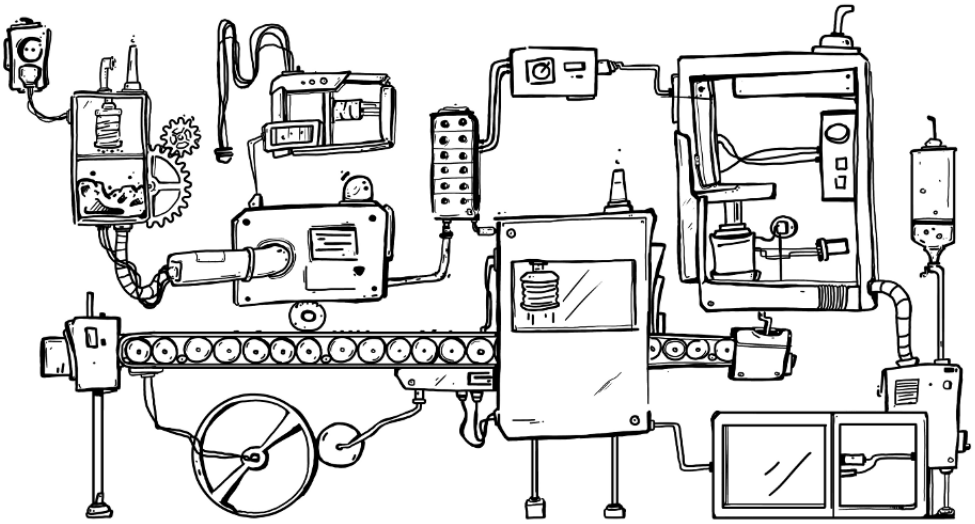


Actionable Information for Security Incident Response

November 2014

European Union Agency for Network and Information Security www.enisa.europa.eu

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Authors:

This document was created by the CERT capability team at ENISA in consultation with CERT Polska / NASK (Poland)[1]

Project Manager:

- Cosmin Ciobanu (ENISA)

Acknowledgements:

- Luc Dandurand (NATO)
- Mark Davidson (MITRE) and STIX / TAXII team
- Bernd Grobauer (Siemens)
- Pavel Kácha (CESNET)
- Aaron Kaplan (CERT.at)
- Andrew Kompanek (CERT/CC)
- Maarten Van Horenbeeck (FIRST)

Additionally, our "thank you" goes to all participants of the Birds of a Feather session *Finding & Sharing Actionable Information,* co-organized by CERT Polska, US-CERT, and Microsoft during the 26th annual FIRST conference in Boston.[2]

Contact

For contacting the authors please use cert-relations@enisa.europa.eu

For media enquires about this paper, please use press@enisa.europa.eu.

[1] Paweł Pawliński, Przemysław Jaroszewski, Piotr Kijewski, Łukasz Siewierski, Paweł Jacewicz, Przemysław Zielony, Radosław Żuber

[2] See http://first.org/conference/2014/program#pbof-finding-sharing-actionable-information

Legal notice

Copyright Notice

ISBN: 978-92-9204-107-6 doi: 10.2824/38111 Catalog number: TP-05-14-107-EN-N

Executive summary

In the world of incident response, information is everything. The sooner incidents and vulnerabilities are detected and understood, the faster they can be handled and the less damage is caused. Accurate and timely information may help incident handlers reduce the number of infections, or address vulnerabilities before they are exploited. Unfortunately, although security information sharing is now commonplace, it has not always improved the situation for incident response teams. Extracting timely information, that can be immediately acted on from vast amounts of all types of data flowing in, remains a challenge. This type of information is referred as “actionable information” and identified as one of the fundamental building blocks of successful incident response.

This document is intended as a good practice guide for the exchange and processing of actionable information. The report is relevant to incident response in all types of organizations, the primary audience of this study isnational and governmental CERTs. The scope of the study is purposefully broad. Many of the issues related to making information actionable for CERTs have not been adequately explored in previouspublications. The goal for this report was to touch on a wide variety of challenges that should be addressed in the area of processing information. Another goal of the study is also to outline a general framework that could be used as the basis for future, more detailed, studies.

The main contributions of this study are as follows:

- A definition of actionable information for CERTs and identification of its 5 key properties: relevance, timeliness, accuracy, completeness, ingestibility.
- Introduction of a generalized information processing pipeline for the processing of actionable information. This pipeline consists of 5 stages: collection, preparation, storage, analysis and distribution. Each stage is discussed in detail with recommendations on how to approach implementation.
- A set of 3 detailed case studies that cover various aspects of handling actionable information by CERTs: “Using indicators to enhance defense capabilities,” “Improved situational awareness through botnet monitoring, ” “Effective data exchange on a national level.”
- A hands-on exercise that expands on these case studies by walking a student through a concrete information processing and sharing scenario.
- An inventory of 53 information sharing standards and 16 information management tools relevant to the concept of actionable information. This inventory is available as a separate document, titled “Standards and tools for exchange and processing of actionable information”.
- Identification of gaps and recommendations in the exchange and processing of actionable information. In particular, despite the improvement ingeneral awareness of the issues involved, the emergence of new standards such as STIX/TAXII, and new tools, the exchanges have not yet reached full maturity.

Based on this study, it is recommend CERTs abide by the following three general principles when building an information-sharing capability:

- Establish a doctrine to set expectations among the CERT community. Define clear sharing rules and labels on the data exchanged, as well as expectations for handling and any specific actions that should be taken by the recipient.
- Try not to start from scratch. Consider what has already been developed and can be leveraged immediately.
- Explore the possibility of applying additional processes that can provide more context and make the information more actionable.

As a set of general recommendations to CERTs and the following are suggested:

- If possible, standard data formats and transports mechanisms should be used. The accompanying inventory document contains a reference to standards that are currently in use within the incident handling community.
- For some recipients, standard formats may be less helpful for distributing actionable information since they lack the capability to process them. Simpler methods should be used in these cases (e.g., human-readable text). Alternatively, a CERT may consider providing automatically-generated, human-readable reports along with the original data in a structured standard format.
- Adjust the way the information is processed and distributed based on the requirements and constraints for each data type. Be sensitive to the overhead of data formats for large volumes of data, and use more elaborate formats for less frequent reports.

The assumptions are that this study will be of help to CERTs and the information security community in general to better understand the issues involved in the creation, sharing, and processing of actionable information as well as aid the development of tools in this area.

Table of Contents

1 Introduction

In the world of incident response, information is everything. The sooner incidents and vulnerabilities are detected and understood, the faster they can be handled and the less damage is caused. Accurate and timely information may help incident handlers reduce the number of infections, or address vulnerabilities before they are exploited. Unfortunately, although security information sharing is now commonplace, it has not always improved the situation for incident response teams. Hundreds of megabytes of data received daily may turn out to be useless if there is no way to cross-check and verify it, or easily manage and apply in the incident handling processes. With narrowing time windows for reaction, it is crucial to obtain or extract timely information that can be immediately acted upon: actionable information.[3]

Extracting actionable information from vast amounts of raw data – often coming from many sources –is crucial but challenging. While data sharing can, and often is done automatically, processing usually requires the intervention of human analysts who understand the nature of a problem. Typically, these analysts should also be involved indefining how and what data should be collected in the first place. This may not only cause delays, but also requires data to be presented in a form that is understandable by humans. Further complicating matters, information that is actionable by one party may not be relevant and actionable to another.

This document is an attempt to describe these and other problems, gaps, and shortcomings of existing solutions as part of an exploration of best practices in extracting and exchanging actionable information.[4]

1.1 Audience and scope

The main goal of this report is to provide guidance for CERTs in regard to one of their core activities: handling information obtained from multiple sources, and translating that information into actions on behalf of their constituents. The scope of responsibilities and capabilities of a CERT vary greatly between organizations[5] – a national CERT usually works in a different way than an internal team[6] within an enterprise – therefore, it is almost impossible to provide practical advice suitable for all environments. Even the mission of national CERTs may differ. Consequently, rather than try to document comprehensive, detailed guidance, in this document a wide variety of issues that should be improved in the area of processing information are touched. The document outlines a general framework that could be used as a reference model for more detailed studies.

The primary recipients of this document should be the personnel responsible for data processing, analysis and exchange, designers and developers of systems supporting these activities, as well as managers and executives responsible for processes and procedures in these areas. Although in this report the focus is on the needs of national and governmental CERTs, the assumption is that much of what is contained in this report can be applied to any CERT or security organization that has responsibility for collecting security data and translating it into actions on behalf of the organizations it is helping to secure.

[3] The term will be discussed more thoroughly in section 1.2.

[4] Note that purposefully the term “intelligence” is not used in in this document to refer to data exchanged, given the ambiguity and hype around the term (especially around “threat intelligence”).

[5] For more discussion and details of setting up CERTs, see https://www.enisa.europa.eu/activities/cert/support/guide/files/csirt-setting-up-guide

[6] In this document it is often referred to “internal CERTs” or “CERTs with an internal constituency” to describe a situation when the CERT is within the same organization as its constituency. An opposite situation is for “CERTs with external constituency,” which national-level CERTs are prime examples of. For more information see [1].

1.2 Definition of "actionable information"

In business and management the term "actionable information" is often used to describe market data, reporting on trends and other information that can be used to make specific, strategically sound business decisions. To meet the definition of "actionable" to a business executive it must be relevant, timely, accurate, complete with respect to some set of business goals, and ingestible (the meaning is expand in section 1.3). The same concept can be applied to IT security, where actionable information is used to take actions that mitigate against future threats, or help address existing compromises. It is important to note that the actual scope of what is considered actionable will vary between stakeholders.

Consider an alert from a software product vendor about a particular vulnerability:

- For a CERT with national-level responsibility the alert itself is actionable information if that product is used by its constituents, and would result in an action to author an advisory tailored to its constituents. On the other hand, a list of specific vulnerable hosts would likely not be actionable to this CERT because it is not in their power to fix the problem directly.
- For network administrators, the alert or the advisory from the national CERT is not actionable until it is cross-referenced with a list of potentially vulnerable hosts. Once this is done, the list becomes actionable information for applying patching procedures.

An implied consequence of such a definition is that different stakeholders will see different sets of information as actionable and that some of them may process information which is only actionable to others (directly or when combined with other sources of information). In fact, in the example above, information becomes actionable down the distribution chain.

The types of information that are actionable for CERTs cover a broad scope. Some examples include:

- an identified anomaly in network traffic (for example, a host on the network that normal mostly just receives traffic suddenly starts initiating connections), requiring urgent research,
- a list of IP addresses of known C&C[7] servers which can be null-routed in constituency networks,
- a list of IP addresses which attempted to connect to the abovementioned C&C servers, as those machines may require close investigation,
- a complex description of an incident, including vulnerability identifiers, indicators of compromise and attackers' modus operandi, which results in changes in security policies

Such variation of data sets is a fundamental characteristic of actionable information handled by CERTs. This has led to a proliferation of data formats – often used in narrow contexts and applications –and to a wide variety of approaches and methodologies for processing data to extract actionable information.

1.3 Properties of actionable information

In order to be actionable for the recipient, information must meet certain criteria that allows it to be used without an burdensome amount of additional processing effort or additional communication to validate the information. It must also meet some basic quality requirements. Based upon comments and observations of the expert group, information is defined as actionable when it meets five criteria**: relevance, timeliness, accuracy, completeness, and ingestibility**. It should be noted that all of the criteria must be understood in the context of a particular recipient organization. For example, a large

[7] C&C or C2 – an abbreviation for Command and Control, a term applied to Internet properties used to give instructions and (often) collect information from compromised machines.

ISP that deploys NAT over its networks will require that descriptions of network traffic include source and destination port numbers and non-anonymized IP addresses for the information to be regarded as complete.

1.3.1 Relevance

In order for information to be considered **relevant,** it must be applicable to the recipient's area of responsibility, including the networks, software versions, and hardware platforms of its constituents. For example, indicators of compromise will generally be considered relevant when a threat could affect the recipient's systems. On the other hand, a list of known compromised hosts is only relevant when those specific hosts belong to the organization's constituents. It is therefore highly desirable that CERTs are able to describe their constituency in terms of ASNs, CIDRs, and/or domain names as precisely as possible. This allows an organization to subscribe to tailored data feeds, or to filter feeds based on this description.

1.3.2 Timeliness

The second requirement for actionable information is that it is **timely**. In some cases information about events older than a few hours will be considered irrelevant and non-actionable due to rapid changes in threat characteristics. In practice, though, certain limitations will apply to how timely data is available. Sharing large volumes of information in real time could hinder a recipient's ability to consume it (ingestibility). Also, actionable information is often the result of analysis that requires time, so there is often a tradeoff between timeliness and both completeness and accuracy. It is not uncommon for certain persistent threats to be discovered, analyzed and described months after initial compromise. It does not necessarily mean that such information is not actionable, since hosts may still require cleanup, and additional actions may be needed to minimize damage. The bottom line is that all parties involved in processing actionable information should be careful not to introduce unnecessary delays, and carefully consider the value of analysis that will introduce delay.

1.3.3 Accuracy

Information also needs to be **accurate** – the recipient should be able to consume it immediately, under the assumption that the data has been previously verified and is free of errors (subject to local considerations). The accuracy is really a result of a combination of the confidence asserted by the source, the trust placed in the source and the local context of the receiver. A very important, yet often overlooked, factor impacting both trust and source-asserted confidence is the transparency of sources and the means of collection. It is unfair to expect that the other party will act on information without understanding how it was acquired – especially when that party is expected to disable a host, terminate service with a customer, or take another action that will directly impact users. A track record for accuracy can also be established for a source, based on experience with that source over time provided that the consumer can evaluate the quality of received data (e.g., by estimating false positive and false negative rates by cross-referencing them with other data feeds). In such cases, it is worthwhile to provide feedback to the source to help the producer evaluate and improve the data feed.

1.3.4 Completeness

Actionable information should stand on its own, and provide value to the recipient in the context of the information readily available to the recipient. In many cases, it may be difficult for the producer of information to determine what the recipient may be missing. For example, when a scan is detected, complete information would include not only the source address, but also the destination address, source and destination ports, and possibly some other traffic characteristics. On the other hand, many producers decide to limit the information in fear of revealing too much about their investigative

methods. Legal constraints may be another reason for withholding certain pieces of information. For example, in some cases an organization may consider local IP addresses private data and therefore not easily shared. Practically speaking, what this means is that the producer and the consumer need to carefully consider a variety of factors in order to balance the recipient's need for context and the constraints acting on the producer. Finally, it is important to note that completeness, like accuracy, must always be understood relative to the needs of the recipient. Frequently, sources that are incomplete when considered alone become actionable once combined with other data available to the recipient. Similarly, some contextual information may only be useful to organizations able to act on that information. For example, actor attribution will be valuable to an organization that has law enforcement authority, but less relevant to an organization that only has responsibility for network defense.

1.3.5 Ingestibilty

Finally, information must be **ingestible** – in a form that allows the straightforward import of the data into an organization's information management systems, as well as extraction of important observables and indicators. This feature is mostly associated with formats and transfer protocols used for data sharing. In most cases, the recipient's goal will be that of processing actionable information as fast as possible (e.g., to mitigate an ongoing attack or to close security holes). Consequently, actionable information is largely shared machine-to-machine, even when human reaction is required at some later point. Such processes require standardized formats. Parts of this document, as well as of the supporting inventory, are dedicated to different formats of actionable information and guidelines on using them.

Eventually, actionable information should be shared in a format that is capable of clearly describing it in its complete form, and allow the recipient's systems to consume it painlessly and automatically, including correlating and associating it with other information. The choice of a particular format will depend on a number of factors including the number of recipients, the volume and frequency of data, and the type of information being shared.

1.4 Levels of information

Having established the definition of actionable information for the purpose of this study, further will be characterize the space of actionable information in more detail. The collection, processing and exchange of security-related information – whether actionable or not – encompasses a broad spectrum of activities. Any characterization of information sharing will necessarily require some level of generalization in order to address the security domain as a whole. In this section, a set of categories for actionable information is defined, that will be reference throughout the report, citing concrete examples where possible.

There are multiple ways of categorizing security information, in particular the following two approaches have gained recognition within the community in the last few years:

- The data model defined by the STIX [2] standard (a new exchange format created by MITRE), provides an informal ontology[8] that covers a very wide range of security information: from observables (e.g., descriptions of network flows) to descriptions of high-level concepts like threat actors.
- The Pyramid of Pain [3] defines taxonomy of indicators that is organized according to the value of each indicator type for defending against sophisticated adversaries. It defines multiple levels of information, starting from the most ephemeral, least valuable indicators (hashes of

[8] Unfortunately there does not exist a formal ontology for information security. For more information see [4].

files) to those that are the most difficult for an attacker to change (TTPs describing abstract behavior).

The topic of information type is a challenging one as the range of relevant types of information is very broad and diverse, covering a much larger set than just those identified in the Pyramid of Pain. On the other hand, a detailed discussion of the full breadth of information types is beyond the scope of this document. An academic discussion of modeling security data types is likewise out of scope.

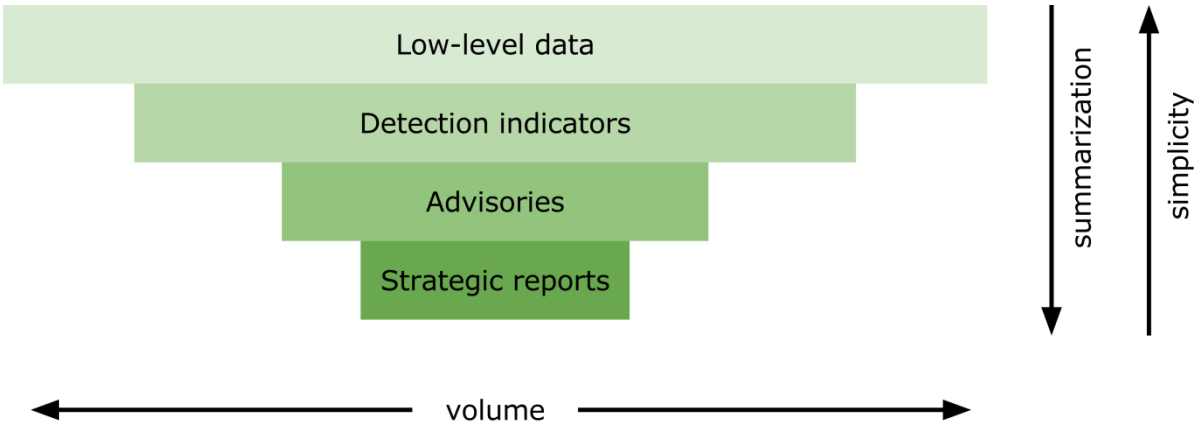
Therefore, for the purpose of this report the information is categorized solely on the basis of how it can be used for the purpose of defending a network (or another resource)– from the consumer point of view. Using this criterion, four distinct levels of information are identified: low-level data, detection indicators, advisories, and strategic reports. Table 1 lists commonly collected and processed types of information for each level.

