"	The fast flux domain URI.
"confidence_level": 1.0	The level of confidence put into the accuracy of the report.
"version": 1	The version number of the data format used for the report.

a) Fast flux domain of the XYZ bot

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.fast_flux", "report_type": "[Experiment] [ECO][Test] Fast flux domain of the XYZ-eco bot", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "dns:fast_flux.example.com", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.fast_flux","report_type":"[Experiment] [ECO][Test] Fast flux domain of the XYZ-eco bot","reported_at":"2015-05-22T13:46:55Z","source_key":"uri","source_value":"dns:fast_flux.example.com","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7bcf7765625852080000"}

b) Domain example.ru was detected as fast-flux

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.fast_flux", "report_type": "[Experiment] [ECO][Test] Domain example.ru was detected as fast-fluxâ€¦", "source_key": "uri", "source_value": "dns:example.ru", "timestamp": "2015-05-22T13:33:33Z", "duration": 10848, "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"duration":10848,"report_category":"eu.acdc.fast_flux","report_type":"[Experiment] [ECO][Test] Domain example.ru was detected as fast-fluxâ€¦","reported_at":"2015-05-22T13:47:24Z","source_key":"uri","source_value":"dns:example.ru","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7bec7765625852090000"}

3.9.7. eu.acdc.malicious_uri

Malicious URI - eu.acdc.malicious_uri	
"report_category": "eu.acdc.malicious_uri"	The category of the report
"report_subcategory": "malware"	The type of the malicious content at the URI. **exploit, malware, phishing, other**
"report_type": "[Websites] [ECO][Test] A URI detected doing malicious activities"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "uri"	The type of the reported object: a domain URI.
"source_value": "http://example-url.de/ohnoes.exe"	The URI to the malicious content.
"confidence_level": 1.0	The level of confidence put into the accuracy of the report.
"version": 1	The version number of the data format used for the report.

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malicious_uri", "report_subcategory": "malware", "report_type": "[Websites] [ECO][Test] A URI detected doing malicious activities", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "http://example-url.de/ohnoes.exe", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malicious_uri","report_subcategory":"malware", "report_type":"[Websites] [ECO][Test] A URI detected doing malicious activities","reported_at":"2015-05-22T13:47:53Z","source_key":"uri","source_value":"http://example-url.de/ohnoes.exe","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7c097765625862020000"}

Other Examples for Malicious URI - eu.acdc.malicious_uri

a) Malware

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malicious_uri", "report_subcategory": "malware", "report_type": "[Websites] [ECO][Test] Malware was found..", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "http://example-url.de/5nW.exe", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malicious_uri","report_subcategory":"malware", "report_type":"[Websites] [ECO][Test] Malware was found..","reported_at":"2015-05-22T13:48:16Z","source_key":"uri","source_value":"http://example-url.de/5nW.exe","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7c2077656258520a0000"}

b) Phishing

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malicious_uri", "report_subcategory": "phishing", "report_type": "[Websites] [ECO][Test] URI of phishing site", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "http://example-url.de/language/index.html", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malicious_uri","report_subcategory":"phishing", "report_type":"[Websites] [ECO][Test] URI of phishing site","reported_at":"2015-05-22T13:48:43Z","source_key":"uri","source_value":"http://example-url.de/language/index.html","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7c3b77656258520b0000"}

c) Exploit

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malicious_uri", "report_subcategory": "exploit", "report_type": "[Websites] [ECO][Test] Exploiting Fuchs des Feuers", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "http://example-url.de/malicious.html", "confidence_level": 1.0, "version": 1, "exploits": [{ "type": "cve", "value": "eco-2014" }]}'

output:

{"confidence_level":1.0,"exploits":[{"type":"cve","value":"eco-2014"}],"report_category":"eu.acdc.malicious_uri","report_subcategory":"exploit","report_type":"[Websites] [ECO][Test] Exploiting Fuchs des Feuers","reported_at":"2015-05-22T13:49:22Z","source_key":"uri","source_value":"http://example-url.de/malicious.html","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7c627765625862030000"}

3.9.8. eu.acdc.malware

Malware Sample - eu.acdc.malware	
"report_category": "eu.acdc.malware"	The category of the report
"report_type": "[Experiment] [ECO][Test] Bot from honeypot capture"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	Timestamp date-time
"source_key": "malware"	The type of the reported object: a malware sample.
"source_value": "933ff380eb1448c5c583796fdc6ce842eec14a23b686fff4672c85a7ee9d7d0a"	The SHA256 hash of the malware sample.
"sample_b64": "ewogICAgInRpdGxlIjogIk1hbHdhcmUgU2FtcGxlIiwgCiAgICAiZGVzY3JpcHRpb24iOiAK..."	The source code of the sample encoded in Base64.
"confidence_level": 1.0	The level of confidence put into the accuracy of the report.
"version": 1	The version number of the data format used for the report.

a) Bot from honeypot capture

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malware", "report_type": "[Experiment] [ECO][Test] Bot from honeypot capture", "timestamp": "2015-05-22T13:33:33Z", "source_key": "malware", "source_value": "933ff380eb1448c5c583796fdc6ce842eec14a23b686fff4672c85a7ee9d7d0a", "sample_b64": "ewogICAgInRpdGxlIjogIk1hbHdhcmUgU2FtcGxlIiwgCiAgICAiZGVzY3JpcHRpb24iOiAK...", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malware","report_type":"[Experiment] [ECO][Test] Bot from honeypot capture","reported_at":"2015-05-22T14:41:44Z","sample_b64":"ewogICAgInRpdGxlIjogIk1hbHdhcmUgU2FtcGxlIiwgCiAgICAiZGVzY3JpcHRpb24iOiAK...","source_key":"malware","source_value":"933ff380eb1448c5c583796fdc6ce842eec14a23b686fff4672c85a7ee9d7d0a","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b88a87765625862070000"}

b) Malware sample was found in spam

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malware", "report_type": "[Experiment] [ECO][Test] Malware sample was found in spam.", "source_key": "malware", "source_value": "902fe4a680a1b42cdba57c551b32c13b", "sample_b64": "e32a5fa1...", "timestamp": "2015-05-22T13:33:33Z", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malware","report_type":"[Experiment] [ECO][Test] Malware sample was found in spam.","reported_at":"2015-05-22T14:46:27Z","sample_b64":"e32a5fa1...","source_key":"malware","source_value":"902fe4a680a1b42cdba57c551b32c13b","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b89c37765625862080000"}

c) Malware received by honeypot.