Table 1. Selected types of security information.

Level of information	Types of information
Low-level	• network flow records and full packet captures • application logs, including typical IDS alerts • samples of executable files, documents, and email messages
Detection indicators	• IP addresses, DNS names, and URLs • specific values of format-specific fields, for example email headers • artifacts (e.g., hashes, registry, keys) related to malware • sequences of low-level events (e.g., syscalls, packets) linked to malicious behavior
Advisories	• vulnerabilities, exploit code, patches and patch status • high-level patterns of activity on a host, service, network or internet level
Strategic reports	• highly summarized threat analyses, written in prose

These levels can also be understood in terms of the steps involved in processing and summarizing each of these different categories of data to obtain increasingly more abstracted, knowledge-rich information. In a typical analysis scenario, analysis begins with the collection of a large volume of low-level data. The analysis of this data yields indicators and advisories, which are then assembled into a strategic report that contains high-level, generalized conclusions. Figure 1illustrates this process.

Figure 1. Levels of information



1.4.1 Low-level data

The first step in implementing an information-driven approach to defense is finding good sources of data. Typically, an internal CERT has access to multiple monitoring systems collecting data related to various activities occurring within an organization. These activities include network traffic, actions performed by users, behavior of applications, and many others. In most cases such data is not useful without additional context. For example, an analyst investigating an incident, would retroactively query data regarding potentially affected systems, making the incident (knowledge of which can be considered high-level information) the context. The context can also be created by matching observed activity against known patterns (such patterns are considered *indicators*, which are discussed in the next section), which is what many automated security systems like IDSs do. Finally, the context can be derived from the properties of the data itself, as is the case with anomaly detection.

An example of such data might be network flow records collected within a corporate network: it could be used to identify exfiltration of intellectual property, find characteristic patterns corresponding to Remote Administration Tools (RATs) or C&C communication, and many other anomalies. While information about these threats may be present in the available dataset, it requires non-trivial analysis to be performed before useful (actionable) data can be obtained.

Other examples include logs – from HTTP servers, authentication facilities, operating systems, etc. – that contain both benign and potentially malicious actions. Also activity that by definition can be considered suspicious, like network traffic coming to a honeypot, often requires further refinement to extract elements describing the threats (e.g., attacking IP addresses) and to remove unrelated noise (e.g. replies to spoofed packets sent by DoS victims on the internet).

In the proposed nomenclature, such sources provide low-level data, as it has to be processed in order to be applied for defensive purposes. In consequence, regardless of its value, information on this level cannot be considered actionable (according to definition in section1.2).If shared, low-level data is typically supporting higher-level information, for example raw logs are often attached to incident reports.

In a typical production environment low-level security data is usually machine generated [5] in volumes that make manual analysis infeasible and its processing is often highly automated (see section 2.4).

1.4.2 Detection indicators

There exist several types of indicators that are relevant to security, but only detection indicators [6] are both actionable and commonly used. In simple terms, a detection indicator is a pattern that can be matched against low-level data in order to detect threats. Such indicators may consist of simple elements of data like IP addresses, URLs, MD5 hashes of files, but in principle any characteristics that can be observed on a network or host's included, for example, consider specific strings in email headers or patterns of invocations of system calls by an application.

Apart from the pattern used for detection, a crucial part of an indicator is the contextual information included with the indicator. Without context, such information would be classified as low-level data. The quality of the contextual information is critical – ideally, it should allow an analyst to clearly understand the threat the indicator is meant to detect, but unfortunately in practice indicators often lack sufficient context. An example of good practice would be the inclusion of malware family and variant with a malware hash, while an example of bad practice would be marking an IP address as malicious without providing further explanation.

If an indicator is of sufficient quality, it can be immediately applied for the purpose of detecting malicious behavior without additional processing apart from translation between formats. In principle, such indicators can be deployed in systems working at various layers of a network –firewalls, proxies, network sensors, SIEMs or on hosts – so detection and even preventive actions could occur automatically in real-time. It follows that information conveyed by detection indicators can be actionable as long as it is possible to translate the indicator into the configuration of an appropriate security control.

The kill-chain model, adapted for the domain of information security by Lockheed Martin [7] can be considered the current state-of-the-art for information-driven, intelligence-based defense against advanced adversaries. Its entire approach is based on leveraging various types of indicators to detect and disrupt various phases of an attack, from reconnaissance, through exploitation, to acting on objectives within the compromised network. The case study “Using indicators to enhance defense capabilities” provides a more in-depth analysis of the application of actionable information in this context.

Information at this level may be the result of manual analysis(e.g. reverse engineering a malware sample by hand), or the result of automated analysis based on analysis of malware behavior as observed in a sandbox, sinkhole, or client- or server-side honeypots. Alarms generated by an IDS with a fine-tuned set of signatures (which are indicators themselves) can also be considered indicators, since they contain network addresses attackers or victims with an associated identifier of a threat.

Frequently shared detection indicators include the following:

- IP addresses of infected machines
- blocks of IP addresses historically associated with malicious activity
- DNS names for botnet C&C servers
- IP addresses of hosts performing malicious actions
- URLs of websites hosting malicious files and performing drive-by downloads
- addresses of misconfigured services that can be abused for DoS attacks

Many tools and platforms that CERTs use for processing and sharing security information –e.g. AbuseHelper and CIF–are essentially indicator management systems, designed to handle only lightweight indicator-like data. It means that their capabilities in regard to handling additional contextual information (e.g., associated vulnerabilities and actors) is limited. The focus of these tools can be explained by their origin –their development was driven by CERTs with external constituencies, which face the problem of effective distribution of indicators on a large scale.

The use of the word detection may be slightly misleading, since detection indicators can be also used for actively blocking threats – the key issue here is that threats must be identified first before they can be blocked. For simplicity, in the following chapters of this document just the term "indicator," will be used when referring to detection indicators. The term "indicator of compromise" (IoC) is often used in this context as well. IoCs describe artifacts and behaviors associated with an intrusion, therefore they can be considered to be a subset of detection indicators, which can be used to identify other types of malicious activity. Also signatures used by anti-virus software and similar solutions can be classified as a subset of indicators, since they provide a mapping from a known bad pattern to an identifier of a threat.

1.4.3 Advisories

This category includes several sorts of information that cannot be directly translated into a process for preventing or detecting threats, but which still provides information for analysts that might trigger a defensive action or help shape the nature of those actions. It includes any piece of potentially actionable information that is not a detection indicator. Notable examples are discussed below.

- *Vulnerability advisories*. Such reports contain information not only about vulnerabilities and affected software (or hardware), but may carry a lot of context as well –occurrences of attacks spotted in the wild, sample exploits, mitigation techniques, etc. Handling such information involves multiple steps –identification of affected assets, risk analysis, development and deployment of protective measures. All of these steps usually require correlation of data present in the report with various organization-specific data.
- *High-level alerts requiring interpretation by analysts*. Some monitoring systems, in particular early warning systems (see section2.4.3.3), provide information on abnormal activities even if they are unable to link them to a particular threat. Such high-level alerts point to interesting events that should be investigated manually, often using low-level data.
- *TTPs of adversaries*. Apart from dealing with individual attacks, it is possible to characterize behavior of an adversary on a higher level. For example, an adversary may typically employ a particular sequence of exploit approaches, or follow a particular timing pattern for the registration, parking, and activation of new malicious domains. If such information is available in sufficient detail, new detection indicators may be derived from it and deployed in automated monitoring systems.

Information at this level is very valuable from a defensive point of view. Unfortunately, much of the information produced at this level is not structured in a way that can be easily translated into actions. It is often made available as free-formed textual reports, and frequently analysts must resort to manual analysis to extract relevant data from advisories. High-level information requires structured data formats for exchange. There are ongoing efforts to develop such formats, most notably STIX , but at the time of the writing of this report most of the advisories available to national and governmental CERTs are not in standardized formats.

From the consumer perspective, in order to make information actionable, it must be put in the context of a specific organization and its environment. This especially applies to advisories. Since they describe threats in a more general way than simple indicators, their interpretation requires not only knowledge of the assets that are being protected, but also a good understanding of how the information can be related to the internal operations of an organization. Consequently, handling information of this level is often difficult, and automating processing is even more challenging.

1.4.4 Strategic reports

Information can also come in the form of highly summarized reports that aim to provide an overview of particular situations. The scope of the reports may vary from analyses of individual campaigns to

studies with a global scope. Such information can be used by analysts or policy-makers to support the decision making process and plan for future activities. A good example of a report with strategic information is the Data Breach Report published annually by Verizon, [8] which can be used to estimate the risks specific to a particular business sector and set priorities for an organization. Similarly, the OECD reports [9] provide a way to compare capabilities of national CERTs and characteristics of reported threats between different countries.

Due to their high-level of abstraction, strategic reports cannot be considered actionable in the sense the term is used in this document. However they may be complementary, providing additional context for the interpretation of other technical data. In particular, they may provide useful information for updating security procedures or modifying technical controls. For example, if a reliable publication suggests that the exfiltration of data through SMTP is becoming more popular, an organization might reprioritize the collection and analysis of outbound SMTP logs.

Although strategic reports represent an important category of security information, the level of abstraction in these reports generally means that is infeasible to automate the translation of reports into actions. For the remainder of this report the focus will be on the issues related to processing the first three categories of information (low-level, indicators and advisories).

2 Processing actionable information

This is the main part of the report, describing how actionable information is obtained, utilized, and shared in a systematic manner. The following conceptual model will be proposed that will provide the structure for the study: a generalized information processing pipeline (Figure 2). Steps in the pipeline –collection, preparation, storage, analysis and distribution –correspond to the natural flow of information in many relevant contexts, such as existing processes in CERTs or workflows utilized by systems that are used to manage security data. The purpose of the model is to facilitate discussion of the complex issues associated with information processing. The assumption is that this is a useful way to conceptualize information processing, not claiming that it will address every possible information-handling arrangement. In particular, the attempt is not to address ad-hoc, manual handling of data in much depth. Its main goal is to describe multiple aspects of systematic information handling, i.e. recurring processing tasks associated with incident handling, monitoring, intelligence gathering and related tasks.

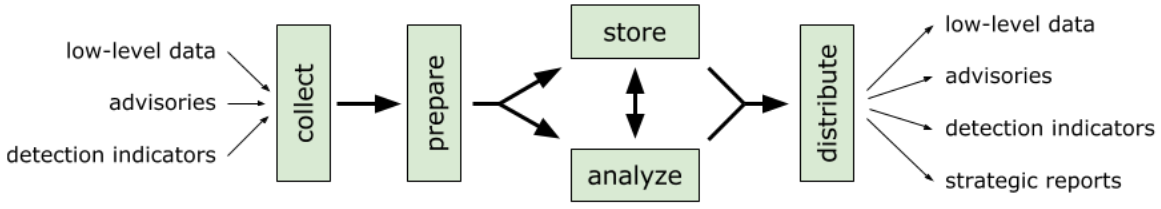
In general the steps are identical for information of all levels, from low-level to advisory (see section 1.4), however, specifics of handling data within individual steps usually depend on the level. Due to the associated lack of actionability, *strategic reports* are not allowed to be an input to the pipeline, although it might be generated during processing. Low-level data is included for cases when it can be made actionable and it can appear both as input and output of the pipeline, since even though this data is generally not directly actionable, it might be attached to actionable information to support research (e.g., a PCAP file might accompany an indicator allowing an analyst to better understand how that indicator was developed).

In practice, there is always more than one processing pipeline being used in parallel, since CERTs handle different types of information separately. A typical example would be that of managing incidents and other high-level information through ticketing systems like RTIR, having a separate infrastructure for a distributed networks of sensors, and another processing path for honeypots deployed in an internal network. Separate pipelines can merge at some point, usually in the analysis step, during fusion of data from multiple sources (see section 2.4 for more in-depth discussion of this topic).

The ability to fuse data could be further improved by consolidating all processing into a single system (with centralized management as another added benefit), but technical limitations make such an approach infeasible. The current best practice, successfully used in large enterprises, [10] is to consolidate processing whenever tools and resources used allow it, and to deploy specialized systems to handle other types of data.

This chapter is structured as follows: each section provides a detailed description of a single processing step, along with examples and relevant recommendations. The emphasis will be on how various parts of the pipeline are implemented by existing tools. However, the model itself is more generic and automation is not strictly required.

Figure 2. Generalized information processing pipeline.



2.1 Collection

The first stage in the information processing pipeline is a process for obtaining the initial data that will serve as the input for the overall process. All of the properties of actionable information (see section 1.3) are influenced by the way in which data is collected. While further processing can improve the quality of information, any shortcomings in the collection process will have major negative consequences for all further stages, and ultimately, for the usefulness of the information produced.

2.1.1 Sources of information: internal vs. external

There are several aspects of collection that are common to all information levels (see section1.4). Probably the most fundamental is the nature of the sources of the data, where "source" refers to the combination of the originating organization (e.g., some vendor), the transport mechanism (e.g., email), and the data format (e.g., IODEF) of a particular stream of information. A source can be internal to the organization, for example, a monitoring system deployed within a corporate network, or an analyst employed by the company. Conversely, external sources include companies providing reputation services, a national CERT, or any other source that is not under the direct control of the recipient.

Reliance on external sources has multiple consequences, most importantly uncertainty regarding the accuracy, which is a product of the confidence asserted by the source and the trust in the source itself. External sources often do not provide exact information on their collection methods, which significantly contributes to the uncertainty faced by the recipient. In some cases information comes through a proxy organization (e.g., a data clearinghouse like Shadowserver) where it is transformed, and some features –like the original source –are hidden from the recipient. This makes it more difficult to evaluate the source. The effective exchange of information between different organizations can also be hampered by confusion regarding the meaning or completeness of some elements of information. One of the most glaring examples to be found in some real data feeds is the under specification of timestamps by omitting time zone information. Internal sources, where the collection methods are to some degree under the control of a CERT, generally present fewer of such problems. In most cases they are easier to integrate and the way they collect data can be adjusted to specifics of a local environment (e.g., by adjusting addresses or ports that a honeypot is listening on). Additionally, internal sources have an important advantage with regard to timeliness. As long as they are actually monitored, and the data that they generate is processed, they will most likely provide information earlier than third-party feeds, since they avoid any delays that would occur during the processing of information by the external provider.

In-house collection capabilities depend greatly on the type of organization. In enterprises where the CERT has intrusion detection responsibilities, the most basic sources are going to be from intrusion detection systems, web application firewalls, proxies, system logs, server honeypots, and other monitoring systems. Some organizations are able to track threats in a more proactive manner, tracking actors that conduct targeted and APT attacks, investigating botnets, malware and spam campaigns, etc. A proactive approach might also include using client-side honeypots, [11] and other technologies to detect malicious and defaced websites relevant to the organization (e.g., detecting watering hole attacks [12]). Organizations with a more global visibility –including large, international IT companies, search engine providers and security service vendors –often have access to data from large, globally distributed monitoring capabilities.

Many types of information either cannot be collected in-house at all, or it would be prohibitively expensive to do so. A prime example would be gathering intelligence data on sophisticated, state-sponsored actors, which require significant analytical resources. Moreover, many threats are targeted,

so potential victims can observe it directly only once they are attacked. Therefore, it is generally difficult to avoid reliance on external sources.

Apart from sources of information that can be used proactively, a CERT collects a lot of heterogeneous information when handling incidents. In general, most of this data can be considered as external, since it is sent by a person reporting an incident or requested from the affected constituents by the CERT itself. In most situations, the CERT has no control over the format of the data that it receives, yet it must make the best use of whatever information is available. The information collection process cannot place restrictions on the way that information is gathered. It means that in many cases data is collected by a human, that uses her domain knowledge and tools appropriate for a particular task. For example a malware sample related to an incident is sent via email in an encrypted archive – someone has to read the email to understand its contents, decrypt the sample, and then put it into a suitable internal system for further reference.

Finally, it is important to recognize that any information that can be collected internally can also be shared with other entities, so an internal source data of one organization can become an external source for the other (sharing issues are discussed in section 2.5).

2.1.2 Level of automation

Another important aspect of an information source is the level of automation used to generate the information: was it generated by an entirely automated system or is it a result of analysis performed by security specialists? Between these extremes lies information that, while generated automatically, was reviewed manually to confirm its correctness. In general, if human analysts are involved in the process of collection (as opposed to performing maintenance work on an automated system), the volume of data will be limited and its cost substantially higher compared to fully automated systems. On the other hand, data produced or vetted manually is likely to be more reliable due to the fact that many errors – in particular false alarms – can be easily detected by specialists with appropriate domain-specific knowledge. Usually the degree of automation is dictated by the level of information, where low-level data is collected directly from sensors, and advisories are often the result of a human analysis. The nature of indicator data is varied in this regard, encompassing both the result of automated analysis (e.g., runtime analysis of malware to extract callback domains) and human analysis (e.g., careful reverse engineering to develop signatures for a piece of malware).

The origin of information, especially related to an incident (e.g., a breach or data leak), can also be viewed in the following way: was it reported by a constituent or did the CERT detect it before the affected party? In the latter case, where discovery of malicious activity is driven by the CERT itself, threats (either affecting the constituency or the broader community) will often be detected significantly more quickly and yield more information than would be possible in the case where the CERT is responding to reporting. If the monitoring and data acquisition infrastructure is already in place and the collection is a continuous process, then it may be much easier to observe the development of attacks and put it into context. An extensive report on the subject of proactive approach to collection of information was published by ENISA in 2011 [13] and expanded later in [11], and is still relevant today.

2.1.3 Properties of data collection methods

From a technical perspective, data collection methods have several important properties that determine their value in a particular setting, and dictate how they are used.

Recurrence. Was obtaining information in a particular manner a singular event (e.g., a credential dump on an online discussion board, a vulnerability that is reported directly by a researcher, or forensics data attached to an incident report), or can it be considered a regular feed of information

(e.g., reputation feeds, vulnerability advisories from large vendors, incident reports shared through automated exchanges)? In general, processing of one-time reports requires more effort, as it cannot be completely automated.

Consumption model. A data source can be queried for the data (**pull** model) or it may send it to the receiver (**push** model). Note that the choice of a particular model has a direct impact on timeliness. The pull model gives more control to the recipient – she or he chooses when to request information from the source. RESTful APIs, which are becoming ubiquitous for internet-facing services, are a prime example of the pull model. However, the model has two downsides. First, it introduces additional latency to the distribution of data, since new data will not be delivered until someone explicitly requests it. Second, when the number of recipients is large, it can introduce scalability problems. In general, the push model does not have these problems and is commonly used when working with high-volume sources (stream processing). The most popular Internet-scale technology for pushing data remains email. A downside of the push model is the limited control of the recipient over the time range, format, and volume of information that is sent – even if a source has a configurable subscription mechanism, in most cases its flexibility is far from what most proper pull APIs (e.g., RESTful) offer(there are exceptions, for example TAXII gives the subscriber a flexible query mechanism). These two models of consumption complement each other and in principle all sources, except ones that have specific technical limitations, can support both. Unfortunately, in practice, implementations usually focus on a single method and do not leave consumers with any choice in this regard.

Granularity. In an event-oriented approach, every element of information is sent separately, for example each packet received by a honeypot or each TCP connection to a sinkhole are reported separately. This method of distribution is used primarily in stream-based publish-subscribe systems (e.g., AbuseHelper is built on this paradigm), although it is also used by some of the pull APIs. The other approach is to send data in batches. Many sources provide files (mostly sent by email or available through HTTP) containing daily digests of observed threats or other kinds of information coming from monitoring systems. It is also common to group data by topic, or example by putting all indicators related to a campaign together. While batch processing can be easier to implement, both on the provider and recipient side, in general it has a negative effect on the timeliness of information – in the case of daily digests, this approach can introduce a delay of up to 24 hours.

The attempt is not to describe all types of actionable and potentially actionable data that can be collected due to the vast number of possibilities, and because existing reports [11][13] cover this subject extensively. The following general, and perhaps obvious, observation can be made: security information of different types, and especially of different levels, is collected by using a very diverse set of methods. Frequently the same kind of data can be collected in multiple ways, each having its own set of advantages and disadvantages, e.g. using a high- or low-interaction honeypot to detect malicious activity. Given a huge number of options, it can be challenging even for experts to choose which types of data are most important for the organization, and which methods of collection are most effective.

There is a very large number of potential external sources that can provide useful proactive security information. The access to some of them can be purchased (examples include commercial “threat intelligence” services and passive DNS providers), others require membership in closed trust groups (feeds provided by data clearinghouses or other CERTs), and many are available in the public domain (research communities providing data on botnet controllers for defense purposes, reputation services used for spam filtering).

Above and beyond of the fees actually charged by the data provider, the integration of each source – even sources available publicly – requires certain investments on the part of the recipient. The most obvious cost is the development or adjustment of internal infrastructure to ingest a new information feed. Fortunately, if technical characteristics of the new feed are similar to ones that were already processed by the recipient, integration will not require much effort. However, there remains a second major issue: the cost associated with finding and evaluating the sources of information.

2.1.4 Evaluation of data sources

Evaluation is a crucial step, since the independent assessment of properties of the received data (see section 1.3) allows the recipient to determine its quality and, ultimately, its actionability. CERTs should use the results from the evaluation to prioritize sources for integration on the basis of a cost-benefit analysis. The benefit of a source should be understood here not only as the quality of information provided, but also the quantity of information over time. The cost of a source consists of the fees of the provider (if any) but more importantly, the effort required to integrate and maintain the infrastructure to collect and process data. The decision criteria for whether a particular source is worth collecting are unique to each environment and more specific guidance in this regard will not be provided.

Unfortunately, a comprehensive evaluation is difficult, labor-intensive, and may even be impossible to perform in short term without actually integrating a data source and assessing its value operationally. Since CERTs often have limited spare resources that can be allocated to this task, it is common to rely on rudimentary manual checks, and on external opinions from the community to gauge the potential usefulness of the sources. Detailed inventories of sources published by ENISA [11][13] and other independent organizations provide reference material that is helpful for choosing information feeds. However in the end, a CERT should thoroughly verify and continuously monitor the quality of all of its data sources (section 2.4.5 contains more information on evaluation of sources).

If a CERT organization does not have enough resources to properly analyze collected information, it may not be able to properly estimate its accuracy and relevance. While it does not directly affect technical aspects of data processing, this uncertainty can have an adverse impact on the actionability of the information.

One-time reports are a special case since they may contain valuable information but a certain amount of analyst effort is required to process each of them. Usually it is not necessary to evaluate them as thoroughly as recurring sources that are collected and processed automatically, nevertheless some validation is generally required.

2.1.5 Recommendations

The following principles are recommended to be applied to the collection process:

- Take a proactive approach to identifying and evaluating potential external sources:
  - learn about the available feeds from reports published by independent organizations
  - evaluate the feeds before committing significant resources into purchasing and integration
  - first consider information freely available from open sources (e.g., reports published by vendors) and provided by exchanges operated by national CERTs or other entities coordinating cross-organizational sharing
  - prioritize feeds based on a consideration of both value and cost
- Even if data is not actionable at the moment of collection, consider whether it can be correlated with other sources to generate actionable information.

- Data, especially in the operational environment, can be collected and exchanged in many ways, so the collection infrastructure must be flexible and capable of accommodating various transport mechanisms.
- When considering adding a new high-volume source, first make sure that your current infrastructure will be able to handle the data.
- Consuming data of unknown quality may incur problems in the long term, so try to continuously monitor the existing sources.
- Automate the collection process as much as possible and carefully consider the effort required to collect and process unstructured data sources.
- Knowing the original source –or key information about that source –is important to build trust in the value of the data, even if that data was forwarded by a trusted proxy organization, like a national CERT.

2.2 Preparation

Once the data has been collected, it is transformed to make it more useful (or, in other words, more actionable) from the recipient's point of view. The most basic property that can be improved is ingestibility (see section 1.3), however, other properties can be enhanced as well.

2.2.1 Parsing

As it was noted in the previous section, information comes in multiple formats – some of them are standardized and supported by existing tools, while others are vendor-specific or created ad-hoc by the producer. However, in the end, for analysis it is only the meaning of data that is important and the format that was used to express it is irrelevant as long as it is possible to extract that meaning. This requires the development of parsing and normalization processes for each distinct input format. It is common practice to parse raw input in order to extract relevant elements of information – like IP addresses, domain and timestamps – and translate them into a normalized form for later use in the processing pipeline. Details of this step can vary and will depend greatly on the implementation of the whole processing pipeline (an example of such a transformation is presented in Figure 3).

Figure 3. Example of publicly-available indicator-level dataset published by Dragon Research Group[9] before and after parsing, normalization, and enrichment by IntelMQ. For brevity, only the first indicator has been shown on the output.

Input:

```
# Formatting is as follows:
# ASN|ASname|saddr|utc|category
#
174 | COGENT-174 - Cogent Communicat | 162.244.12.245 | 2014-07-03 19:21:12 | vncprobe
286 | KPN KPN International / KPN Eu | 188.201.136.73 | 2014-10-01 12:21:57 | vncprobe
174 | COGENT-174 - Cogent Communicat | 162.244.10.100 | 2014-07-03 04:53:57 | vncprobe
```

Output:

```
{
```

[9] Dragon Research Group: http://www.dragonresearchgroup.org. IP addresses are provided only as an example and the accuracy of the information was not verified.

```
    "feed": "dragonresearchgroup",
    "source_cymru_cc": "US",
    "reported_source_ip": "162.244.12.245",
    "feed_url": "http://dragonresearchgroup.org/insight/vncprobe.txt",
    "source_time": "2014-07-03T19:21:12+00:00",
    "taxonomy": "Intrusion Attempts",
    "source as name": "COGENT-174 - Cogent Communications,US",
    "source_ip": "162.244.12.245",
    "source_registry": "arin",
    "reported_source_asn": "174",
    "application protocol": "vnc",
    "source_bgp_prefix": "162.244.8.0/21",
    "type": "brute-force",
    "source_allocated": "2014-03-04",
    "source_asn": "174"
}
```


The following are examples of the parsing step based on implementations using two common frameworks for processing and managing log-like data:

- Logstash,[10] an open source log processing software, is often used to read log entries consisting of semi-structured text and convert them to JSON documents that have a predefined structure (a normalized form). This transformation is realized by a built-in engine that matches incoming logs against a list of preconfigured patterns. Later, the JSON documents are fed into other components responsible for storage and analysis: frequently Elasticsearch and Kibana are used for this purpose in a configuration commonly referred to as an "ELK" stack (where ELK stands for Elasticsearch,[11]Logstash and Kibana).
- Splunk,[12] analytics software which is also focused on log-like data, accepts any text-based input and preserves its original format. Extraction of information is defined on-the-fly through regular expressions, which allows fields to be extracted from entries at any time, as part of the ingest or retroactively at query time. Because the raw data is archived in Splunk, it is common to extract only these elements that are needed for a particular task. Note that Splunk was chosen as an example here, even though it is also commercial software (with a free limited version available) as it has arguably developed into a de facto industry standard that can be easily adapted to fit many of the scenarios explored in this report.

In the first example, data is fully normalized – an original entry is replaced with a structure that is used to represent this particular type of information (e.g., an access log entry) internally. This approach is used by many tools for management of security information, including AbuseHelper,[13]CIF,[14]CRITs[15] and MISP[16]and ArcSight.[17]

This contrasts with the second example, where the original data is preserved and can be re-parsed at any time. Full normalization is not required in this case and the original entry is augmented by the elements extracted through partial parsing. An important advantage of such an approach is its flexibility, since the way that the data is processed can be modified relatively easily. However, not many tools provide this capability, and configuration of general-purpose software like Splunk is

[10] See http://logstash.net
[11] Official website of Elasticsearch and Kibana: http://www.elasticsearch.org
[12] See http://www.splunk.com
[13] See https://bitbucket.org/clarifiednetworks/abusehelper
[14] See https://code.google.com/p/collective-intelligence-framework/
[15] See http://crits.github.io
[16] See https://github.com/MISP/MISP
[17] See http://www.arcsight.net

inherently more complex compared to specialized tools. Moreover, in order to extract certain element of information, one has to realize that it is present in the original data in the first place.

Sometimes a middle way is chosen where by fully normalized data is sent for further processing but the original input is preserved for reference. Several tools and data formats provide built-in capabilities for such an arrangement. For example,n6[18] keeps both raw files and streams of events in an archival database. Similarly, Megatron[19] preserves the original log lines as one of its database fields. However, even if the software used for processing data does not handle raw archives natively, almost any solution can be adapted to keep copies of unprocessed input in some way.

Keeping data in the original format has multiple advantages, even if a CERT is not using software capable of analyzing it on-the-fly:

- It allows a user to verify if the parsing was performed correctly (e.g., that no fields were omitted and correct encoding was used) if any problems arise at a later point in time.
- When sharing data with external entities, providing the original form can increase the confidence into the received report.
- If the normalized form changes (e.g., new elements are introduced),then existing data can be parsed again, which should guarantee that no information is lost during conversion (evolving normalized forms is discussed further in this section).

Naturally, keeping data in both formats introduces an overhead in terms of computational and storage resources, so it does not necessarily make sense to use this approach in all situations. Especially low-level data, which usually comes in large volumes, may not be worth retaining after the extraction of features of interest.

2.2.2 Normalization

While parsing may seem straightforward, the difficulties become evident when one has to perform normalization, that is, define how it is mapped into an internal data structure. There are two dimensions to this problem: heterogeneity of data and a lack of common ontology.

Heterogeneity is an inherent characteristic of information sources that are collected by CERTs. In this document the information is based on its level of abstraction (section 1.4). However, this categorization is coarse-grained and sources of the same level can provide data that is vastly different. For example, access logs from an HTTP server do not share many features with network flow data (except perhaps the IP address fields which are a sort of shared key field), and a vulnerability advisory is quite distinct from a description of a C&C channel. Heterogeneity is usually somehow limited by the fact that information of different levels or otherwise dissimilar is processed in separate pipelines. Nevertheless, developing a unified internal representation remains a challenge, since it must find a compromise between two opposing design goals:

- generality, which allows the normalization of a wide variety of information but brings complexity that increases costs in further processing steps;
- specificity, which makes further processing simpler and, consequently, cheaper, but limits the types of information that can be contained in the given representation.

The generality versus specificity trade-off is a well-known problem and it also must be tackled during the distribution stage (see section 2.5). Analysis of the trade-off in context of data exchange formats was performed by Mann et al. [14]

[18] See http://n6.cert.pl
[19] See https://github.com/cert-se/megatron-java

The second issue with normalization is the lack of a commonly used ontology for information security that can be used as a basis of the normalized form (see section 1.4 for more information on existing ontologies). At the time of writing this report the STIX language, which provides an informal ontology, is gaining widespread recognition within the community, so the situation may improve in the near future. However, until that occurs, the choice of a representation and semantics is somewhat arbitrary. The only hard requirement is consistency, since the formats and data structures are internal to the organization processing the data. Nonetheless, the lack of clear standards makes designing a sound normalized internal representation a challenge, since many details have to be agreed on and specified. For instance, the representation of an IP address may seem trivial, however an implementer needs to consider whether version 4 and 6 addresses should be kept separately or in the same field, or whether to use numeric values or dotted quad notation. Identifiers of malicious software (e.g., bots, remote access tools and trojans) are an example of a data element that is particularly hard to normalize, since no common enumeration exists and very often vendors assign different names to the same threats. Established taxonomies can be helpful in normalization of some elements, e.g. CAPEC[20] for types of attack or "Common Language" from Sandia National Laboratories [15] for categories of incidents, but in the end, an implementer faces a variety of trade offs that need to be made.

Ultimately, there is no single normalized form that is suitable to represent all, or even most, security information. STIX is a notable example of a broad-scope standard that can be used to define the semantics of a normalized form, however its primary XML representation can be inconvenient for internal use.[21] In practice normalization is usually applied within groups of similar data sources (e.g., alerts from network monitoring systems, or blacklists of malicious IPs and domains from multiple vendors). In this way, a CERT can standardize on a small number of normalized forms that make sense for the organization's processing infrastructure.

Off-the-shelf solutions usually impose their own internal data models, so if an organization does not choose to invest in development, it may be left with little choice in the matter. The origin of the data models used in existing tools can be roughly grouped into three categories:

- *Existing information exchange standards* provide relatively well-defined semantics and can be used as a basis of the internal model. Examples include CIF (IODEF), Microsoft Interflow[22] (STIX), and MANTIS (STIX and IODEF).
- If the goal is to use the same data model internally and for exchange with other entities, then some open-source projects start with a custom data model that was created internally, and then attempt to create a *generalized specification through a community-driven process*. Example: AbuseHelper ("data harmonization ontology" [16]).
- Many projects simply design their *custom data models* from scratch in a way that it is best suited to the architecture of the software. Separation of the internal normalized form from exchange formats allows to create a form that is optimized for a particular use case. Examples include CRITs and Megatron.

A potential problem that may occur is a sort of impedance mismatch between the formats that were originally used to represent information and the internal, normalized data model. This issue is more likely to appear when one attempts to normalize a diverse set of inputs into a single form or when dealing with structured inter-dependent data. The following example can illustrate such a situation: some systems used for managing security information (e.g., AbuseHelper and Splunk) use a data

[20] See https://capec.mitre.org
[21] According to information from MITRE obtained at the time of writing this report, alternative serialization formats for STIX are under development.
[22] Microsoft Interflow, Private Preview: http://technet.microsoft.com/en-us/security/dn750892

model built upon discrete events that contain a flat list of properties, each represented by a key-value pair. A major advantage of this approach is its simplicity, which makes further processing much easier by eliminating the need for complex parsing to extract values of interest (reduction of complexity even in automated processing should not be underestimated). It is common to have events that contain a property corresponding to a domain name. Since a DNS name may resolve to multiple IP addresses, they are also contained in events as multiple properties. However, problems arise when information regarding autonomous systems and countries the IP addresses are located in are stored. It is not obvious how to represent such a seemingly simple relationship using a flat key-value list. Basically, there are two solutions: drop some of the information (disregard which IP is located in which ASN and just add a combined list of all ASNs) or use a more complex data model.

The lack of a common way to normalize heterogeneous data is seen as a deficiency of the current generation of systems for management of security information. One of the experts proposed the development of a software platform with an "agile data model," which would be able to support and unify all standard data formats. This idea is realized in part in MANTIS, an information management tool that is able to ingest and normalize several structured data formats into its generic data model. [17]

Incompatible or incomplete taxonomies can be considered another aspect of the same problem. Incident reports exemplify this issue well. The classification of the same event can differ between CERTs, depending on methodologies they adopted. An intrusion through a vulnerable web site which exposes contents of a database might be called an "application compromise" by one CERT and "unauthorized access to information" by another. Since CERTs collect information from heterogeneous external sources, it can be assumed that there is always a risk of misinterpretation of some elements of information (type of incident, name of botnet, attack technique, etc.).

Although one-time data exchanges are most prone to problems with incorrect interpretation, there are risks even when dealing with recurring, established sources. It is not uncommon for various aspects of information collected from external sources to change occasionally. This includes changing a data format entirely, which forces recipients to expend effort adapting parsers. However, a more dangerous situation may arise when the overall format stays backward compatible, but certain characteristics of the data feed change. For example, a data feed that previously contained only IPs suspected of sending spam was extended to add IPs that are infected by malware. While the existing parser might accept the new data, it will be unaware that the semantics of certain entries has changed, so in the normalized form all IPs are incorrectly marked as spam sources, resulting in undesirable consequences in further processing steps. This is not a strictly theoretical risk, because many data formats are underspecified, so certain assumptions must be made during parser development.