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.malware", "report_type": "[Experiment] [ECO][Test] Malware received by honeypot.", "timestamp": "2015-05-22T13:33:33Z", "source_key": "malware", "source_value": "902fe4a680a1b42cdba57c551b32c13b", "sample_b64": "e32a5fa1...", "confidence_level": 1.0, "version": 1 }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.malware","report_type":"[Experiment] [ECO][Test] Malware received by honeypot.","reported_at":"2015-05-22T14:47:02Z","sample_b64":"e32a5fa1...","source_key":"malware","source_value":"902fe4a680a1b42cdba57c551b32c13b","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b89e67765625852100000"}

3.9.9. eu.acdc.spam_campaign

Campaign – eu.acdc.spam_campaign	
"report_category": "eu.acdc.spam_campaign"	The category of the report
"report_type": " [Experiment] [ECO][Test] Campaign Spam Case occurred.."	[Experiment Name] [Company] [Sensor] description
"source_key": "subject"	The type of the reported object: an email subject.
"source_value": "All your base belongs to us"	The common subject of the spam campaign.
"timestamp": "2015-05-22T13:33:33Z"	The timestamp when the spam campaign was first observed.
"sample_sha256": "4d410bc194c5fbdf20d15fae6c6bd807f66b56c36afe3a3def37e4369193ed2e"	The SHA256 hash of the malware distributed with this campaign. This references a separate eu.acdc.malware report
"confidence_level": 1.0	
"version": 1	The version number of the data format used for the report

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.spam_campaign", "report_type": " [Experiment] [ECO][Test] Campaign €ECO WAS HERE€™ occurred..", "source_key": "subject", "source_value": "All your base belongs to us", "timestamp": "2015-05-22T13:33:33Z", "sample_sha256": "4d410bc194c5fbdf20d15fae6c6bd807f66b56c36afe3a3def37e4369193ed2e", "confidence_level": 1.0, "version": 1}'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.spam_campaign","report_type":" [Experiment] [ECO][Test] Campaign Spam Case occurred..","reported_at":"2015-05-22T13:50:51Z","sample_sha256":"4d410bc194c5fbdf20d15fae6c6bd807f66b56c36afe3a3def37e4369193ed2e","source_key":"subject","source_value":"All your base belongs to us","timestamp":"2015-05-22T13:33:33Z","version":1,"report_id":"545b7cbb7765625862040000"}

3.9.10. eu.acdc.vulnerable_uri

Vulnerable URI – eu.acdc.vulnerable_uri	
"report_category": "eu.acdc.vulnerable_uri"	The category of the report: a vulnerable URI.
"report_type": "[Experiment] [ECO][Test] SQL injection in search field"	[Experiment Name] [Company] [Sensor] description
"timestamp": "2015-05-22T13:33:33Z"	The timestamp when the vulnerable URI was observed.
"source_key": "uri"	The type of the reported object: a URI
"source_value": "http://example-url.de/search"	The URI to the vulnerable resource.
"confidence_level": 1.0	The level of confidence put into the accuracy of the report.
"version": 1	The version number of the data format used for the report.
"vulnerabilities": [{ "type": "cve", "value": "eco-2014" }] }'	An array of objects describing vulnerabilities discovered at the vulnerable URI. Array items: object (identifier scheme, vulnerability identifier)

curl -v -k -XPOST -H 'Authorization: Token token="Your_API_Key"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports -d '{ "report_category": "eu.acdc.vulnerable_uri", "report_type": "[Experiment] [ECO][Test] SQL injection in search field", "timestamp": "2015-05-22T13:33:33Z", "source_key": "uri", "source_value": "http://example-url.de/search", "confidence_level": 1.0, "version": 1, "vulnerabilities": [{ "type": "cve", "value": "eco-2014" }] }'

output:

{"confidence_level":1.0,"report_category":"eu.acdc.vulnerable_uri","report_type":"[Experiment] [ECO][Test] SQL injection in search field","reported_at":"2015-05-22T13:51:39Z","source_key":"uri","source_value":"http://example-url.de/search","timestamp":"2015-05-22T13:33:33Z","version":1,"vulnerabilities":[{"type":"cve","value":"eco-2014"}],"report_id":"545b7ceb77656258520c0000"}

3.10. Real-Time Access / Channels

The following Figure 19 illustrates the concept of the data flow to an authorised stakeholder from the Community Portal, the assignments of policies based on his permissions and interest, into the CCH and the channels he is authorised to receive his data.