2.2.3 Aggregation

Apart from parsing and normalization, several other transformations on the incoming data can be performed in this processing step. Some of the sources provide information that is more detailed than required, for example an access log from a web server can contain thousands of entries related to a single scanning activity. Processing of all similar entries may require non-negligible computational resources but each subsequent entry does not necessarily add much value. In such cases, similar events can be aggregated into a single one, that represents some activity as a whole. **Aggregation** can be built into the processing pipeline directly, e.g. n6 groups events related to network activity by time and a combination of other properties including network addresses. Alternatively, it can be integrated on the source level, which means that the collected data is already aggregated, e.g. Cymru[23] and

[23] See http://www.team-cymru.org

Shadowserver[24] provide lists of infected IP addresses aggregated by day. Aggregation is used mostly with high-volume sources of low-level data, since information of higher levels is usually already summarized in some fashion.

2.2.4 Enrichment

Parsing, normalization, and aggregation improve the ingestibility of the information, but other properties can be enhanced as well. Enrichment is a process of adding additional context to existing information, thus increasing completeness. From the technical perspective, enrichment is realized by correlation with multiple databases using various elements of the collected information like addresses and identifiers. These databases can be internal to the organization or access to them can be provided by an external service. The following correlations are implemented in many existing capabilities:

- If an incoming report contains domains, they are resolved to determine on which addresses they are currently hosted;
- past relationships between IP addresses and domains are stored in passive DNS [18] databases, which make them a very useful source for enrichment;
- countries and autonomous system numbers are determined for IP addresses through geolocalization;
- contact addresses and other information is obtained from WHOIS databases;
- reputation services are consulted to determine if an address is known to be malicious.

National CERTs often maintain databases that map IP allocations, autonomous systems and domains to their constituents. When dealing with large amounts of data, integrating this source into the automated enrichment process is essential for determining which entities were affected by threats. Constituent databases are built into some information management tools including Megatron and n6. If the currently deployed software does not have such a feature, then the external contact databases can be integrated to realize this function. A recent example of a data source that can be adapted for this purpose is the open source ContactDB[25] project, which is being developed by several European CERTs.

Similarly, if a CERT has visibility into protected assets (e.g., software that is used in the organization, services running within a network and their importance), it can leverage this information to further enrich incoming reports. A common use case is ranking severity of vulnerability reports based on their impact on a specific infrastructure.

The accuracy of the information can be improved at the preparation step by performing preliminary **quality assurance** and data cleanup. Possible actions include the following:

- Verification that data elements are well formed, e.g., URLs have valid syntax. It may be performed during parsing, but rules should be kept identical for all sources.
- Artifacts resulting from deficiencies in the collection method that are not worth analyzing (i.e., “noise”) can be filtered out. Example: a honeypot reports all network traffic that it received as suspicious; since it includes packets that are echoes of DoS attacks (replies to a spoofed IP addresses), during preparation such entries are detected and discarded.
- Whitelisting can be employed to filter obvious false positives. For example, certain IP addresses or URLs (e.g., Google) can be used by malware for checking internet connectivity; these addresses might be incorrectly reported as being command and control servers,

[24] See https://www.shadowserver.org/wiki/
[25] See https://github.com/certtools/contactdb

however by matching incoming reports against a whitelist, it is possible to detect some of the false alerts.

- Source-specific heuristics ("sanity checks") can be applied to verify if a particular report has features within predefined limits. For example, files that are orders of magnitude bigger than usual or near-zero size may indicate technical problems with the feed, as are reports containing duplicated information.

Most of the existing tools offer very limited support for data cleanup – usually a limited set of elements are verified for syntactical correctness. However such checks are usually quite easy to implement via simple scripts or other mechanisms that work with already deployed software. Naturally, data cleanup must be done with care, as certain invalid data elements which can be manipulated by an attacker are not errors but features of the threat. For example, malware might use malformed DNS names to evade detection, so an "incorrect" name must be preserved as is.

2.2.5 Automation

Most of the actions in the preparation step are automated, even if data was collected in a manual way. In cases where data is provided only once, and in a particular form (e.g., a non-recurring source such as logs attached to a single incident report), it is usually more effective to adapt existing automated tools (e.g., prepare a parsing script, configure an import module) than to deal with it manually. Since inserting one-time reports into the main processing pipeline takes effort, CERTs often consider the cost of integration too high, and data such as this is handled on an ad-hoc basis. Given the limited resources of many CERTs, such an approach is understandable, nevertheless it has significant drawbacks – existing infrastructure will not be used to facilitate analysis and distribution of this information, so it will often end up being underutilized.

The choice of systems used for management of information plays a huge role here, since their capabilities determine how easy (or how difficult) it is to integrate a new data source. Systems that require creation of a parser, even a simple one (AbuseHelper, typical SIEMs), obviously require more effort than ones that need just a minor reconfiguration (e.g., Splunk).

The work associated with the integration of new sources, regardless of their recurrence, depends greatly on the degree of their similarity to other sources that are already supported by the processing pipeline. If the type and format of data provided by the new source does not differ much from an existing source, integration is straightforward and usually requires only small adjustments in parsing that were discussed above. In contrast, adding a completely novel source is more challenging.

In the worst case scenario, the type of information is not supported at all by the tools used in current processing pipeline, which means that the processing infrastructure needs to be reworked (existing pipeline redesigned or a new one created). Many information management tools including Megatron, n6, AbuseHelper, and CIF are focused on processing indicators and cannot accept data of other levels (e.g., PCAPs or vulnerability reports).

Nevertheless, an incompatible source can usually be adapted to the existing processing infrastructure at the cost of a partial loss of information. It can be achieved by replacing basic parsing with the selective extraction of only those elements from the input that can be normalized to the form already accepted by currently deployed tools. In this arrangement, information that cannot be normalized is simply left out.

Consider a scenario where reports from a sandbox contain a significant volume of behavioral data related to a collection of malware samples. Some elements of the report like IP addresses and URLs contacted might be immediately used as actionable indicators, while other ones like the names of files dropped or system calls will require further analysis to draw any conclusions and produce indicators.

Such a source can be integrated with AbuseHelper or a similar system by extracting only indicator-level data that is similar to other types already accepted by the software and omitting the rest. This solution, while imperfect, allows the automatic handling of at least part of the available data, without redesigning the entire processing infrastructure.

2.2.6 Recommendations

- If data is considered valuable in the long term (months or years), it usually is worth to keep it in the original format.
- Normalization can simplify further processing but must be applied with care. It does not always makes sense to normalize different types of data into a single form. Integration of heterogeneous data might be feasible when using ontologies that allow to describe a wide range of entities.
- Data obtained from one-time reports should be integrated with the rest of the collected information, if feasible.
- Information should be enriched through correlation with multiple internal and external databases, particularly DNS (including passive DNS), asset databases within the organization, abuse contact databases and reputation services.
- As enrichment does not always provide entirely accurate information, consider distinguishing primary data from elements imported from external databases. With dynamically changing data – like DNS – storing the exact time when the enrichment took place might be useful during further analysis.
- Preliminary quality assurance and data cleanup should be performed, although data elements that can be manipulated by an attacker must be preserved as-is.
- Output of existing parsers needs to be monitored to detect unexpected changes in the semantics of the incoming data. Since a source can re-use existing syntax, additional verification is required to catch such changes, since the parser will continue to work without reporting errors.
- This processing step should be fully automated and the selected tools should be easily adaptable to new data formats.

2.3 Storage

The choice of a data storage approach may seem like a minor technical detail at first, but in practice it is an important part of the processing pipeline and requires careful design. The choice of a storage technology should not affect any properties of the information itself but it will have a significant impact on the implementation of the analysis and distribution processing steps. In this section will be discussed various factors that have to be taken into consideration when designing a repository for collected data, and their implications for the way the information is processed.

A CERT must first decide whether it should use off-the-shelf software for this purpose or attempt to build its own solution. The choice of software for information processing will dictate many aspects of the storage solution, which may be not optimal in a particular environment. A notable exception is AbuseHelper, which does not have any storage backend by default. However, building and integrating a custom storage backend, even if it is based on existing software components like general-purpose database engines, takes a considerable amount of effort, so in many cases the right choice may be not obvious.

Many considerations in this section apply regardless of that choice. Even if the CERT decides to use one of the existing systems with an integrated storage backend, knowing about the advantages and limitations of different technologies should be helpful in making an informed decision in this regard.

2.3.1 Retention time

One of the most important design decisions for a data repository is determining how much history needs to be kept. In some cases, there will be data that need only be preserved until it is analyzed and distributed. At the other extreme, there may be data that should be archived for long periods of time in order to support long-term statistical studies and longitudinal analysis.

In general, it is better to preserve data. Historical data can have many uses, both in the operational and analytic contexts and often its value is only clear well after the data was collected. Nevertheless, there are two factors that limit the retention period, and which need to be considered:

- **Legal issues**. If data contains personally identifiable information (PII), as is often the case with network traffic captures, local laws often define a maximum time after which the data must be permanently deleted.
- **Technical issues**. Data may require significant resources to store, and query performance can sometimes be impacted by data volumes. Available storage will sometimes create a need to prioritize what data is preserved. Since the usefulness of data generally declines over time it will usually be possible to identify data that is not worth storing for extended periods. For example, after extracting relevant information from raw network traffic, it is often not necessary to keep the original PCAP files.

2.3.2 Scale

The storage solution must be able to cope with the following scalability requirements:

- keep up with writing the incoming data without introducing additional delays, thus preserving timeliness;
- store data for the chosen retention period;
- provide read access to archived data with adequate performance.

Depending on the source, the volume of incoming data may range from near-zero (e.g., a onetime incident report, or an infrequent series of alerts) to terabytes per day or more (e.g., dumps of the raw network traffic collected at the perimeter of a large enterprise). In cases when there is little data to process – up to an order of magnitude of dozens of megabytes per day – performance and disk space requirements are not an issue. However, when dealing with high-volume sources, storage may become a bottleneck in the processing pipeline.

Since CERTs tend to add additional data sources over time, they should be certain that the solution they have chosen is scalable. Theoretically speaking, many systems provide linear scalability, that is, their capacity and performance of the system should be proportional to the computing resources allocated. Unfortunately, this is difficult to achieve in practice, and storage backends that scale poorly can be a source of problems in the long run.

In most cases the volume of the data is inversely proportional to its level:

- **Low-level data** generally comes in large quantities, however once the actionable information is extracted from it, its value drops significantly. Often there is no need to keep it for extended periods.
- **Indicators**, even large sources like sinkholes, when aggregated (see previous chapter) tend to generate no more than several records per infected machine daily and most botnets are in the range of thousands to tens of thousands infections. [19] The amount of indicator data coming from other types of sources is usually significantly lower – at least an order of magnitude in our experience.

- **Advisories and strategic reports**, when compared to previous levels, generally impose negligible storage requirements.

If the original input data is stored along with the normalized form, disk space requirements can easily increase, even by an order of magnitude. This is the main argument against archiving original data for high-volume sources.

2.3.3 Dataset management

When working with multiple sources, efficient management of archived information becomes one of the critical issues. Each source introduces its own data management requirements, which it's referred here as dataset management. [20] Without appropriate procedures in place, it may become increasingly difficult to keep up with the growing complexity of a data repository. This is especially evident when storing information coming from one-time sources.

In order to have a good understanding of the information that has been stored in the repository, metadata – additional information regarding the stored data that is not part of the data itself – has to be kept for each distinct dataset. The exact structure and content of the metadata will depend on the processing infrastructure and procedures that are specific to a particular organization. In many cases it may be sufficient to maintain a textual description of each data set, while in others it may be appropriate to structure the metadata to support automated processing. Some of the characteristics of a source or a dataset that can be included in metadata are listed below:

- when and how the dataset was collected by the CERT;
- what transformations were performed during the preparation step;
- whether it contains sensitive information that requires special handling;
- how the dataset is intended to be used internally (e.g., a data set may be tagged as an input to an annual report);
- any extra information that can be helpful for analysts when working with a particular datasets, including notes about accuracy or other properties of the data.

Producing metadata takes some effort, which is unnecessary when working with a small number of well-understood sources. However, once the number of sources reaches dozens or hundreds, having a "knowledge base" describing collected datasets can be invaluable. Existing off-the-shelf systems have limited capabilities for dataset management, therefore external tools are frequently used alongside these systems to manage the metadata. These can be general-purpose documentation capabilities like wikis or custom-built software applications.

CERTs are presented with a variety of challenges when handling sensitive information. In addition to ensuring that the information handling rules are followed in later stages of processing so that the information is never inappropriately distributed, there are implications for storage in order to ensure the security of the repository. First, appropriate access control mechanisms need to be in place to ensure any access restrictions for internal personnel are enforced. That requires the translation of any labeling into access control policies implemented in the system. Suitable mechanisms also have to be deployed to ensure integrity and confidentiality of information, even in the face of unexpected problems like hardware failures. After the retention period, data must also be erased in a secure way. The specific techniques that should be used for that purpose (e.g., encryption and backups) depend on the technologies used, including the underlying operating systems and hardware, and are outside of the scope of this paper.

As a CERT's information handling requirements become more complex, the need for well thought out, structured approaches to tagging datasets with metadata may become necessary, but in the meantime simply having mechanisms for associating notes with datasets may be sufficient.

2.3.4 Technologies

The choice of technology for the data storage backend has profound consequences for scalability and imposes constraints on the implementation of integration with off-the shelf or custom-built software. Databases, and data storage in general represent huge topic areas in computer science and engineering. A detailed discussion of technical considerations is therefore outside of the scope of this document. Instead, the database technologies and their use for security information management will be briefly characterized.

Relational Database Management Systems (RDBMS) remain the most established database technology across a variety of applications. In large part this is thanks to SQL, which provides a common, standardized way to interact with database systems from multiple vendors, enabling complex queries on structured data. There are many tools and software libraries for interacting with database systems using SQL. Relational databases are a mature, well-understood technology, with a wide choice of both commercial and open-source solutions to choose from. It should come as no surprise that many existing systems for management of security information use SQL databases (e.g., Megatron, n6, CIF, MANTIS, MISP). Nevertheless, RDBMSs have certain disadvantages, for example, under some circumstances it can be difficult to scale an RDBMS to meet the needs of applications that require very high ingest rates of semi-structured data. This has led to the development of so-called "big data" solutions including Hadoop and a variety of other NOSQL technologies, which are considered below. The relational model itself imposes some constraints on how data is represented since it requires that a fixed data schema be defined a priori, with the consequence that an RDBMS may be not optimal for representing certain types of data where the structure of that data may not be known at initial ingest time.

A set of alternative database technologies collectively known as "NoSQL" databases have been developed over the last decade. "NoSQL" is an ambiguous term covering a very diverse set of technologies. What "NoSQL" databases have in common is that they use different data models from the structured, schema-based relational model, instead relying another often more open structure for representing data optimized to particular query patterns (e.g., relying on simple key-value schemes, keyword-based indexing, or graph representations). Some write data onto disk, while others are in-memory, and they differ significantly in their functionality, with some offering relational-like features – like transaction support, indexing, SQL-like query languages and schema support – while others depart considerably from that model. Their back-end technologies are also typically based on horizontally-scalable cluster that resembles the architecture introduced by the Google Big Table [21] implementation that influenced the development of Hadoop.[26]

From the developer's point of view, the framework provided to define the data model of an application can be considered as the most important feature of a data store, but in practice information in almost any form can be mapped to the model used by a particular backend. For example, consider the object-relational mapping that is part of many popular software frameworks for building object-oriented applications on top of relational databases. In general, better performance can be achieved by choosing a database that is tailored to the particular application (its data ingest rates, query access patterns, etc.), but it is impossible to say if the gain will be significant without detailed analysis of how a particular system works.

When dealing with large volumes of data, the crucial aspect of a data store is its scalability (see earlier part of this section), and this is where some of the NoSQL solutions stand out. Many of the databases in this category offer good horizontal scalability, which allows the storage of terabytes or petabytes of data on large clusters of commodity servers. For some applications this capability is a must-have,

[26] See http://hadoop.apache.org

but many others can easily work for years without ever needing to scale out to a degree that requires the added complexity of a cluster. NoSQL databases are already a popular choice for open-source tools, for example CRITs, MalCom use a document database – MongoDB[27] – as their backend.

A simple alternative to a complex database system is storing data directly in files using features of the file system itself to impose structure. This is sometimes referred to as a “flat file database." With such an arrangement, standard system utilities, like grep or awk, can be used to access the information. Certain data formats are better suited to such storage. Unfortunately, such simplicity comes with a price. It is difficult to query the repository without writing additional layers of query utilities and the representation of relationships between data records can be problematic. Frequently, a file-oriented approach is used for storing the original raw input data and for archival while the parsed version is inserted into a proper database where it can be more conveniently processed.

Due to its popularity, Apache Hadoop[28] requires a special mention. It is not a single tool but a whole open source platform “that allows for the distributed processing of large data sets across clusters of computers.” The foundation of the Hadoop ecosystem is a distributed file system (HDFS) and a framework for distributed analysis. A variety of database technologies have been built on this foundation, including NoSQL databases (Hbase, [29] Cassandra, [30] and many others), and as well databases that support SQL (Hive, [31] Spark SQL, [32] etc.). A variety of analysis, visualization, management, and application development tools have been built on top of the core technology. This includes tools and libraries to facilitate data analysis, including frameworks for statistical analysis and for the development of machine learning algorithms.

NoSQL solutions, including ones related to Hadoop, are being developed at a very fast rate. As a consequence, some software may be unstable or lack documentation, and it is not uncommon for projects to merge or become obsolete. Keeping track of the whole ecosystem is not trivial and requires an ongoing effort.

2.3.5 Recommendations

The following are our suggestions and considerations related to storage:

- Before committing to the development of your own software carefully consider your requirements and goals in regards to:
  - scalability
  - security
  - performance
  - management options
  - ease of querying
- When using COTS or existing OSS solutions, evaluate whether their default storage backend fulfills all of the requirements
- When using COTS/OSS for processing, consider tools with built-in facilities for archiving less frequently accessed data to eliminate the need for separate data repositories

[27] See http://www.mongodb.org
[28] See http://hadoop.apache.org
[29] See http://hbase.apache.org
[30] See http://cassandra.apache.org
[31] See https://hive.apache.org
[32] See https://spark.apache.org/sql/

- Management overhead of Hadoop or other cluster-based technologies becomes less of an issue if multiple applications can share the common infrastructure, so evaluate if such an arrangement is possible in your organization
- Before undertaking the development of a custom data storage and management solution, carefully consider the knowledge and level of effort required for development, maintenance, and management of the solution
- Tailor the data retention policies to each dataset, considering both the legal constraints on retention and the impact on storage resources
- When storing data from multiple sources, keep metadata that can be useful to manage it in the long term, including information about its origin and quality, how was it processed, and any observations regarding a particular dataset that should inform how it is interpreted.

2.4 Analysis

Once the data has been collected, prepared and stored, there is an opportunity for the CERT to do further analysis of the information before it is distributed to its final recipients. Analysis is not strictly required for information processing – under some circumstances information may be passed directly to constituents. However, there is often an opportunity to gain additional insight into threats by fusing data from multiple data sources. It may also be possible for a CERT to provide more relevant information by investing analysis time in contextualizing information for its constituents.

2.4.1 Fundamentals

In the most general terms, the analysis step as a process is defined, that takes collected and prepared information as the input and produces new conclusions. In contrast to enrichment, which is a part of the data preparation step, analysis is about deriving new information beyond that context that is explicitly linked to the original data. For example, DNS names might be resolved IP addresses that could then be used directly in access control lists. This is an example of a simple but valuable analysis step that can be done as part of the initial enrichment step. A more complex analysis could be done to enhance a DNS name with additional context by combining data from reputation services, passive DNS, and other sources to determine if a domain name was used by malware. It is worth noting that the distinction between preparation and analysis is often arbitrary: some initial processing to make these sorts of associations might be reasonably called “preparation” rather than analysis.

The input to this step of analysis will frequently not be directly actionable. In fact, increasing the level of information generated is one of the most important gains that can be achieved during the analysis: actionable indicators can be extracted from low-level data, and strategic reports for executives may be generated from vast amounts of indicator-level data.

A good example of obtaining actionable information from low-level data is using full packet captures of network traffic to develop an indicator. Analysts often execute malware samples in a sandbox and then analyze their network activity to identify IP addresses of C&C servers that can be used as indicators for differentiating malicious traffic associated with the malware from other connections.

A recent guidebook for security teams (Zimmerman [22]) contains a comprehensive list of capabilities that security teams (including CERTs) may provide. Among these capabilities are several that are essential to the analysis step.

Two of the activities described in this guide are fundamental to all types of analyses:

- Cyber Intel Creation – authoring new threat notices, indicators, etc., based on research performed by the CERT; it puts the CERT in the role of a producer of actionable information, instead of being just a consumer or a proxy for information published by other sources.

- Cyber Intel Fusion – a CERT can leverage its position as an analytical center to synthesize data coming from multiple sources and generate new actionable information based on existing data.

The following relate primarily to incident investigations (see section 2.4.2):

- Malware and Implant Analysis – reverse engineering of malware samples in order to determine their functionality, infection vector and characteristic features.
- Forensic Artifact Analysis – examination of low-level data (network traffic, memory dumps, etc.) to establish facts relevant to a particular investigation.

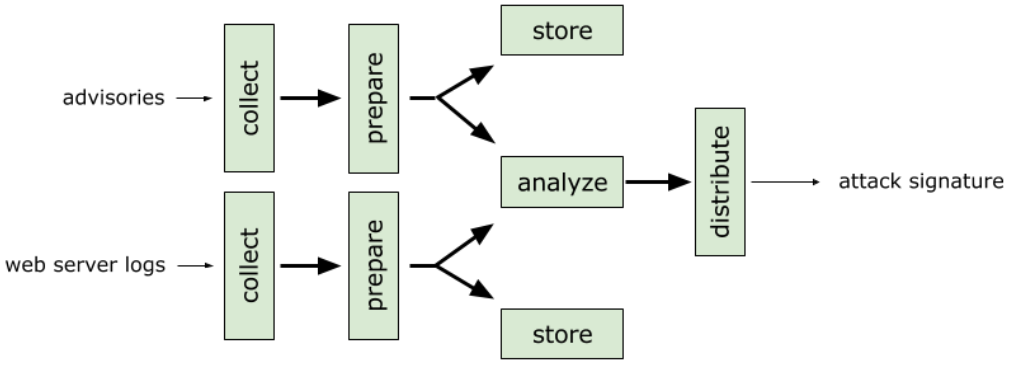
The *Trending* capability described in the guide corresponds to the long-term analysis of threats and defensive measures, which are discussed in section 2.4.4.

Finally, *Threat Assessment and Tradecraft Analysis* (studying techniques of attackers) are the basis of external situational awareness (see section 2.4.3.3).

These activities are considered in the specific context of the analysis step of our information processing model, but more generally they represent crucial capabilities of any CERT that aims to maintain a proactive posture.

Data fusion refers to the process of combining information from multiple sources, at different levels, for analysis in order to reveal relationships that would otherwise remain hidden. In terms of the conceptual processing model proposed at the beginning of this chapter, it may correspond to a situation where multiple pipelines merge in the analysis step. The diagram below (Figure 4) illustrates such an approach: web server logs from a honeypot are combined with a vulnerability advisory to create a new web application firewall (WAF) signature (an indicator) that identifies attacks on the service.

Figure 4. Merging multiple processing pipelines in the analysis step.



It is almost impossible to list all of the methods and tools that are used for analysis. The attempt is to highlight general approaches to systematic analysis of information by CERTs. It is worth pointing out that due to its complexity, this processing step is difficult to automate. Full automation (i.e., no human intervention required other than occasional maintenance) is possible only for a limited set of well-defined situations. For example, a simple process can be implemented such that, upon receiving a report that one of the hosts on the network is sending out spam, an analysis script is run that verifies if the reported IP address indeed showed high SMTP activity recently. The result of that check can then be used to trigger an action to automatically block that particular host. Nevertheless, in most cases, automated systems facilitate analysis but it is a human that draws conclusions and decides what

is the final product of this processing step. Systems developed to support this activity should be designed with this decision-making workflow in mind.

2.4.2 Investigation

Investigative work is undoubtedly a core activity of CERTs. What exactly is investigated depends on the particular situation – it may be a potential intrusion, a phishing campaign, a botnet, or any other threat in some way relevant to the constituency. While the focus is on the analytical aspect, the processing of information in the course of an investigation often includes all of the steps introduced in the pipeline model.

Most importantly, investigation will often deliver new meaningful and actionable information, such as indicators discovered through malware analysis, or a vulnerability announcement following research of a compromised system. As this is often the only way to recover such information – it may be impossible to describe the domain generation algorithm used by a malware sample without reverse-engineering its code– the benefits will often offset the effort invested.

2.4.2.1 Overview of investigative work

Regardless of the scope of an investigation – a single incident or a problem affecting a large number of constituents – the usage of information during analysis follows certain common patterns, listed below:

- Initially, an analyst needs to decide which data to analyze and in what order: data that will most likely contain relevant information, and is most accurate or complete will be analyzed first.
- In the course of an investigation there often arises a need to collect additional data. Typically it is performed on an ad-hoc basis (e.g., memory or disk dumps are obtained from an infected machine), but sometimes more permanent collection tools are deployed (e.g., a honeypot to catch activities of attackers that already have presence in the defended network).
- Virtually any investigation requires the use of data from multiple sources, so correlation of information is an intrinsic part of the analysis.
- Multiple queries to internal and external data repositories need to be performed to gather all necessary information. Consequently, these systems must provide an adequate query interface and have sufficient read performance.

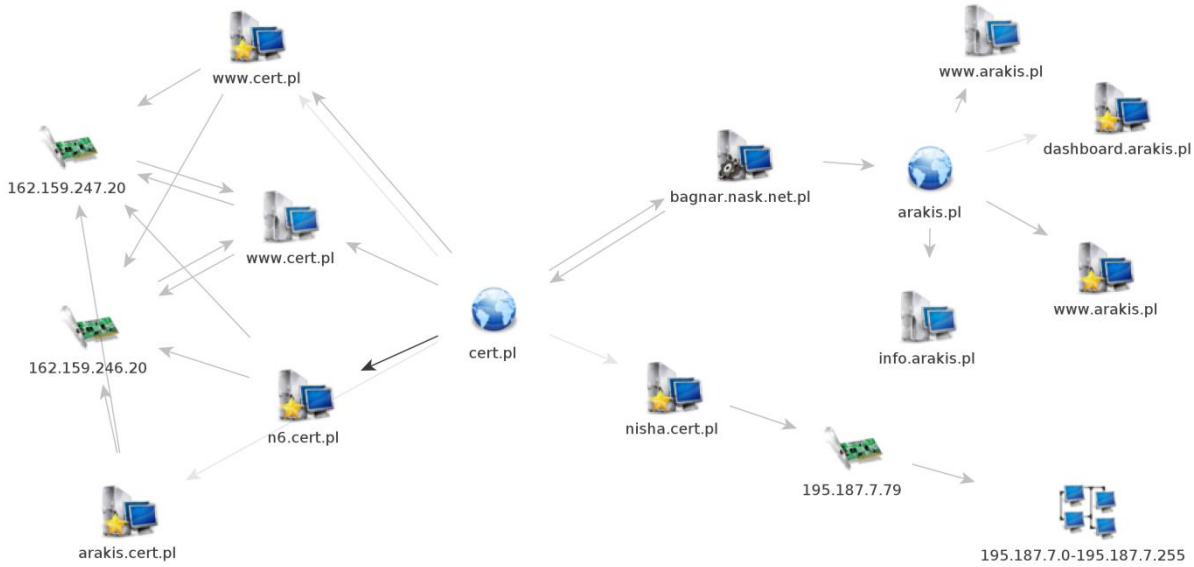
Correlation is the heart of investigative work – it allows an analyst to understand the context of observed activity and discover new facts based on the initial information available. One can also correlate data from multiple sources to see if their classification of a particular entity (website, IP address) or event is consistent, which can also be considered as a form of verification of uncertain reports. This is the approach applied by VirusTotal[33] and similar services for the comparison of results of multiple antivirus engines. In terms of properties of information (see section1.3), correlation can be used to increase completeness and accuracy.

Utilization of graph-based visualization techniques can significantly increase productivity when correlating data from multiple sources. Several commercial tools – Maltego[34] (see Figure 5) or Palantir[35] – and open-source ones – MalCom, CRITs, STIXViz[36] – provide an interactive visual

[33] See https://www.virustotal.com
[34] See https://www.paterva.com/web6/
[35] See https://www.palantir.com
[36] STIXViz is an open-source proof of concept tool for dynamically visualizing relationships between elements of information expressed in STIX, seehttps://github.com/STIXProject/stix-viz

representation of relationships between various types of entities that are being investigated which makes it much easier for an analyst to spot patterns and identify interesting elements. A practical example of using correlation is described in the case study ”Using indicators to enhance defense capabilities. ”

Figure 5. Graph-based visualization of network entities in Maltego.



In general, the tools used by a CERT should minimize the time that analysts must spend analyzing a particular case. Therefore the software should make data access as easy as possible – for example web based graphical interfaces allow to provide access regardless of the client platform used, and command line interfaces are preferable whenever there is a need to automate some tasks.

2.4.2.2 Triage and results

Traditionally, an investigation is launched when an incident is reported or the CERT detects a potential intrusion through its own monitoring capabilities. A typical incident report is in the form of a textual description of a problem (i.e., an advisory), optionally supplemented by reference data like system logs (i.e., low-level data or indicator-level information). If a CERT detects a threat on its own, it is usually through automated monitoring systems like an intrusion detection system (IDS) or anomaly detection systems, which generate indicator-level information (e.g., an IDS alert pointing to a specific TCP transmission) or advisories (e.g., unusual activity by one of the users).

In such circumstances, the CERT plays a reactive role. When an analyst responds to the incident, the ultimate goal of the analysis is to determine how to mitigate the current threat and decide if any actions should be taken to effectively defend against similar attacks in the future. According to best practices, the analysis should start with triage, [23] which provides a framework for prioritization of incidents to decide whether the incident should be handled immediately, queued for later, or simply rejected. In order to perform triage, an analyst determines the severity of the incident in the context of the constituency. For a large ISP, a report regarding one of its individual clients might not get much attention. But when a part of a business-critical infrastructure is being attacked, the report will certainly be handled without delay. To estimate the severity of an incident, the analyst must know what is being affected, so he or she must correlate information in the incoming report (e.g., a victim

IP address) with data sources containing a catalog of resources within a CERTs constituency (e.g., an inventory of servers and their importance to the business).

The development of an investigation depends on the particular threat and environment that the CERT operates in. In general, a potential threat must be verified: the perceived accuracy of information must reach a sufficient level. This can be done by cross-checking facts in sources of low-level information like logs, NetFlow records, and other host and network data. In some situations the available data will be insufficient for verification and additional actions will need to be taken, including, for example, forensic analysis of the potentially compromised host.

Once the threat is confirmed, there are important questions that should be answered during analysis, including the following:

- What is the overall impact of the threat on the constituency?
- Are the events and entities that we examine unique, or are they a part of some larger campaign?
- What information is still missing?
- What should be done to eliminate the threat, or is it possible at all?

Formulating answers to these questions requires putting known facts into a larger context, which means that the completeness of gathered information must be adequate. That is where correlation plays the most important role.

An investigation can yield a variety of information that can be used to mitigate the current threat and similar ones in the future. In the case of an intrusion, it may be possible to identify the C&C infrastructure (IP addresses, DNS names), malware (hashes of files, IoC extracted from infected hosts), and even TTPs of attackers (e.g., common elements of spear-phishing campaigns, targeted resources). When facing sophisticated adversaries, it might even be possible to identify previously unknown vulnerabilities that were used to exploit systems within the constituency. Since analysts within the CERT itself produced the information, it is usually accurate, complete and, in consequence, actionable.

When investigating an active threat (e.g., an intrusion in process), time is a critical factor. Actionable information that can be used to block an attack should be produced as quickly as possible – the tools available to a CERT should enable analysts to work efficiently toward that goal. The manual effort required for querying data sources, correlating, sharing data with other members of the team, and other common actions should be minimized. A simple example of improving efficiency with tool support is the automatic creation of new tickets in the issue tracker whenever information requiring action is received via email or another channel. Minimizing the analyst overhead required for these activities should be a goal when choosing and customizing tools used by analysts.

2.4.2.3 Exploratory analysis

An investigation can also be initiated by the CERT itself, without waiting for an external report. Such a proactive approach is used for research and exploratory analysis that allows to obtain better understanding of the environment, constituency, and potential threats.

The application of the proactive approach for finding malicious activity within a network is sometimes referred to as Hunting operations. [24][25] Analysts can use their knowledge of the protected assets, normal behavior and security controls in the organization to search large repositories of low-level data (e.g., application logs or network traffic records) and identify suspicious activity. Outliers (anomalous activities) may reveal threats that were not detected by an automated monitoring systems. For example, network traffic analysis might identify the C&C activity of a new piece of malware for which there is no existing IDS signature.

This type of analysis is typically performed iteratively. At each step actionable information regarding threats that were identified is extracted. Frequently, this information can be used to automate the detection and/or mitigation of similar activities in the future – see section 2.5.1.1. But even if further investigation of a suspicious activity proves that it was legitimate, the analysts will have gained a better understanding of the protected infrastructure.

Cisco CSIRT uses the name "playbook" to describe a proactive approach to analysis – "plays are self-contained, fully documented prescriptive procedures for finding some sort of undesired activity." [26] A play contains instructions for an analyst to construct a query against data repositories storing low-level information and indicators, and guidance regarding the interpretation of results. A playbook undergoes constant evolution – plays are updated, added and removed as a CERTs knowledge about the constituency increases.

Exploratory analysis is also a fundamental component of a forward-looking approach to security. While providing less of immediate practical value, such research might give insights into new types of threats, and it is an opportunity to develop core technical competencies of the team – something that might prove valuable in future operational work. Moreover, conclusions from research, even if not directly applicable within the constituency, could turn out to be useful for wider community. For example, anomalous scanning activity detected by a honeypot might signal the rise of a new global threat.

2.4.3 Situational awareness

The term situational awareness will be used in accordance to definition provided by the U.S. Committee on National Security Systems: [27]

> *"Within a volume of time and space, the perception of an enterprise's security posture and its threat environment; the comprehension/meaning of both taken together (risk); and the projection of their status into the near future."*

In this usage, situational awareness corresponds closely to the concept of a strategic "threat assessment" – a capability defined by Zimmerman as a "holistic estimation of threats posed by various actors against the constituency, its enclaves, or lines of business, within the cyber realm." [22] It was already mentioned that understanding the larger context is important during any investigative work, however here it is indeed the crux of the matter. Overall, achieving a good level of situational awareness for a CERT means that it has an understanding of the security posture of its constituency and it is able to identify the most important threats to that constituency, their key characteristics and, at least to some degree, predict likely developments in the near future. In other words, it can be viewed as a continuous monitoring of threats.

2.4.3.1 Overview of situational awareness

A CERT that maintains situational awareness is better prepared to handle incoming attacks, malware outbreaks and other security problems by guiding both the development of preventive measures and the remediation process. Additionally, early warning services, which can be considered a part of situational awareness, allow a CERT team to spot new threats quicker and take appropriate remediation steps before any serious damage is done. The exploratory analysis described in the previous section contributes to situational awareness, however here a more systematic approach will be considered, where analysis and building situational awareness are parts of a continuous process.

A CERT can assess its level of situational awareness by attempting to answer important questions regarding the threat environment. Some examples are included below:

- A national CERT would like to know which botnets are currently spreading most quickly and what are the most frequently used infection vectors.
- It could be very useful to know how bots are used by criminals and what are the consequences for owners of the infected machines.
- A significant issue for an internal CERT can be knowing the current patch status within the enterprise or the effectiveness of deployed security controls.

Answering many of these questions requires the performance of a risk assessment that depends on a CERT having knowledge of the key resources that are part of its constituency. Such knowledge is also a requirement for the proper triage and investigation of incidents.

Malicious activity usually corresponds to a small fraction of the data that is collected by a CERT and the flood of information rules out manual inspection. It is simply not possible for a human to identify and investigate all of the unusual behavior that might indicate an intrusion from the network traffic of the thousands of devices on a corporate network just by looking into network flow records. Automation is a critical part of supporting analysis for situational awareness.