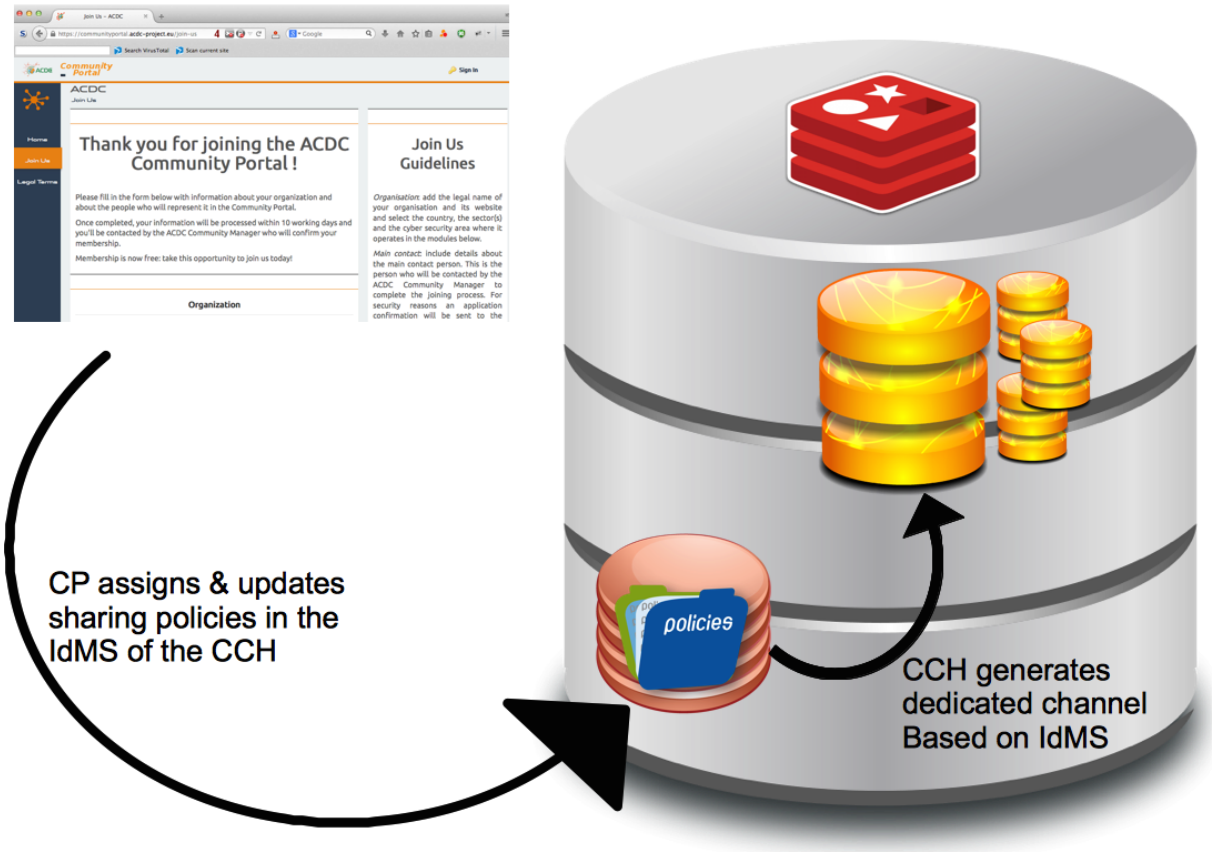

Figure 19: Channels for Real-Time Access

3.10.1. Get All Data Channels

This service retrieves the list of all channels (read or write) for data sharing; each channel corresponds to an API-Key.
The operation returns all the channels (other API-keys) that can be associated to the API-Key indicated in the request path. The returned key depends on the type of the key indicated:

- If the API-key is a read key, then all the Write-Keys IDs are returned
- If the API-key is a write key, then all the Read-Keys IDs are returned

The authorization token of a CCH Master-Key must be provided to perform this operation.
The table below contains parameters for the implementation of the service.

GetAllDataChannels Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/channels/
Headers	Authorization: Token token="Community_Portal_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/channels/
Path Params	/ID_KEY (read or write key)
data: {.....}	-
Response	JSON (sample): { "keys": [{"id": 255, "email": "email 1", "description": "Test User 1"}, {"id": 324, "email": "email 2", "description": "Test User 2"}, {"id": 13, "email": "email 3", "description": "Test User 3"}] } see: curl -XGET -H 'Authorization: Token token="Community_Portal_Key"' -H 'Content-Type: application/json' -k https://HOSTNAME:3000/api/v2/channels/145

Table 13: GetAllDataChannels Service

3.11. Data Output without the XMPP Channels

The XMPP protocol is used as the main Data Output source of the CCH.
Every API key has it´s own XMPP channel, where he can connect to and will receive all datasets the key is permitted to see, as described in chapter 4.
In addition, a dataset can be retrieved directly.The API key that delivers data can ask for its submissions of the last 15 minutes and will get a JSON formatted response:

3.11.1. Submissions of the last 15 minutes

GetSubmissions	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/reports/
Headers	Authorization: Token token="API_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/reports/
Path Params	/ID_KEY (read or write key)
data: {.....}	-
Response	JSON (sample): { "confidence_level":0.11, "report_category":"eu.acdc.minimal", "report_type":"API Test V2", "reported_at":"2015-05-22T07:03:17Z", "source_key":"uri", "source_value":"http://example_Domain.de", "timestamp":"2015-05-21T17:00:00+02:00", "version":1, "report_id":"553dex357765626c33040000" }, { "confidence_level":0.11, "report_category":"eu.acdc.minimal", "report_type":"API Test V2", "reported_at":"2015-05-22T07:05:03Z", "source_key":"uri", "source_value":"http://example_Domain.de", "timestamp":"2015-05-21T17:00:02+02:00", "version":1, "report_id":"553dex9f776562513a000000" } see: curl -v -k -XGET -H 'Authorization: Token token=" Your_API_Key "' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports

Example: API (read) key with the access token “API_TOKEN” asks for his submissions:

```
curl -v -k -XGET -H 'Authorization: Token token="API_TOKEN"' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports
```

output: Three incidents have been submitted by this key in the last 15 minutes.


```
{
"confidence_level":0.11,
"report_category":"eu.acdc.minimal",
"report_type":"API Test V2",
"reported_at":"2015-05-22T07:03:17Z",
"source_key":"uri",
"source_value":"http://example_Domain.de",
"timestamp":"2015-05-21T17:00:00+02:00",
"version":1,
"report_id":"553dex357765626c33040000"
},
{
"confidence_level":0.11,
"report_category":"eu.acdc.minimal",
"report_type":"API Test V2",
"reported_at":"2015-05-22T07:05:03Z",
"source_key":"uri",
"source_value":"http://example_Domain.de",
"timestamp":"2015-05-21T17:00:02+02:00",
"version":1,
"report_id":"553dex9f776562513a000000"
},
{
"confidence_level":0.11,
"report_category":"eu.acdc.minimal",
"report_type":"API Test V2",
"reported_at":"2015-05-22T07:06:07Z",
"source_key":"uri",
"source_value":"http://example_Domain.de",
"timestamp":"2015-05-21T17:01:03+02:00",
"version":1,
"report_id":"553dexdf776562513a010000"
}
```


3.11.2. Update a report

With the returned “report_id”, it is possible to update a report.
Only keys, which have submitted a report or have a sharing policy with the submitting Key in place are able to update a report. Every Key that is not allowed by the sharing policies will get a “access denied” error while trying to update.

Show the report with the ID : ” 551418337765623b7aac1600”

```
curl -k -XGET -H 'Authorization: Token token=" Your_API_Key "' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports/551418337765623b7aac1600
```

Output:

{"confidence_level":0.11,"ip_protocol_number":6,"ip_version":4,"report_category":"eu.acdc.attack","report_subcategory":"dos","report_type":"[ATTACK] [ECO][Test] TCP SYN Flood by eco","reported_at":"2015-03-26T14:31:15Z","source_key":"ip","source_value":"121.154.32.23","src_ip_v4":"121.154.32.23","src_mode":"plain","timestamp":"2015-03-26T15:30:33Z","version":1,"report_id":"551418337765623b7aac1600"}

Here we see a report with the confidence_level 0.11, which is very low.
We can update the report with a new confidence_level or other details:

```
curl -k -XGET -H 'Authorization: Token token=" Your_API_Key "' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/reports/551418337765623b7aac1600
```

output:

{"confidence_level":0.7,"ip_protocol_number":6,"ip_version":4,"report_category":"eu.acdc.attack","report_subcategory":"dos","report_type":"[ATTACK] [ECO][Test] TCP SYN Flood by eco","reported_at":"2015-03-26T14:31:15Z","source_key":"ip","source_value":"121.154.32.23","src_ip_v4":"121.154.32.23","src_mode":"plain","timestamp":"2015-03-26T15:30:33Z","version":1,"report_id":"551418337765623b7aac1600"}