Of course, it is not possible to create situational awareness entirely through automated analysis. An automated system can produce qualitative and quantitative information on output but interpretation requires a human analyst. In the business context, the term “operational analytics” is sometimes used to describe such an approach. Quantitative statistical information can provide solid grounds for decision making. For example, when mitigating a DDoS attack, knowing the proportion of malicious to benign traffic for multiple autonomous systems may be very helpful for planning a filtering policy. Obviously quantitative data is not always available and many systems simply report suspicious events without providing quantitative evidence that can be weighed directly by an analyst – for example, software that monitors behavior of users and reports when current activity deviates from a baseline profile may only generate an alert.

Therefore the role of an analyst is still essential. An analyst will leverage automated tools to summarize and find interesting elements in large data sets, but then use his own judgment to formulate conclusions. These conclusions may have various forms, for example:

- The inspection of network traffic associated with attacks might allow an analyst to identify a new exploit in the wild leading the analyst to publish a vulnerability warning (advisory).
- An automated system might flag IP addresses that are associated with multiple instances of suspicious activity, leading an analyst to associate those IPs with infrastructure operated by specific malicious actors, and eventually to add those IPs to blacklists (indicators).
- A network traffic analytic might alert on an unexplained surge of traffic from a server that requires further inspection, leading the analyst to examine the associated PCAP file (low-level data), and ultimately create an incident report that includes that data.

Consequently, the systems supporting situational awareness can yield actionable information that can be used by the CERT to mitigate attacks or other threats and shared with external organizations. Moreover, there is a feedback loop – information obtained from threat monitoring can trigger investigations (further analyses) and knowledge gained from these investigations can be applied to improve situational awareness.

Information of almost any type and level might be in some way useful for establishing situational awareness and each type of information can be analyzed in various ways. The attempt is not to provide an exhaustive list of all analysis methods, but rather describe several real-world approaches that are used to gain insight into some aspects of the threat environment (globally or locally in an enterprise).

2.4.3.2 Internal situational awareness

Having a good understanding of the constituency is necessary for effective defense, but what "knowing the constituency" actually means depends greatly on the role of the CERT. An internal CERT should be aware of the assets and infrastructure within its organization and assess their importance with respect to the sensitivity of the data they contain, and their role in supporting business functions. This includes having an accurate inventory of workstations, servers, control systems, as well as a detailed understanding of the allocations of network addresses, network topology and links to ISPs. Typically this is accomplished using inventory management systems that allow even very large enterprises to keep such data up to date.

This knowledge adds context, which is invaluable when a CERT obtains information about a new threat. A CERT with a good degree of internal awareness is able to determine the potential impact of a new threat on the security of data and operations within the organization. For example, consider a vulnerability advisory: if the CERT is aware of which systems are susceptible to a newly discovered exploit, it is able to quickly take appropriate mitigation steps (possibly even disabling some services), monitor patching status and investigate any hosts that could already be compromised. In this case information about software assets is combined with an advisory, yielding actionable information which is used for mitigation.

Information about network infrastructure also includes continuous monitoring of routing. When new routes are unexpectedly advertised it may signal an attempt to bypass security controls, for example through route hijacking. For example, motivated by this concern, US-CERT researched IP blocks of the US government incorrectly advertised in other countries. [28]

Obviously, CERTs with an external constituency have a different definition of an "asset." For a national CERT the analogue of a system inventory would be an inventory of IP allocations within the country that includes up-to-date information about organizations and specific points of contact for incident response (information beyond public WHOIS records)to which incident reports can be sent. A national CERT should also be aware of critical infrastructure and key sectors of industry that might be affected by threats, so incoming reports can be adequately prioritized, and information can appropriate tailored and forwarded.

Apart from inventorying important assets, another element of internal situational awareness is profiling activities occurring within an organization. Profiling can be applied to a variety of behaviors, however in case of internal CERTs it is primarily applied to network traffic [29] and application and system log data in order to characterize the typical behavior of networks and systems. In the process of creating the initial profile and ongoing maintenance a CERT can gain a better understanding of the protected systems and spot suspicious behaviors that require further investigation.

This approach can be extended for the purpose of anomaly detection. By comparing the current state of a system with the previous one – a previous state might be interpreted as "known good," but this assumption is not necessary – it is possible to observe changes. In many cases such changes are completely benign – for example new traffic might be observed simply because a new server was deployed in the network – but sometimes they might be an indication of malicious activity.

Another challenge faced by CERTs with external constituencies is the limited supply of available solutions for monitoring at the scale required, and limitations on the ability to collect data from constituent networks. Typically a national CERT is not in a position to provide comprehensive national-level security monitoring. CERTs with an internal constituency are in a better position in that regard, since there are multiple COTS solution available:

- SIEM or SIEM-like systems allow a CERT to aggregate data from multiple sensors and controls to get some insight into activities in an enterprise, although they often lack advanced analytic capabilities (e.g., it is not possible to build statistical models describing network traffic).
- Systems for the analysis of network traffic and anomaly detection provide tools for building profiles of networks and alerting on changes.
- Vulnerability assessment capabilities provide a way to identify vulnerable systems by checking their patch status directly and through network-based scanning of exposed services.

While these commercial systems might not address all the needs of an organization, they still allow analysts to get at least a partial understanding of the current situation within the constituency.

Because of the nature of their relationship to their constituencies and the scope and scale of their missions, national and governmental CERTs typically rely on data sharing arrangements, manual analyses and custom tools in order to maintain a good picture of situational awareness within their area of responsibility.

2.4.3.3 External situational awareness

Achieving a high level of situational awareness in regard to relevant threats will always be difficult for an organization. A CERT may simply not have access to the data that is required to understand some sources of risk. For example it is difficult to assess whether particular APT and nation-state actors are targeting the constituency when the CERT has not received reports of such attacks. It is difficult to determine whether this lack of reporting means that the constituents were not compromised, were compromised but did not detect the intrusions, or just chose not to report it. Techniques described in this section are focused on data that in general is obtainable for any CERT but availability of relevant information for analysis is a crucial issue that needs to be addressed in the collection step.

If a CERT has sufficient capabilities, it may attempt to get to the root of the problem and track the malicious actors. Such an approach requires not only significant analytical resources but also access to good sources of relevant information. Therefore in practice, CERTs tend to focus on the observed tactics of malicious actors rather than attempting to formulate a comprehensive picture of the actors themselves. They do this by tracking campaigns, both those targeted at particular organizations (some of these might be considered “APTs”) and those affecting the general population–which is often the case with criminal activity. A common example of criminal activity are spam campaigns distributing malware. To understand these campaigns, a CERT will collect samples of the spam emails, analyze the attached malware, and compare the results of this analysis with information about other malware, in order to gain insight into the prevalence and propagation of entire families of malware.

By tracking campaigns and malware activity a CERT may eventually be able to map out parts of the malicious infrastructure supporting the campaigns. With the right data, a CERT team can identify C&C servers, victims, indicators of compromise, and many other types of actionable information. As more information is collected it is possible to correlate new observations to known facts. The maintenance of situational awareness becomes easier with time as the picture becomes more complete.

Unfortunately the availability of automated tools that support these types of analyses is limited. It is much easier to monitor internal networks than observe attacks happening in the wild across the Internet as a whole. External situational awareness on a large scale requires asking difficult questions and there is no COTS solution that would provide an answer to what kind of attacks are most commonly used to infect computers in a country. Yet, this is an information that could be very valuable to a national CERT.

This lack of ready-to-use tools has prompted some national and governmental CERTs to develop their own monitoring systems, tailored for their specific needs. ARAKIS[37] is an example of a distributed monitoring system that allows analysts to study attack techniques and detect worm propagation. It was developed by CERT Polska[38] and deployed in multiple governmental networks in Poland in 2007. Its main data source is a network of low-interaction server honeypots that are used to obtain suspicious traffic coming from vulnerability scanners, worms, reconnaissance activities, etc. Additional data is obtained from IDS probes listening to the honeypot traffic and firewalls deployed in the production networks. ARAKIS uses this data to automatically track multiple features of the network traffic and raise alerts whenever a significant change is detected – e.g. a sudden increase of connections on one of the ports. The system is also able to discover common patterns in traffic that correspond to widespread attacks and to generate Snort signatures for them.

Processing in ARAKIS is an example how fully automated analysis of low-level data can yield higher-level actionable information. The resulting advisories are actionable in context of a national CERT – they often trigger an investigation, which provide insight into current attack techniques. Without automated identification of changes in network traffic, it would be very difficult to have an up-to-date view on the large-scale attacks within constituency and situational awareness would be negatively affected.

Similar distributed systems were also developed by other CERTs:

- EINSTEIN [30] is a capability developed by US-CERT that has been deployed at federal agencies in the US since 2003. The first version of EINSTEIN (EINSTEIN-1) was used to detect anomalies through the analysis of network flow data. Subsequent iterations of the project have introduced signature-based intrusion detection (IDS)and, most recently, intrusion prevention (IPS) functionality using a set of known indicators (including network addresses, DNS names, and email headers) for the identification of malicious traffic. EINSTEIN can also be used to correlate attacks across multiple agencies.
- CarmentiS[39]was developed by a group of German CERTs which also collects low-level data from multiple honeypots that have been paired with IDS probes. Unfortunately there is no publicly available information on the automated analyses performed by the system.
- TSUBAME[40] is another large-scale network monitoring system. The project was initiated by JPCERT, and starting in 2007 it was deployed throughout the Asia Pacific region under the auspices of APCERT. TSUBAME has built-in visualization capabilities that allow analysts to get an overview of widespread attacks, and then investigate those attacks using collected data.

The systems described above are focused on the collection of low-level data, primarily network traffic summaries augmented by IDS alerts, and their main purpose is the observation of trends, detection of common patterns and intrusion detection based on other indicators. An online service provided by Team Cymru – TC Console[41]– is an example of a system that addresses a different aspect of situational awareness. It is a graphical interface for data repository maintained by Team Cymru that contains a large volume of indicator-level data of global scope, primarily in the form of reports on infected machines and addresses of malicious web sites. TC Console gives an analyst an environment where she can see what kind of malware was detected on her network, visualize trends in botnet infection rates (see the next section for more detailed discussion on visualization techniques) and compare her

[37] See http://www.arakis.pl
[38] Authors of this report are members of CERT Polska.
[39] See http://www.carmentis.org/index.html
[40] TSUBAME Working Group: http://www.apcert.org/about/structure/tsubame-wg/index.html
[41] See http://www.team-cymru.org/Services/TCConsole/

constituency with others in regard to the number of identified threats. This service does not produce actionable information (although individual indicators can be accessed and they might be actionable) but rather builds on top of it, providing an overview.

BGP Ranking by CIRCL[42]is another example of a service working on indicator-level data that can be used for comparison purposes. It is a publicly available list of ASNs that have been scored based on their degree of "maliciousness." Although this data has limited value by itself for situational awareness, global reputation scoring data may be a useful input for a situational awareness capability.

Another system developed by a national CERT (NCSC-NL) –Taranis– is focused on processing high-level information, primarily vulnerability and incident reports, and warnings about recent threats. The system facilitates the collection, assessment and distribution of high-level information (often coming in an unstructured form) by an analyst, but it does not have capabilities to automatically analyze data extracted from the reporting.

The case study "Improving situational awareness through botnet monitoring" contains a more detailed example of how a national CERT can use combination of several techniques to track major threats within its constituency.

Overall, the development of techniques that can be used to improve situational awareness is very much an ongoing research topic. In particular, there are many opportunities to apply machine learning methods – which are already used by some of the existing automated systems –to the development of new analysis algorithms. One example of an active research area is approaches for the clustering of malware into families based on similarities in their implementations (see [31][32] [33]). These algorithms can be used to automatically classify a much larger number of malware samples than would be possible with manual methods. A number of other research efforts focus on the mining and correlation of other sorts of large security datasets. The NECOMA[43] project is an example of an ongoing joint project between EU and Japanese institutions that attempts to improve these aspects of analysis.[44]

2.4.3.4 Visualization

Visualization was already mentioned in the context of investigations but it is for situational awareness that advanced visualization methods start to be essential. Using visual techniques for data analysis is not specific to the domain of information security. On the contrary – leveraging human perception to spot patterns and outliers has a long history, going back to 19th century or even earlier. [34]

Visualization allows an analyst to display a huge amount of data in a relatively small amount of space, which can be very helpful for the exploration of the large machine-generated security datasets that CERTs have to process. The modern approach to interactive visualization was formulated by Shneiderman as the so-called visual information-seeking mantra: "Overview first, zoom and filter, then details-on-demand." [35] This quote neatly encapsulates how an analyst will interact with a dataset to glean meaning from the data, and the model applies to a variety of analysis tasks involving large volumes of data.

The application of visualization to the security domain is not a new concept and many existing tools generate charts, graphs, maps and other visualizations to support analysis and exploration. Marty [36] provides a good reference of available ready-to-use solutions, while a recent book by Jacobs and Rudis [37] is a good source for anyone that wants to use more generic tools to implement customized

[42] See http://bgpranking.circl.lu
[43] See http://www.necoma-project.eu
[44] Some of the authors of this report take part in the NECOMA project.

visualizations. In the context of situational awareness, visualization is especially useful when working with time-based data – a human can relatively easily recognize and understand periodic behavior, trends, anomalous events and causal relationships when events are visualized along a timeline. Naturally, visualization of other types of data can be enlightening as well. For example, the distribution of categorized data (e.g., numbers of different incident types) can be shown using histograms, a variety relationships can be visualized as edges between nodes, and plots can be used to identify clusters of related activity. Since an exhaustive list of all the methods that can be applied to represent data visually is out of scope of this document, which is instead referred to the reader to the aforementioned publications .

It is important to recognize that the visualization used in the analysis step of the processing pipeline is just a means to an end and it is just one of multiple methods to facilitate analysis. An analyst may use it to get an understanding of complex datasets relevant to current threats and to formulate conclusions, for example to identify IP addresses that exhibit anomalous behavior that can be examined to determine whether they are malicious and should be published as indicators. An example of visualization that gives an illusion of situational awareness but does not really provide actionable information is representing attackers or victims as points on a geographic map. Often such visualizations highlight population centers instead of answering questions regarding adversary behavior, security trends, or technical features of threats.

However sometimes simpler, table-oriented methods of displaying information can still be effective, for example displaying lists of known types of malware, sources and targets of attacks, all sorted by the number of related incidents in the last week. This way of communicating data may seem very basic but still it gives an idea of what services are currently most threatened and which attackers generate the most alerts and should be blacklisted. An important reason for choosing simpler techniques is also that in many cases preparing a good, reusable visualization is a non-trivial task that requires a substantial effort both for design and implementation.

The capabilities of most tools for security information management (see the accompanying inventory document) in regard to visualization are very limited or non-existent. Some projects employ general-purpose software for this task, for example IFAS uses Kibana to create interactive dashboards. Codenomicon VSRoom[45] is an example of a visualization tool designed specifically for situational awareness, and includes support for multiple output methods, including maps, histograms and contingency tables. It is also capable of processing real-time data feeds from AbuseHelper. Unfortunately the project seemed to be inactive at the time of the writing of this report– the software was last updated in 2012.

Data visualization is a dynamically developing field and there is a constant flow of new ideas and tools. The abundance of existing tools and software libraries means that producing a visual representation quantitative information is now easier than ever, but on the other hand, it might be difficult to choose software. The following are examples of some well-known generic tools that can serve as a good starting point for developing security data visualization tools: D3js[46] and the libraries based on it, Tableau,[47] Splunk and R.[48]

[45] See http://www.codenomicon.com/vsroom/
[46] See http://d3js.org
[47] See http://www.tableausoftware.com
[48] See http://www.r-project.org

2.4.4 Metrics

CERT teams employ a variety of approaches to evaluate their performance and plan future operations. There is a great deal of interest in data-driven approaches based on quantitative information. This goal can be achieved through the use of metrics that can be compared across CERTs, especially those operating at the national or ISP level. Metrics can describe various aspects of a CERT's operations and the security status of its constituency, for example:

- number of machines infected by malware
- number of attacks originating from the constituency
- remediation rate –the proportion of IP addresses that are repeatedly reported in incident reports to the total number of attack sources

Such metrics can be used to determine the effectiveness of a CERT, for example in the context of malware cleanup efforts. Long-term analyses (on the scale of months or years) can also indicate if particular threats are getting worse or better, e.g. if banking trojans are affecting a larger percentage of the population over time indicating that remediation efforts may not be effective. Knowing the trends allows a CERT to effectively allocate resources, both internally within a CERT and within the whole organization.

This type of analysis could be viewed as a part of the situational awareness building activity described in the previous section, since it provides the CERT and policy makers with a better high-level understanding of the threat environment. However, in this report the metrics where chosen separately from situational awareness due to differences in the way the information is processed and used. The metrics considered here are based on a statistics computed primarily from indicators and the results of their application, and from high-level information taken from strategic reports. The metrics are not actionable by the CERT since the output is not used to drive a direct defensive action (see the definition of actionable information in section 1.2), unlike some of the other products of situational awareness which could lead to information that can inform an action (e.g., a new malicious IP might be identified through situational awareness activity that is then added to a blacklist).

Comparative metrics-based studies of information security issues are published occasionally (e.g. by OECD [9]), but the methodology employed is still not mature. Crucially, there is no agreed-upon set of metrics that would allow straightforward cross-country comparison. The issue is further complicated by the limited number of data sources that are representative globally – threats are usually monitored selectively, so a data source might have plenty of information regarding attacks targeting one country and no information on another. The Cyber Green Initiative [38] is a new project started by JPCERT that aims to improve the current situation by providing a set of comparable metrics based on a global-scale aggregation of data .

2.4.5 Meta-analysis and source evaluation

When dealing with multiple data sources, especially external ones (see section2.1.1), it is often difficult to determine the comprehensiveness and quality of collected information. [49] Since actionability depends on quality (see section 1.2), this represents a significant problem for a CERT, which depends on reliable data to make decisions about mitigations. Also when a CERT shares the information with external entities, it should know (and communicate clearly) the degree to which the original source is reliable.

[49] This discussion does not apply to low-level data for the most part, since it contains raw record of facts, without classification

There are approaches that allow a CERT to analyze the coverage of some types of blacklists [39] and compare their effectiveness. Notable parameters that are useful for the evaluation of sources are the rate of addition of information about new threats, and the overlap and representativeness in regard to threats in different parts of the world [40][50] . Meta-analysis of this sort can provide objective, quantitative information about data sources, which in turn can affect the way that they are used. For example, a CERT can compute the overlap across multiple datasets in order to identify sources that provide duplicative data.