We can see, the confidence_level has been updated to 0.7

3.11.3. Data queries

It is possible to query the CCH for Data on a given argument, like

- ASN
- Domain

Get Submissions per domain	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/ domains /
Headers	Authorization: Token token="API Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/domains/
Path Params	-
data: {.....}	-
Response	JSON (sample): [{"confidence_level":0.11,"report_category":"eu.acdc.minimal","report_type":"Test-Report","reported_at":"2015-04-22T09:53:40Z","source_key":"uri","source_value":"http://example_domain.com","timestamp":"2014-09-30T17:00:00+02:00","version":1,"report_id":"54365b24776562224b000000"}] see: curl -v -k -XGET -H 'Authorization: Token token=" Your_API_Key "' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/domains/ example_domain.com

This API call returns all incidents that are stored for the given domain, based on the view rights oft he calling API key. For privacy and security reasons, only super-user Keys will get the complete list of results. Other calling API keys are limited by viewing rules accordingly.

Get Submissions per ASN	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/ asns /
Headers	Authorization: Token token="API Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/asns/
Path Params	-
data: {.....}	-
Response	JSON (sample): [{"confidence_level":0.11,"report_category":"eu.acdc.minimal","report_type":"Test-Report","reported_at":"2015-04-22T09:53:40Z","source_key":"uri","source_value":"http://example_domain.com","timestamp":"2014-09-30T17:00:00+02:00","version":1,"report_id":"54365b24776562224b000000"}] see: curl -v -k -XGET -H 'Authorization: Token token=" Your_API_Key "' -H 'Content-Type: application/json' https://webservice.db.acdc-project.eu:3000/api/v2/asns/AS16276

This API call returns all incidents that are stored for the autonomous system (AS), based on the view rights of the calling API key. For privacy and security reasons, only super-user Keys will get the complete list of results. Other calling API keys are limited by viewing rules accordingly.

3.11.4. X-ARF output for CERTS

X-ARF is an email format to report different types of network abuse incidents to network owners. Especially fort he CERT Community we have chosen X-ARF reporting to be included in the CCH functionality.
A Flag was introduced in the API Key creation process to show that a receiving Key does not needs to be connected to the XMPP Server to fetch data but to get his incident reports by a Mail in X-ARF format.

Example of a API Key for a CERT

{	
"id":12345,	Number of the API key
"access_token":"****access_token****",	Secret Access Token
"ttl":308262214,	Time To Live for the Key in seconds
"email":"contact@cert.TLD",	Contact Mail adress
"description":"Neme oft he CERT",	Short description , preferrably name of CERT
"superuser":false,	
"created_at":"2015-03-04 12:51:25 UTC",	Creation date
"updated_at":"2015-05-06 09:21:46 UTC",	Updated
"group_id":7,	CERT Group
"key_type":"read",	A read key to get Data out oft he CCH
"data_schema_url":null,	No schema needed for read keys
"asns":["ASxxx",“ASyyy“],	List of ASN the CERT signs responsible
"ips":[],	List of IP or IP Ranges
"x_arf":true	X-ARF Flag
}	

Figure 20: CERT API Key

If the flag “x_arf”:true then all data for this key is not processed by the XMPP server but send via mail in X-ARF format to the contact eMail address, which is stored in the Key.

Example for an X-ARF reports: eu.acdc.attack report


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


Figure 21: sample eu.acdc.attack

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

Dear customer,
this is an X-ARF report on activity related to your ASNs and networks that was recently submitted to the ACDC project.
A host attacks another system: TCP SYN Flood
For more information on the ACDC project please visit http://www.acdc-project.eu/
For more information on X-ARF please visit http://www.x-arf.org/

Attachment: none
Category: abuse
Confidence-Level: 1.0
Date: '2014-06-15T15:47:12Z'
Dst-Ip-V4: 198.51.100.111
Dst-Mode: anon
Dst-Port: 80
Ip-Protocol-Number: 6
Ip-Version: 4
Report-Description: 'TCP SYN Flood'
Report-ID: *some-report-id*@acdc-project.eu
Report-Subcategory: dos
Report-Type: eu.acdc.attack
Reported-At: '2014-06-15T15:44:45Z'
Reported-From: xarf@acdc-project.eu
Schema-URL: https://www.acdc-project.eu/eu.acdc.attack_2.0.0.json
Source: 141.39.242.16
Source-Type: ipv4
Src-Ip-V4: 192.0.2.14
Src-Mode: plain
User-Agent: acdc-xarf-export/1.0
Version: 0.2

Figure 22: X-ARF translation of the report

The submission of multiple reports is supported by the feature X-ARF bulk messages. It bundles several stand-alone reports into one email.

The Schema is described in Deliverable 1.7.2 Appendix E.

3.12. Internal Statistics

The CCH counts every submission in real time and stores it with the Key_ID of the submitting partner and basic information in a SQL Database, to perform basic statistics. The CCH stores the submitting API-KEY_ID, the Date and the "report_category" (the schema), in which the incident was submitted to the CCH and adds the numbers of incidents per category.
Only Super-User Keys will see all data, if a query from a "normal" Key is sent to the CCH, the returned JSON object will only show data from this Key.

Statistics for one day:

GetStatistics per Date Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/stats/
Headers	Authorization: Token token="Super-user_API_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/stats/
Path Params	/date
data: {.....}	-
Response	JSON (sample): { "1":{"start_date":"2015-06-13","end_date":"2015-06-13","total":0,"count_by_categories":{}}, "3":{"start_date":"2015-06-13","end_date":"2015-06-13","total":0,"count_by_categories":{}}, ... "801":{"start_date":"2015-06-13","end_date":"2015-06-13","total":7446,"count_by_categories":{"eu.acdc.spam_campaign":7446}}, "802":{"start_date":"2015-06-13","end_date":"2015-06-13","total":84,"count_by_categories":{"eu.acdc.spam_campaign":84}}, "803":{"start_date":"2015-06-13","end_date":"2015-06-13","total":796,"count_by_categories":{"eu.acdc.spam_campaign":796}} } see: curl -XGET -H 'Authorization: Token token="Super-user_API_Key"' -H 'Content-Type: application/json' -k https://webservice.db.acdc-project.eu:3000/api/v2/stats/2015-06-13

Figure 23: Incidents on a given Date

Statistics for a Timeframe:

GetStatistics per Time Interval Service	
Protocol	SSL
Host	HOSTNAME
Port	3000
Base URL	/api/v2/
Service Name	/stats/
Headers	Authorization: Token token="Super-user_API_Key"
Method	GET
Content Type	application/json
QueryString	-
URL	https://HOSTNAME:3000/api/v2/stats/
Path Params	/start_date/end_date
data: {.....}	-
Response	JSON (sample): { "1":{"start_date":"2015-05-13","end_date":"2015-06-13","total":0,"count_by_categories":{}}, "3":{"start_date":"2015-05-13","end_date":"2015-06-13","total":0,"count_by_categories":{}}, "4":{"start_date":"2015-05-13","end_date":"2015-06-13","total":0,"count_by_categories":{}}, ... "801":{"start_date":"2015-05-13","end_date":"2015-06-13","total":58210,"count_by_categories":{"eu.acdc.minimal":1,"eu.acdc.spam_campaign":58209}}, "802":{"start_date":"2015-05-13","end_date":"2015-06-13","total":594,"count_by_categories":{"eu.acdc.minimal":1,"eu.acdc.spam_campaign":593}}, "803":{"start_date":"2015-05-13","end_date":"2015-06-13","total":7058,"count_by_categories":{"eu.acdc.minimal":1,"eu.acdc.spam_campaign":7057}} } see: curl -XGET -H 'Authorization: Token token="Super-user_API_Key"' -H 'Content-Type: application/json' -k https://webservice.db.acdc-project.eu:3000/api/v2/stats/2015-05-13/2015-06-13

Figure 24: Incidents in a given Time Frame

The data output is a JSON object. Although only one Schemata should be used by a API Key, the example output shows 2 different schemata for some API Keys – this was mostly due to functionality tests. In the CCH, the policy of 1 Key – 1 Schema is strongly promoted.

- All data submissions for a given day:

```
curl -XGET -H 'Authorization: Token token="API-Token"' 'Content-Type: application/json' -k https://webservice.db.acdc-project.eu:3000/api/v2/stats/2015-04-30
```

Figure 25: Query for all submitted reports of April 30th.

output :


```
{
"1":{"start_date":"2015-04-30","end_date":"2015-04-30","total":0,"count_by_categories":{}},
"3":{"start_date":"2015-04-30","end_date":"2015-04-30","total":0,"count_by_categories":{}},
...
...
...
"669":{"start_date":"2015-04-30","end_date":"2015-04-30","total":63,"count_by_categories":{"eu.acdc.malicious_uri":63}},
"670":{"start_date":"2015-04-30","end_date":"2015-04-30","total":0,"count_by_categories":{}},
"671":{"start_date":"2015-04-30","end_date":"2015-04-30","total":2,"count_by_categories":{"eu.acdc.malicious_uri":2}}
}
```


Figure 26: output of data query

- Data submissions for a timeframe

```
curl -XGET -H 'Authorization: Token token="API-Token"' 'Content-Type: application/json' -k https://webservice.db.acdc-project.eu:3000/api/v2/stats/2015-05-01/2015-05-31
```

Figure 27: Data query for May, 1st to 31st.

output:


```
{
"1":{"start_date":"2015-05-01","end_date":"2015-05-31","total":0,"count_by_categories":{}},
"3":{"start_date":"2015-05-01","end_date":"2015-05-31","total":0,"count_by_categories":{}},
...
...
"669":{"start_date":"2015-05-01","end_date":"2015-05-31","total":1095,"count_by_categories":{"eu.acdc.malicious_uri":1095}},
"670":{"start_date":"2015-05-01","end_date":"2015-05-31","total":1787,"count_by_categories":{"eu.acdc.malicious_uri":1787}},
"671":{"start_date":"2015-05-01","end_date":"2015-05-31","total":24715,"count_by_categories":{"eu.acdc.malicious_uri":24715}}
}
```


Figure 28: output for data submissions from May, 1st to 31st

The output of the CCH statistics function is a JSON Object, containing a list of API Keys (from 1 to the last key) with the submission count for the given Timeframe and the category in which the reports have been delivered.
The JSON object can be processed externally and the API_Key_ID numbers can be linked to a partners organisation name, so we can create a excel sheet of this data to visualize it.

Datasets Mai 2015							
Partner	Key_ID		Total Number	Schema used / Dataset	Count	Schema used / Dataset	count
montimage	109		54	eu.acdc.minimal	54		
cassidian	333		95	eu.acdc.malware	95		
atos	358		295	eu.acdc.fast_flux	295		
DFN-Cert	365		132	eu.acdc.attack	132		
cyscon	366		429555	eu.acdc.malicious_uri	429550	eu.acdc.minimal	5
inteco	378		7530	eu.acdc.fast_flux	7530		
inteco	397		263	eu.acdc.malicious_uri	263		
de-cix	412		8978295	eu.acdc.attack	8978294		
signal spam	430		9508	eu.acdc.malicious_uri	9508		
cyber defcon	440		1	eu.acdc.bot	1		
inteco	454		592	eu.acdc.malware	592		
atos	463		311	eu.acdc.bot	311		
inteco	491		83024	eu.acdc.bot	83024		
inteco	493		252	eu.acdc.malware	252		
eco	497		1	eu.acdc.minimal	1		
Ifis	508		32	eu.acdc.c2_server	32		
atos	512		7714	eu.acdc.attack	7714		
garr	516		1458961	eu.acdc.attack	1458961		
garr	517		432501	eu.acdc.malicious_uri	432500		
cert be	518		201860	eu.acdc.malware	201860		
carnet	519		20	eu.acdc.bot	20		
carnet	520		443	eu.acdc.malware	443		
carnet	521		581	eu.acdc.spam_campaign	581		
carnet	522		186	eu.acdc.fast_flux	186		
carnet	524		874	eu.acdc.malware	874		
carnet	525		3205	eu.acdc.malicious_uri	3205		
carnet	527		605	eu.acdc.malicious_uri	605		
carnet	528		264	eu.acdc.malicious_uri	264		
carnet	529		1940	eu.acdc.attack	1940		
montimage	532		29	eu.acdc.minimal	29		
gdata	535		7401	eu.acdc.malicious_uri	7401		
gdata	536		104643	eu.acdc.malware	104643		
gdata	537		2	eu.acdc.c2_server	2		
cert-ro.eu	538		615692	eu.acdc.attack	615692		
telecom italia	547		154	eu.acdc.malware	154		
telecom italia	548		9517621	eu.acdc.attack	9517620		
atos	553		482	eu.acdc.malicious_uri	482		
atos	562		110	eu.acdc.malware	110		
Telefonica	563		120962	eu.acdc.attack	120962		
Telefonica	566		2463	eu.acdc.malicious_uri	2463		
Telefonica	568		8	eu.acdc.malicious_uri	8		
Telefonica	569		1462	eu.acdc.malicious_uri	48	eu.acdc.malware	1414
Montimage	572		56	eu.acdc.minimal	56		
FCCN	576		667	eu.acdc.fast_flux	667		
Cyberdefcon	589		2930	eu.acdc.c2_server	2930		
xlab	620		99	eu.acdc.malware	99		
atos	625		67	eu.acdc.malicious_uri	67		
inteco	669		1095	eu.acdc.malicious_uri	1095		
inteco	670		1787	eu.acdc.malicious_uri	1787		
inteco	671		24715	eu.acdc.malicious_uri	24715		
inteco	677		11795	eu.acdc.malicious_uri	11795		
inteco	679		3392	eu.acdc.malicious_uri	3392		
uni.lu	697		23	eu.acdc.bot	23		
ifis	706		4162	eu.acdc.c2_server	4162		
ifis	707		332	eu.acdc.c2_server	332		
montimage	709		12	eu.acdc.minimal	12		
fccn	720		1832	eu.acdc.attack	1832		
fccn	721		14987	eu.acdc.bot	14987		
ifis	768		13000	eu.acdc.malicious_uri	13000		