But even with these methods, accuracy is particularly difficult to estimate when the CERT acts as a proxy and forwards indicators or incident reports to other organizations. Obtaining the ground-truth regarding infections and several other types of threats is almost impossible without direct access to a compromised machine or C&C server, an opportunity that is rarely available to CERTs that act as coordinating centers. A partial solution to this problem is providing a feedback mechanism that allows the final recipients of the information to report false alarm rates and other data quality issues.

Preferably, the feedback channel should be implemented as another data source and integrated with the rest of the processing infrastructure. However, currently available tools lack such capability, which is a major barrier to the widespread collection of this sort of meta-information about quality and effectiveness. TC Console is an exception here in that it supports the reporting of false alarms directly from the web interface.

Overall, the evaluation of sources should be considered a part of an ongoing quality assurance process. This allows a CERT to identify unreliable sources that produce information that must be verified before taking any actions, and to confirm the quality of known-good sources, without relying on subjective trust in a producer. It is also worth noting that the evaluation of sources does not produce actionable information as such – rather it is used to improve the processing pipeline itself.

2.4.6 Recommendations

- Correlate information from all available datasets; use tools (both interactive and fully automated) that facilitate correlation.
- Make collection of additional data during an investigation (e.g. ,deploying a honeypot) and integrating it with existing knowledge as easy as possible.
- Apply visualization techniques for exploratory data analysis and understanding large datasets, but only where visualizations provide insight that can lead to actions.
- Minimize the time required for analysis, especially during investigations. To do so, try to remove technical obstacles, e.g. inconvenient interfaces, slow storage, and incomplete automation.
- Try to detect and analyze threats proactively, and put routine processes to support this sort of analysis, for example by implementing a "playbook" approach in the organization.
- Understand your constituency and know the critical resources.
- Identify and monitor threats relevant to the constituency.
- Observe the (short- and long-term) dynamics of threats to predict future challenges.
- Where possible use automated systems to derive various types of actionable information from low-level data.
- Analyze and evaluate the quality of current and potential data sources, developing quantitative approaches to measure accuracy. When possible use this information to improve the way in which data is collected.

[50] "Measuring the IQ of your Threat Intelligence Feeds" provides precise definitions of the terms novelty, overlap and population as part of a description of an evaluation approach.

2.5 Distribution

The final step in the pipeline corresponds to the application and dissemination of actionable information that has passed through the previous stages. The importance of distribution stems from the simple fact that in order to ensure that appropriate mitigation actions are taken, a CERT needs to notify its constituents, who can then act appropriately based on the information in notifications. Disseminating the correct information in a timely fashion is frequently a non-trivial task requiring an investment of time in understanding those constituents' needs.

Under some circumstances the CERT can also perform the mitigation itself. In this case the distribution step corresponds to the deployment of indicators to the relevant security systems. Finally, the distribution step includes sharing of information with trusted partners as part of collaborative analysis.

2.5.1 Recipients of information

In general, the main goal of distributing actionable information is the mitigation of threats that could potentially impact the organizations for which a CERT is responsible. Of course there are many ways to achieve this objective, ranging from direct actions against the offensive capabilities of malicious actors (e.g., with botnet takedowns), to proactive hardening of the infrastructure defended (e.g., through timely application of patches), or remediation of damages already done (e.g., by cleaning up infected machines). This section covers applications of information for such operational security purposes.[51]

To communicate effectively, a producer of information must understand how recipients differ in what they consider actionable, so the content and form of a message can be adjusted accordingly. The attempt is to describe the important classes of recipients below.

2.5.1.1 Internal entities

In the simplest case, a CERT can apply information it has received or discovered to a direct action. This is possible in cases when a CERT has some degree of authority over its constituency, so that it can carry out mitigation actions directly or through close cooperation with relevant departments in the organization (e.g., by directing a NOC to update the configuration of a security control).

In such a setting the distribution step is straightforward. If mitigation involves reconfiguration of parts of the organization's network infrastructure (e.g., a router or an IDS), information just needs to be converted into a form that can be accepted by relevant systems (e.g., the configuration file of a BGP daemon or a rule in an IDS configuration). For actions that must be performed by a human, a brief textual instruction should be sufficient, for example, "re-image laptop with the inventory number #AB1234," as long as the actions are in accordance with some agreed upon set of procedures.

Some common internal applications of actionable information include:

- using an IDS to match destination IP addresses of outgoing connections against a list of known C&C servers to detect potentially infected hosts,
- configuring HTTP proxies to block access to websites hosting exploit kits or phishing,
- null-routing (blackholing) all traffic to known malicious IP addresses,
- searching archived DNS logs upon receiving information about domains used for malicious purposes in order to identify compromised hosts,
- scanning machines for artifacts associated with malware using IoCs extracted from reports on targeted attacks (e.g., APT1 [42]).

[51] The focus is on the use of information for network defense, and do not consider other applications like supporting scientific research or reporting to administrative bodies.

Naturally, this list is far from exhaustive. The case study "Using indicators to enhance defense capabilities" provides a more in-depth example of how actionable information can be applied within an enterprise.

Still, the quality of information (see the criteria for actionability outlined in section 1.3) imposes limits on what can be done with it in practice:

- If the quality of information is insufficient – e.g. it is outdated and incomplete – it will not be utilized at all.
- When the overall quality of information is good, the CERT can use it to detect threats, e.g. through IDSes or other monitoring systems.
- Only when the CERT is confident that the information is very accurate, it should deploy automated blocking (e.g., by blackholing single IP addresses). Fully automated prevention mechanisms cut the reaction time, however they bring the risk of disruption of legitimate services.

The distribution of actionable information internally is entirely under the control of the CERT, and as such should not pose a significant challenge other than the engineering effort required to integrate some of the systems used in a particular part of the organization.

2.5.1.2 External entities

The distribution of information to external entities introduces additional complications. Although the direct approach to mitigation of threats might be the most effective, many CERTs (notably national ones) do not have the authority required to take direct actions in all cases, so they will generally be unable to implement corrective measures on their own. In such a setting, a CERT must act as a proxy or a coordination center, providing information to an entity (e.g., an ISP) that can effectively take a mitigation action. The exact course of action taken by the final recipient depends on the situation.

Due to the global nature of threats, it is common for a CERT to have information that is relevant beyond its immediate area of responsibility. Unless there are important reasons to withhold this information (e.g., a non-disclosure agreement or local laws), it may be useful as an early warning and should generally be shared with trusted external organizations. The global nature of the Internet means that collaboration for analysis and mitigation of threats can be significantly more effective than working in isolation.

Finally, yet another example of sharing information with external entities is sending feedback to data providers, which might allow them to improve their service.

At the time of writing this report, the field of cross-organizational information sharing is developing at a rapid pace. Multiple standards and best practice guides relevant to this topic were published in recent years, including NIST's Guide to Cyber Threat Information Sharing. [41] On the technical side, many tools and services – both commercial and open-source –have been developed to facilitate the process of information exchange. Some of these systems are described in the accompanying inventory document, but readers must take into account that the landscape of information sharing is changing and any inventory will become outdated after a short period of time, so some amount of independent research is necessary before committing oneself to a particular approach.

It is important to realize the capabilities of potential recipients and whether they will fully understand particular information and be able to act upon it. External organizations that are potential recipients can fall into three broad categories:

- **Low capability**. These are usually small organizations, with no CERT and limited security automation in place. Typically these organizations do not have mechanisms for ingesting advanced forms of information feeds like real-time data streams.
- **Medium capability**. These are typically medium to large enterprises, have a CERT or an analogous security operations center, and use automated tools (e.g., a SIEM) to handle security data.
- **High capability**. These consist of security data clearinghouses, coordinating CERTs, and information security vendors who frequently possess infrastructure to process large amounts of data, and handle near-real-time feeds from multiple information sources.

Knowing the capabilities of cooperating organizations will allow a CERT to decide what should be shared and choose the appropriate formats and protocols for communication.

2.5.2 Technical aspects of information distribution

This section covers various technical aspects of information distribution to external organizations. The most important issues related to the way the data is transported were already described for the collection step (see section 2.1.3) and still apply here. The major difference is that during collection, a CERT usually has little influence over the way in which data is made available by its sources. In contrast, in the distribution step CERTs have complete control over all aspects of the publication of information feeds, constrained only by the limitations of the solutions available on the market.

2.5.2.1 Common issues

In order to communicate effectively, data formats and transport mechanisms should be adjusted to the expected recipient. In practice there are many tradeoffs that have to be considered, so it is often difficult to find an optimal solution. Nevertheless, there are certain aspects of information sharing that should be taken into consideration in all cases.

It is generally the responsibility of the producer of information to select information that is, broadly speaking, relevant to the organizations that make up its constituency. For example, for a national CERT details about infections in another country can be considered noise from the point of view of its customers. Part of the value added by an information broker like a CERT is the filtering and contextualization it can do on behalf of recipients, reducing the effort required by a recipient who might otherwise decide that it is not worth the effort and discard everything.

A common use case for national CERTs is the distribution of indicators for threat to those organizations that could be affected by it. The identification of affected organizations is often done by leveraging information about known organizations that was added in the preparation step (see enrichment – section 2.2.4) to **filter** the data prior to distribution. Tools developed by national CERTs usually have such functionality (e.g., consider Megatron, AbuseHelper, and n6).

Another consideration is the timeliness of information. Generally, a CERT should try to minimize delays that are introduced during processing. When it is acting as an intermediary, this will often mean simply passing along information as-is. When information is produced by the CERT itself, it should not delay disseminating results of its analysis, since the threat may change in the meantime. For time-critical information, the implementation of the transport mechanism may be important, requiring near-real-time sharing or frequent batch updates. In general, near-real-time methods based on streaming or publish-subscribe frameworks are preferred but in practice batch modes of delivery remain much more common in real-world information sharing systems.

CERTs should always try to ensure that distributed information is as complete as possible. This means that output formats that allow the representation of internal data structures (see section 2.2.1) with no loss of fidelity during conversion are preferred (if the recipient accepts them). In particular, the

inclusion of accuracy estimates might be very useful when a recipient has to determine what actions to take. If the formats used do not support the transmission of such metadata, it may be sent through a separate channel from the data itself.

If the CERT collects feedback on the data it distributes, it should provide a straightforward way for constituents (or other recipients) to refer to particular elements of the reports (e.g., via unique identifiers associated with particular indicators).In general, providing automatic feedback mechanisms for machine-to-machine communication is challenging, and most feedback is generated and distributed manually, usually after the final recipient realizes that there is some problem with the data (e.g., after seeing excessive false alarm rates, or recognizing that old or out-of-date data records have been received).

2.5.2.2 Recipients with low capability

Small organizations often lack the proper infrastructure to ingest and utilize all available data feeds. If the recipient/consumer does not use automated tools to process information, information like incident reports and early warnings will be processed manually. This must be taken into account when considering the optimal method of distribution.

Under such circumstances, the usefulness of standard communication channels and data formats diminishes –in fact, they introduce complexity that may become a barrier to the ingestion of the actual content. Instead, a simple, universally used method like an email with a textual description of a threat may be the best way to communicate information, allowing the recipient to understand and act on it. This is the approach is taken by Megatron, which uses template-based emails as the primary method of distribution. Most systems for automated management of information can be configured or customized to provide similar functionality.

Additional contextual information is attached to the data itself. That context might include a general description of the nature of the problem and links to reference materials that may prove essential for the proper interpretation of the received information, especially for smaller organizations that lack good situational awareness. Visualizations like diagrams and charts illustrating trends can also facilitate understanding of complex issues.

2.5.2.3 Recipients with medium capability

When sharing with other CERTs or large organizations, the expectations are that at least some types of data will be processed in an automated fashion. Such recipients deal with incident reports, vulnerability announcements, alerts from monitoring systems, and the like on a daily basis and may place importance on using standards for information exchange.

It was previously stated that when a CERT is a recipient of data, it does not have much choice when it comes to the way it is transmitted – data providers do not necessarily have to take a CERT's preferences into account. However, it should be kept in mind that if the receiving CERT was able to implement an automated processing pipeline for some types of incoming information, it should be able to handle accepting a new feed without much effort. When it receives a report in a format that is not supported by its current infrastructure, it may handle it in one of the following ways:

- handle using an entirely manual process,
- manually convert it to the format accepted by currently used tools and then process it,
- adapt the current infrastructure to support the new data source.

But if the effort associated with manual processing or conversion outweighs perceived benefits, data will be discarded, which is definitely undesirable from the point of view of a producer. A recipient organization can only be expected to make changes to its processing infrastructure to accommodate

sources expected to provide ongoing feeds of valuable data. When making the decision to integrate a new source, an organization will consider many factors including the value and uniqueness of the new source relative to existing ones, and the extensibility of the existing information processing solution.

The adoption of standards could help solve this problem by allowing the producer and recipient to more easily integrate their automated systems, thus lowering the cost to ingest information. Unfortunately reaching agreement on a standard when confronted with many competing transport mechanisms and data formats can be difficult. In practice the adoption rate of standards is low.

There is no single reason for this situation, however several contributing factors can be pointed. First, the domain of information security is evolving at a rapid pace so specifications may become obsolete after a few years as they fail to address new issues. Second, it is difficult to create standards that will be used by a diverse group of entities while being specific in details, [14] which is why simple standards like CVE are universally used, but complex specifications rarely see widespread implementations. The companion report "Standards and tools for exchange and processing of actionable information" can be used as a reference of important standards for data formats and transport mechanisms.

Fortunately, the situation seems to be improving, as new MITRE standards for information exchange – STIX as the way to express information and TAXII for transport – get more attention from vendors and other information producers than previous initiatives like IODEF. If this trend continues, it is possible that in the near future standards will play a more significant role, which will benefit the whole sharing community.

Until that happens, an approach that is worth recommending is supporting multiple methods of distribution and allowing the recipient to choose one that is most suitable. The downside of this solution is an increased implementation effort, however some existing tools already offer multiple output formats (e.g., CIF, MANTIS).

2.5.2.4 Recipients with high capability

In many ways coordinating CERTs and data clearinghouses are similar to the recipients described in the previous section. One of the main differences stems from the fact that organizations dealing routinely with large amount of data from multiple sources have expertise and resources to integrate new recurring data feeds relatively easily. They are also able leverage their position to work on a larger scale, for example coordinating remediation efforts globally.

Another issue is related to performance – if a CERT wants to share data on a medium-to-big scale (in the order of magnitude of millions of records or gigabytes of data per day), the efficiency of data formats and transport channels begins to play a significant role. In such situations, it may be advisable to turn to simpler, non-standardized formats (e.g., based on CSV) to avoid the overhead associated with processing of "heavy," XML-based formats. It is also possible to use the semantics and the data model from one of the more complete but complex formats and encode this information using a more efficient representation. There is ongoing work to provide a standard way for doing it for STIX, and in principle this approach can be applied to any data format.

2.5.3 Sharing policy

CERTs routinely handle sensitive information, therefore in the distribution step appropriate measures must be put in place to ensure that the scope of data given to external organizations is strictly controlled. A CERT should have a well-defined sharing policy to determine what types of information can be provided to different organizations, so there is no ambiguity when a new recipient appears or a new type of data is scheduled for distribution.

According to a survey of CERTs made by ENISA in 2013 [43], technical barriers to sharing are more common but are easier to overcome in comparison to legal ones. A good guide on issues related to data disclosure in the context of the EU law was written by Cormack [44]. However, since our focus is on the technical aspects of distribution, an in-depth discussion of law, vetting of recipients and similar matters is out of scope of this report.

When sharing sensitive information, a CERT can add metadata that designates how it should be handled by the recipient, especially if it can be distributed further. The most common method is the TLP, which defines four simple sharing levels. There are also other approaches, for example elements in IODEF have a "restriction" attribute that can be used for the same purpose, and APWG has its own system of data markings.[52] Some standard data formats were designed to hold this kind of metadata (aforementioned IODEF, STIX), but in all other cases it must be communicated out-of-band.

Even if a CERT decides to share with an external organization, information is often transformed in order to remove selected sensitive elements. This transformation is commonly referred to as **anonymization**. The following elements are commonly anonymized:

- data sources – for various reasons the original providers of information often do not wish to be identified to the final recipients;
- victims of an attack – when reporting an attack to the constituency of origin, details of the victim organizations and systems are not strictly required and may be considered sensitive;
- information related to collection methods– addresses of honeypots or other sensitive information that can expose covert collection methods;
- personally identifiable information of any kind.

When distribution of information is implemented by automated tools (and this is the only way that sharing can be done on a large scale), it is essential that CERT's sharing policies are implemented by these systems. The capabilities of systems used by national CERTs' vary greatly – many of them do not even have proper permission systems, some can be configured to implement arbitrary sharing policies but do not provide good configuration tools, and the ones that have advanced features are the exception rather than the rule. Moreover a large majority of existing systems lack anonymization features. As a work-around, this transformation can be done in the preparation step, so the stored data is already anonymized. A more detailed analysis of several solution is available in. [45] One also must keep in mind that anonymization has theoretical limitations and, there are methods that can be used to reveal the original data by correlating the anonymized information (e.g., IP addresses) with other datasets. Therefore some very sensitive information elements should be removed completely from the information shared with external parties.

2.5.4 Recommendations

- In the case where a CERT has authority over its constituency:
  - try to utilize all available sources information to mitigate threats – preferably through reconfiguration of monitoring systems or elements of the production infrastructure;
  - estimate the accuracy of information and decide if it is actionable in the particular environment;
  - automatically block only if you are confident in the high accuracy of information, otherwise just issue alerts that will be verified by analysts.
- When sharing data with external organizations:
  - Identify the capabilities of the recipient and use the most appropriate format and transport mechanism;

[52] Anti-Phishing Working Group Blog, seehttp://apwg.org/data-logistics-blog/data_log

  - support standardized formats, but do not force them on recipients not ready to process them automatically;
  - evaluate which standards are worth implementing given the data that you have and your sharing environment – incrementally adding support for more standards (based enumerations like CVE first, then indicators like OpenIOC, finally broad reporting formats similar to STIX and IODEF) might be a good approach;
  - for many use cases STIX is currently the most promising format for sharing a broad variety of security information, however be advised that STIX-based implementations are still immature;
  - for large-scale data exchange give preference to lightweight formats to minimize processing and size overhead –in particular consider JSON and REST APIs (due to their popularity and widespread support in software);
  - provide recipients with choice of output formats, if feasible;
  - prepare a sharing policy that defines what data is shared with whom;
  - communicate whether the recipient can distribute information further and under what restrictions;
  - anonymize or redact sensitive information;
  - provide a feedback mechanism;
  - use automated systems that allow the implementation of a sharing policy and facilitate the management of the sharing process.
- Minimize any delays in distribution – prefer real time communication channels if accepted by recipients.
- Distribution can be done in parallel to analysis – results of analysis can be sent incrementally (as updates).