Figure 29: Data Output formatted and put in Excel - with Partner Names

The adjusted and formatted numbers clearly show an increase in data submissions. An increase by more than 100% can be seen in most datasets, showing that the ACDC partners established a solid messaging system for incidents.

Datasets received by the CCH			
Schema used	**March**	**April**	**Mai**
eu.acdc.attack	9271130	20529966	20703147
eu.acdc.bot	27275	20028	98366
eu.acdc.c2_server	755	36	7458
eu.acdc.fast_flux	2696	965	8678
eu.acdc.malicious_uri	1422486	761160	942148
eu.acdc.malware	150532	199754	310536
eu.acdc.spam_campaign	366	630	581
eu.acdc.minimal	17	53	157
eu.acdc.vulnerable_uri	6	0	0
Total	10875263	21512592	22071071

Figure 30: CCH overall data March - Mai 2015

The numbers over 3 months
Number count in the CCH is found to be accurate – the CCH counts any incident at the stage of delivery. The incident is then forwarded to be enriched with ASN / IP / Country code data and forwarded to the Redis Database and to the XMPP Server at the same time. We have no reason to mistrust the accuracy and validity of this concept.

Time delays in the delivery of data can play a big role. Especially the XMPP server appears prone that queues are not served, when the stack of data to be sent to heavily increases. Especially when data queues from non-received data reach sizes over 100 MB, they have a big influence in the stability of the XMPP server. We have to keep in mind, that every non delivered queue – that means a data stream where the receiving API Key is not connected to, does not listens and does not fetch his data - is stored in the XMPP server and the server has to check for the presence of the associated client to start the data stream. At now over 700 API keys this is time consuming and prone to errors, so we decided to realise the Metrics for which Work package 4 accounts liable in another way.

3.13. Metrics

The Metrics have been implemented internally in the CCH, instead of using a dedicated XMPP channel for data retrieval as it was planned beforehand. Deliverable 4.4 describes the metrics and the access to the metrics.

A dedicated Server - cch-statistics.acdc.de-cix.net (192.168.162.60) - runs within the CCH and connects to web01.acdc.de-cix.net (192.168.162.10) and web02.acdc.de-cix.net (192.168.162.11) to listen to the aggregated data stream and calculate the metrics. The server fetches all the reports submitted to CCH via a Redis subscription.
The metrics, calculated by cch-statistics.acdc.de-cix.net (192.168.162.60), are then send tot he CCH as a JSON report (eu.acdc.metrics) for which interested parties can subscribe via the XMPP channel.

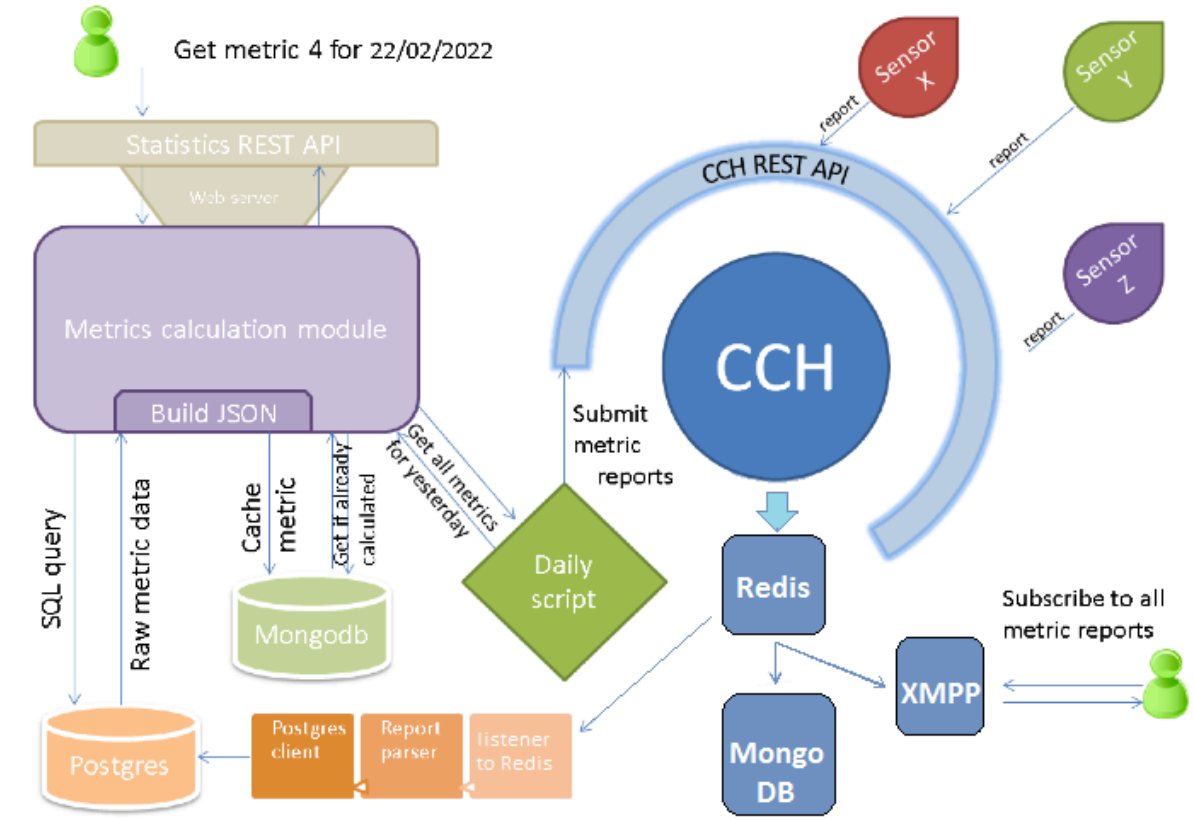

Figure 31: Infrastructure to compute metrics

The CCH JSON reports are received continuously as a stream from the CCH via a redis subscription and are parsed and stored in a local Postgres[10] database. This database is used as temporary storage to facilitate metric computation in the Metrics Calculation module. This module stores metrics computation results in a local MongoDB, calculating metrics on every request.
This way, when a request for a specific metric and time interval arrives to the Metrics Calculation module, it checks whether the corresponding values are already in the MongoDB cache. If so, the value is returned, but if it is not already present, the module posts the SQL query corresponding to the requested metric to the Postgres database. The results are stored in the MongoDB cache.

The data format to represent the raw results of the computation of each metric is adopted from the Research Workflow, as described in D1.7.2 "Data Formats Specification", and specified using the corresponding JSON schema.
There are two ways to retrieve metric data from the statistic server:

- Via XMPP, subscribing to the metrics XMPP channel that is maintained by the University of Luxembourg – Channel ID 460
- Directly from the Statistics Server, by means of a REST API.
  The consumer of metrics results will send an HTTP request to the web service specifying the metric ID and the time frame. The web server will query the Metrics Calculation module for the requested data, which in turns checks the MongoDB cache, and returns a JSON list of records that match the request, one per day per metric.

The resulting JSON object can be used to visualise the metric data:

[10] http://www.postgresql.org

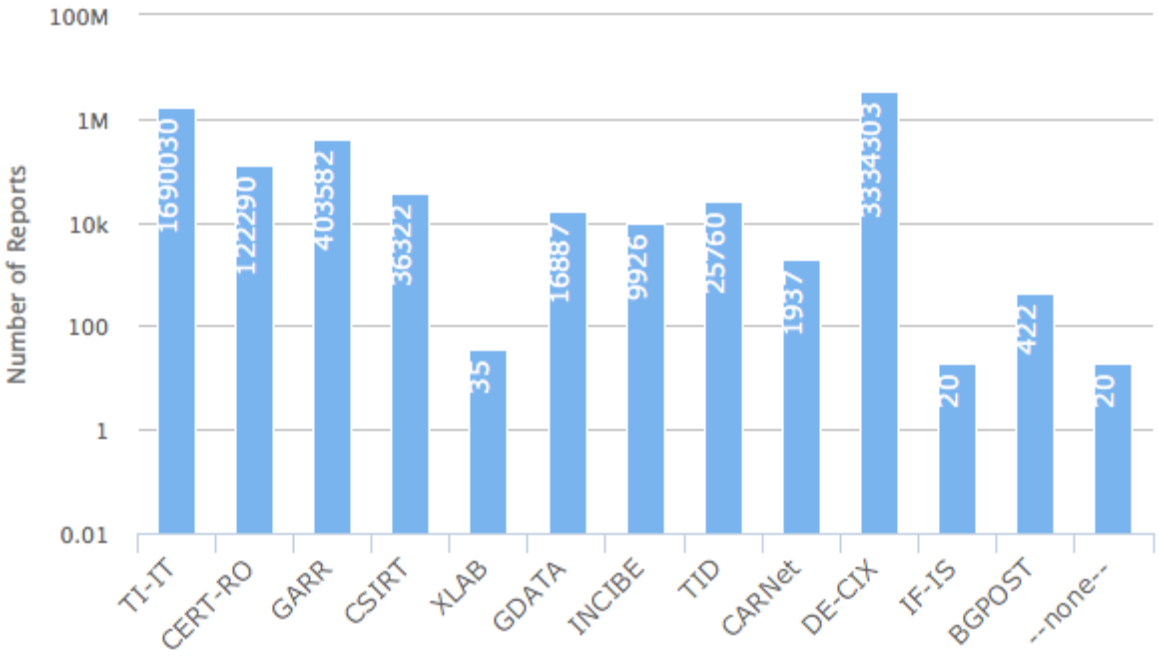

Figure 32: reports per partner (last 7 days)

We can divide between Data Quality Metrics, Botnet impact metrics and operational metrics. The definitions and examples can be found in Deliverable 4.4, Annex III – Specification of Metrics.

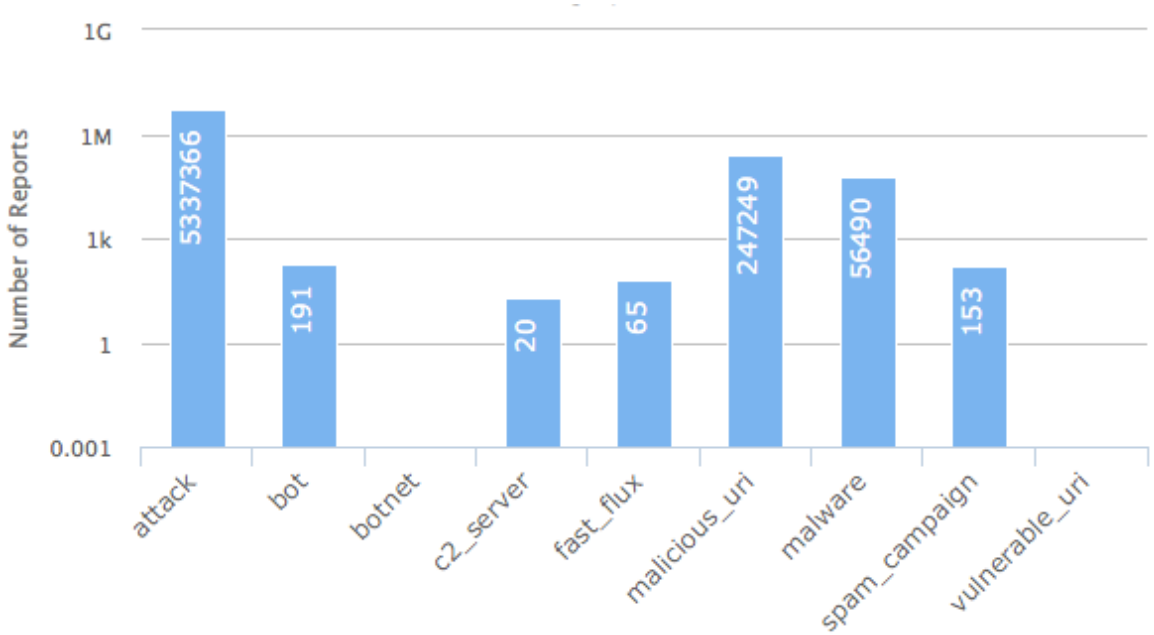

Figure 33: reports per Category (last 7 days)

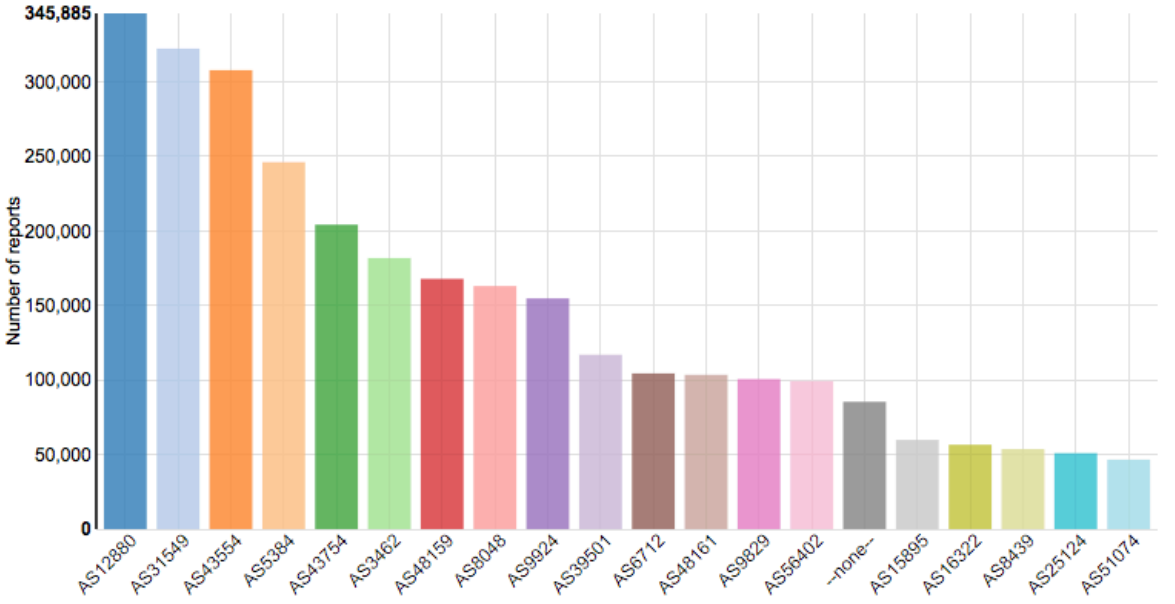

Figure 34: Top ASN by the number of reports submitted (last 7 days)

4. Penetration Test Results

4.1. Introduction

In order to demonstrate the safety of the chosen implementation, we found it necessary to conduct a penetration test of the CCH and the community portal. Additionally it was a recommendation by the reviewers to conduct a penetration test to clearly show the safety of the main ACDC components.
As a member of the project team, the DFN-CERT Services GmbH was tasked to plan and conduct the above penetration test.

4.2. Test objectives

The ACDC project has, as of yet, no written down set of security policies. Nonetheless, it is possible to deduct certain security goals of the ACDC project from the available documentation:

• No unauthenticated user should be able to access any data from either the CCH or the CP.
• It should not be possible to alter existing data or add data as an unauthenticated user.
• Authenticated users should not be able to change their identity or access data of other users outside their scope and permissions.
• Likewise, Authenticated users should not be able to alter or add data of other users outside their scope and permissions.

Thus, the penetration tester should try to successfully violate any one of the above security goals, i. e.

• Obtain access to data as an unauthenticated user
• Alter existing data or add data as an unauthenticated user.
• Change the identity or obtain an identity as an unauthenticated user
• As an authenticated users, obtain access to data outside the scope and permissions of that identity.
• As an authenticated users, be able to alter or add data of other users outside their scope and permissions.

Access to operating system accounts would count as a part success, if no access to other data or to system accounts could be obtained, otherwise as it would practically grant access to all data, it would count as a full success.
Furthermore, the penetration testers should also look out and report any other irregularities they find during the course of the test.

4.3. Tools used

Several tools are to be used in the penetration test:
• Nmap 6.47[11] for basic port scanning of the Internet-facing systems. nmap -sT forTCP services and nmap -sU for UDP services.
• OpenVAS 6[12] for basic vulnerability scanning. The signatures are updated to 23rd of July 2015.
• Burp Suite Pro 1.6.22 (64bit)[13] a general web application security testing tool. Also used is the xssValidator[14] plugin at version 1.3.0 for automation and validation of XSS vulnerabilities.

[11] https://nmap.org
[12] https://openvas.org
[13] https://portswigger.net/burp
[14] https://github.com/nVisium/xssValidator

• sqlmap 1.0-dev-a905b8d[15] an tool for automating the process of detecting and exploiting SQL injection flaws.
• brakeman 3.0.5[16] a static analysis security scanner for Ruby on Rails applications.
• mitmproxy0.13[17] A program to intercept and inspect traffic flows, especially HTTP.

Additionally, access to the Source Code of both the CP and the CCH has been given to the penetration testers, together with the infrastructure map and logins to the infrastructure, so some superficial code review might be possible. In any case, the penetration test will thus be a white box test, as the testers will have in-depth advance knowledge of the target structure.

4.4. Test results

No obstacles have been found.
All findings are confidential and are described in the "*Penetration Test of the Central Clearing House and Community Portal*" document, provided by DFN-CERT.

4.5. Recommendations

All recommendations are confidential and are described in the "*Penetration Test of the Central Clearing House and Community Portal*" document, provided by DFN-CERT.

[15] http://sqlmap.org/
[16] http://brakemanscanner.org/
[17] https://mitmproxy.org/

5. References

http://acdc-project.eu/documents/acdc-deliverables/

Deliverable 1.7.2
Deliverable 2.3
Deliverable 2.4
Deliverable 4.4

http://www.gesetze-im-internet.de/englisch_bdsg/englisch_bdsg.html
http://www.dataprotection.ie/docs/EU-Directive-95-46-EC-The-Recitals-Page-1/90.htm
https://cybox.mitre.org/
http://redis.io
http://www.mongodb.org/
https://prosody.im
https://www.trusted-introducer.org/
http://www.postgresql.org
Nmap homepage: https://nmap.org/
OpenVAS homepage: https://openvas.org/
Burp suite homepage: https://portswigger.net/burp/
xssValidator page on Github https://github.com/nVisium/xssValidator
Sqlmap homepage: http://sqlmap.org/
Brakeman homepage: http://brakemanscanner.org/
Mitmproxy homepage: https://mitmproxy.org/

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.