3 Case studies

The following three case studies cover various aspects of actionable information handling by CERTs. While these scenarios are not always based on real-life stories, they capture the operational processes of real CERT teams and the actual features of the tools they use. Hence, they can and should be used as food for thought about how the tools and techniques described in this report can be applied to improve CERT team's ability to produce, share and use actionable information.

3.1 Using indicators to enhance defense capabilities

This case study describes the steps taken by a defense contractor in order to analyze a targeted attack. The aim of this study is to illustrate the actionable information that can be extracted from an infected network. This information is then used to both establish the scope of the existing infection and prevent its spread. In this study it is also considered how an analyst can sometimes generalize the knowledge gained, and even use it to prevent similar attacks on the infrastructure in the future. Finally, it will be presented how each piece of the actionable information and each attack event can be related to phases in the Kill Chain model. [7]

3.1.1 Collection

STIRC company is a leading contractor in the defense industry, working on highly classified military projects. In the first week of July 2014 an employee of STIRC received an e-mail outlining changes in the agenda for a conference he was supposed to attend in the near future. Unfortunately, the new schedule conflicted with other commitments, leading him to call the organizers in order to withdraw his participation. The organizers informed the employee that they did not send any message regarding changes in the agenda. This worried him. He became suspicious and reported the case to the STIRC security team.

3.1.2 Preparation

A CRITs instance was used to prepare all of the data. This included automatically parsing the data files and extracting metadata.

3.1.3 Storage

A security analyst began the investigation by importing all of the available information into a CRITs instance running at STIRC. This internal instance stored all of the artifacts, including communication dumps in the PCAP format, dropped samples, the initial e-mail, sandbox results, IP addresses and domains associated with this campaign. This standardized, central storage allowed the analyst to share these artifacts with the other security analysts working in the company.

3.1.4 Analysis

The security analyst used CRITs and its services (scripts extending base functionality of the system) to analyze the gathered data. This started with the analyst uploading the initial e-mail, along with the results of a sandbox analysis of the malicious file attached to the email, which included PCAP and executable files generated by the malware. Then, after a semi-automated process, the analyst discovered and documented relationships between the different observables.

The sandbox analysis of the PDF was performed by one of the security team members. It was done by using a specially crafted sandbox environment (based on QEMU[53]), because the analyst was unable to

[53] See http://wiki.qemu.org/Main_Page

open the PDF in a standard VirtualBox[54] installation. Analysis of the sample revealed that it was a malicious PDF file exploiting a previously unknown vulnerability (which later received the identifier CVE-2014-0560). AV analysis using the VirusTotal[55] service showed that this threat was not detected by any of the antivirus solutions.

Analysis using the SysInternals Suite[56] in the specially crafted sandbox environment revealed a malicious file stored in the user AppData folder and downloaded from a domain in the .su TLD, which subsequently led the analyst to identify a number of IP addresses located in an autonomous system (AS) operated by China Telecom. The analyst also noted that the malware updated the Windows autorun Registry key to ensure persistence. Further analysis showed that another file was then downloaded. In this case, the downloaded file was sophisticated malicious code known to be used in targeted attacks against the defense industry, hidden away in the ADS (Alternate Data Stream) of the file.

Analysis using both the CRITs tool and information obtained from other sources, like the VirusTotal and Anubis[57] services, revealed crucial details about the whole campaign, like the fact that the C&C server IP address was the same as the IP address of the email sender. Other details included the attribution of the campaign to a particular group of the attackers known to the defense industry. This attribution was only possible because the internal CRITs instance had been used in the analysis of previous campaigns. Analysts were able to see that attackers had used similar strings across multiple malware samples, and that the weaponization method was similar to a previous campaign.

All of the uncovered evidence and mitigation steps with the Kill Chain model can be aligned. This helps analysts to visualize the attack timeline and to prevent similar compromises in the future. Relating activities to phases of the Kill Chain helps a security team reason more clearly about the proper defensive measures to employ.

3.1.4.1 Reconnaissance

Attackers used publicly available data like the employee's e-mail address, and information published about the conference where employees were scheduled to present their work. The agenda, the conference name and the logo were used in a successful spear phishing campaign. This sort of open source reconnaissance is virtually impossible to mitigate against. Hence, no mitigation took place.

3.1.4.2 Weaponization

A standard document file type, namely a PDF, was used as a weaponized deliverable. The attacker crafted a PDF that exploited the vulnerability CVE-2014-0560 to gain access to the underlying operating system. The PDF's name and a convincing story in the e-mail were used to lure the user into opening the file. Since the user was expecting a similar e-mail, the attack vector proved to be successful.

3.1.4.3 Delivery

The weaponized PDF file was delivered as an attachment to an email message. Since a 0-day exploit was used, none of the company's AV solutions were able to detect it. The message was crafted in a way that was not consistent with spam and the recipients were carefully chosen during the reconnaissance phase. To mitigate against this threat in the future, firewall rules were added to the

[54] See https://www.virtualbox.org
[55] See https://www.virustotal.com
[56] See http://technet.microsoft.com/en-us/sysinternals/bb545021.aspx
[57] See http://anubis.iseclab.org

mail server in order to drop all incoming traffic from the malicious IP addresses and to log all connection attempts so they could be retroactively analyzed. User mailboxes were scanned for attachments having the same SHA-256 or a similar ssdeep[58] hash as the original one. Five other email messages were identified but only two other users said that they had opened the email message.

3.1.4.4 Exploitation

The exploit targeted an unknown and unpatched vulnerability, which would later be assigned the identifier CVE-2014-0560. The user was misled into opening the PDF, the Adobe Reader software was exploited, and the attacker eventually gained access to the victim's machine. The PDF document was sent to Adobe for analysis, and a patched version of Adobe Reader was later pushed out to all of the user workstations.

3.1.4.5 Installation

The malicious document contained shellcode which downloaded an additional piece of malware. This malware was a specially-crafted downloader, which was responsible for the installation of the malicious software that would ultimately be used to exfiltrate files from the user's machine. This software was configured to persist using a registry key, and, using covert techniques, contacted the C&C server. From that point onward the attackers had a full access to the target environment.

3.1.4.6 Command and control

The C&C server's traffic was routed through an autonomous system (AS) operated by China Telecom. All of the C&C traffic was over HTTPS. This was done in order to hide the malicious traffic within normal web browsing activities. Firewall rules were put in place at the Internet gateway to block all incoming and outgoing traffic to the identified malicious IPs. NIDS (Snort[59]) logs were also analyzed for any alarms raised by connections with suspected IP address and subnets. Special rules were written in order to inform administrators of the connection attempts. Name server logs were then retroactively analyzed for any connections with the C&C server, but no new infections were found.

3.1.4.7 Actions on objectives

Intruders, after gaining the control of the environment, proceeded to exfiltrate data from the target machines. The data and applications targeted on end-user machines was specific to the defense industry. One of the exfiltrated files contained passwords for different company services. The attacker used this login data to try to obtain the access to these services. This included attempts to connect to the corporate VPN.

These login attempts were identified by analyzing network flow data using the Argus [60] tool. Fortunately, VPN access was secured using two-factor authentication. The other internal services were not exposed to the Internet and all of the connection attempts to them were blocked. Hence, some of the actions were mitigated by security controls that were already in place.

During the course of the following weeks, all traffic from the identified attacker IPs was redirected to a specially prepared honeypot server. This was done using firewall (iptables) ACL rules and various open-source and proprietary software for simulating workstation behavior, including the Dionaea[61] and Kippo[62] honeypots.

[58] See http://ssdeep.sourceforge.net
[59] See https://www.snort.org
[60] See http://qosient.com/argus/argusnetflow.shtml
[61] See http://dionaea.carnivore.it
[62] See https://github.com/desaster/kippo

3.1.5 Distribution

The analysis revealed a variety of actionable information, which was then used to take additional actions to mitigate against similar attacks. Unfortunately, CRITs does not support the automatic deployment of indicators in external security controls, like firewalls or IDSes, so all of the actions were performed manually, and then tracked by logging actions to CRITs. This logging provides a useful resource for reasoning about future changes to security controls at the network perimeter.

In order to detect any activities similar to the ones performed by the analyzed malware hosts HIDS (Mandiant Redline[63]) rules were created to detect the Windows artifacts obtained during the analysis. A scan was then performed on all of the Windows-based workstations on the internal network to identify any additional machines that had been compromised.

The configuration of the internal name server (BIND) was changed to sinkhole the C&C connections. This led to the identification of two more infected users.

This analysis led to the development of a list of indicators of compromise (see Figure 6) corresponding to the activity observed during the attack. For the sake of completeness, a description of the extracted IoCs in OpenIOC format are included. This file was created with the help of Mandiant IOCe.[64]

Figure 6. Indicators of compromise in the OpenIOC format.

```
<?xml version="1.0" encoding="UTF-8"?>
<iocxmlns="http://schemas.mandiant.com/2010/ioc"xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"id="3b92a703-2f56-4b9d-862f-35c1fa2847d6"
last-modified="2014-10-15T09:16:04">
  <short_description>*New Unsaved Indicator*</short_description>
  <authored_date>2014-10-15T06:30:05</authored_date>
  <links/>
  <definition>
    <Indicatoroperator="OR" id="2655453b-799d-480d-a170-99d821f4ad1d">
      <IndicatorItemid="6a6cdce1-372e-4d25-9aed-76164c6a1983"condition="is">
        <Contextdocument="FileItem"search="FileItem/StreamList/Stream/Name"    type="mir"/>
        <Contenttype="string">encrypted</Content>
      </IndicatorItem>
      <IndicatorItemid="77269c72-f649-4e7f-9dcb-4b1adadfe2cb"condition="contains">
        <Contextdocument="Network"search="Network/DNS"type="mir"/>
        <Contenttype="string">home.windows-security.su</Content>
      </IndicatorItem>
      <IndicatorItemid="70005dc6-e884-4b15-b986-6060fe8586ec"condition="is">
        <Contextdocument="DnsEntryItem"search="DnsEntryItem/Host"type="mir"/>
        <Contenttype="string">home.windows-security.su</Content>
      </IndicatorItem>
      <IndicatorItemid="3e3716f7-a8d3-4eb1-b990-dce2b441d11d"condition="is">
        <Contextdocument="DnsEntryItem"search="DnsEntryItem/RecordData/IPv4Address"type="mir"/>
        <Contenttype="IP">184.128.98.111</Content>
      </IndicatorItem>
      <IndicatorItemid="f5156778-bc61-4b1c-ab60-02e2a7514141"condition="is">
        <Contextdocument="DnsEntryItem"search="DnsEntryItem/RecordData/IPv4Address"type="mir"/>
        <Contenttype="IP">184.128.152.18</Content>
      </IndicatorItem>
      <IndicatorItemid="7bc87e9f-4da2-41c6-a143-660cf060b6a0"condition="is">
        <Contextdocument="DnsEntryItem"search="DnsEntryItem/RecordData/IPv4Address"type="mir"/>
        <Contenttype="IP">184.128.2.34</Content>
      </IndicatorItem>
      <Indicatoroperator="AND" id="9dda7744-001d-49bf-a2ad-579daa4f5b2c">
        <IndicatorItemid="391b98de-0383-440e-a0cf-9cfe6663764d"condition="is">
          <Contextdocument="RegistryItem"search="RegistryItem/KeyPath"type="mir"/>
          <Contenttype="string">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run</Content>
        </IndicatorItem>
        <IndicatorItemid="31afd6cd-7c98-44b8-b188-d2e92d374e65"condition="is">
          <Contextdocument="RegistryItem"search="RegistryItem/ValueName"type="mir"/>
          <Contenttype="string">driver32</Content>
        </IndicatorItem>
      </Indicator>
    </Indicator>
```

[63] See https://www.mandiant.com/resources/download/redline
[64] See http://www.mandiant.com/resources/download/ioc-editor/

```
  </definition>
</ioc>
```

3.1.6 Summary

The actionable information gathered during the course of investigation helped address the immediate need to contain the infection as well as the mitigation of future similar threats originating from the same source. The actions taken were based on DoD information operations [46] doctrine. The table below describes the actions taken by the security team in terms of the "courses of action matrix" proposed by Lockheed Martin. [7] The other possible actions were either not applicable or would not have prevented future intrusions.

Table 2. Applied mitigation measures (based on courses of *action matrix* in [7]).

Phase	Detect	Deny	Distrupt	Degrade	Deceive	Destroy
Reconnaissance						
Weaponization	NIDS					
Delivery		Firewall ACL				
Exploitation		Patch				
Installation	HIDS					
C&C	NetFlow, NIDS	Firewall ACL			DNS redirect	
Action on Objectives	Audit log				Honeypot	

3.2 Improving situational awareness through botnet monitoring

This case study is focused on improving situational awareness on Internet threats based on information gathered from various data-sharing sources. The primary actor in this case study is a national CERT that has begun to focus its operations on malware analysis and on monitoring botnet activity.

Networks of compromised computers represent one of the most well-known threats of the current Internet landscape. They are used as a tool of crime, cyber warfare [47] and espionage. The monitoring of botnet activities is a challenging task that requires the proper infrastructure and a team of dedicated experts. From the national CERT perspective, botnet monitoring is an important tool for properly assessing infection rates in their constituency networks, identifying infection hotspots, and correlating locally observed outbreaks with global reports. A botnet tracking capability enables a CERT to better help its constituency to handle incidents in their networks, and provides the visibility needed to coordinate efforts to take down a botnet or limit its spread.

3.2.1 Collection

In order to monitor botnet activity the CERT started by gaining access to feeds with information about compromised hosts and C&C servers. The CERT leveraged several of the sources described in the ENISA report "Proactive Detection of Network Security Incidents," [13] most notably:

- The Shadowserver Foundation[65] publishes lists of the IP addresses of zombie computers, and domain names and IP addresses for C&C servers,
- Abuse.ch,[66] like Shadowserver, operates several botnet tracker services for tracking the IP addresses and domain names associated with the ZeuS, SpyEye and Palevo botnets.

The CERT has also started receiving data about infections in its constituency from other national CERTs, a cooperative effort based on bilateral agreements. Information sharing was facilitated by various systems, including AbuseHelper and the CERT Polska n6 platform.

The next step the CERT took was to put processes in place for obtaining malware samples. This included ensuring that mechanisms were in place to receive samples from trusted parties and from incidents reported by constituency members. The CERT also stood up a spamtrap. Additionally, by accessing its network of client honeypots and looking for compromised malicious web sites, the CERT was able to obtain additional samples. This task was performed using Thug[67] and The Honeyspider Network[68] configured to crawl web sites and save the malware it found for later analysis.

3.2.2 Preparation

The vast volume of data obtained and the limited resources at the CERT's disposal created a requirement for pre-processing in order to limit the number of samples to be analyzed. The pre-processing of data from the spamtrap[69] was implemented as a simple filtering scheme that selected samples based on the national TLD of the CERT and the existence of particular keywords in the CERT's native language. This allowed the CERT to significantly decrease the number of samples that were of less interest to the CERT or its constituents.

3.2.3 Storage

In order to be able to efficiently collect, search and compare malware samples, the CERT developed a custom malware repository. A custom flat file database was used which, aside from just storing the samples, allows submitted malware samples to be tagged and then queried using those tags, and hash values based on SHA-256 and ssdeep.

3.2.4 Analysis

Once the CERT had a solution in place for receiving and storing malware samples, the team began to regularly review samples. Later, several samples extracted from spam emails attracted special attention since they seemed to be associated with a new campaign. This led to further analysis.

Malicious samples obtained from spam or harvested from web sites typically act as installers (or droppers) which drop a payload that actually implements the final stage of a compromise on a victim system. The CERT created a specialized laboratory environment to obtain and analyze this final stage malware. There are two basic approaches to setting up the infrastructure to support an analysis sandbox environment. One can start with a virtualized infrastructure or build sandboxes directly on

[65] See https://www.shadowserver.org
[66] See http://www.abuse.ch
[67] See https://github.com/buffer/thug
[68] See http://www.honeyspider.net
[69] Spamtrap is a type of honeypot dedicated for collection of spam messages.

real, "bare-metal," hardware. KVM[70] and VirtualBox[71] are two popular free virtualization solutions. Tools for managing a virtual machine environment include virt-manager[72] and the libvirt library.[73]

Running malware directly on real hardware may be necessary when analyzing malware that employs analysis evasion techniques and refuses to run in a virtual machine. The bare-metal environment is much harder to manage and setup, but may be the only way to reproduce the environment of a user's computer. In this case, the CERT chose to set up bare-metal machines using the Diskless Remote Boot in Linux[74] (DRBL) tool and software developed in house, which allowed the team to create a laboratory based on a version of the Windows operating system. The team managed the laboratory using a Remote Desktop Protocol[75] client and a network-enabled KVM switch.[76] The environment gave the CERT a way to perform both automatic and manual analysis of malware samples.

Not every dropper delivers botnet malware. The CERT analyzed selected samples using runtime analysis tools that can quickly recognize typical botnet behavior. This type of analysis is typically implemented by running the malware through a sandbox tool that automatically performs the check. Three common sandboxes are Cuckoo Sandbox,[77] which can be installed and used in the laboratory, and Anubis[78] and Malwr,[79] which are online services that do not require any special software. Online solutions are much simpler to use, but entail the disclosure of information about the sample to other users of the service, which is not always a desired outcome. Sandboxes allow an analyst to characterize the behavior of a malware executable, including the system calls it makes, processes it spawns, IP addresses it contacts and files it downloads. In addition, if a sample is discovered to be malicious, an expert can extract unique signatures, for example, YARA[80] rules to use in threat monitoring and detection software.

In this particular case, the CERT started with samples of malware attached to spam. The team then discovered that some additional files were downloaded and used to execute new processes. These new processes again tried to contact some servers located on the Internet. This sequence of behavior is consistent with a typical botnet malware infection. This provided strong evidence that further analysis was justified.

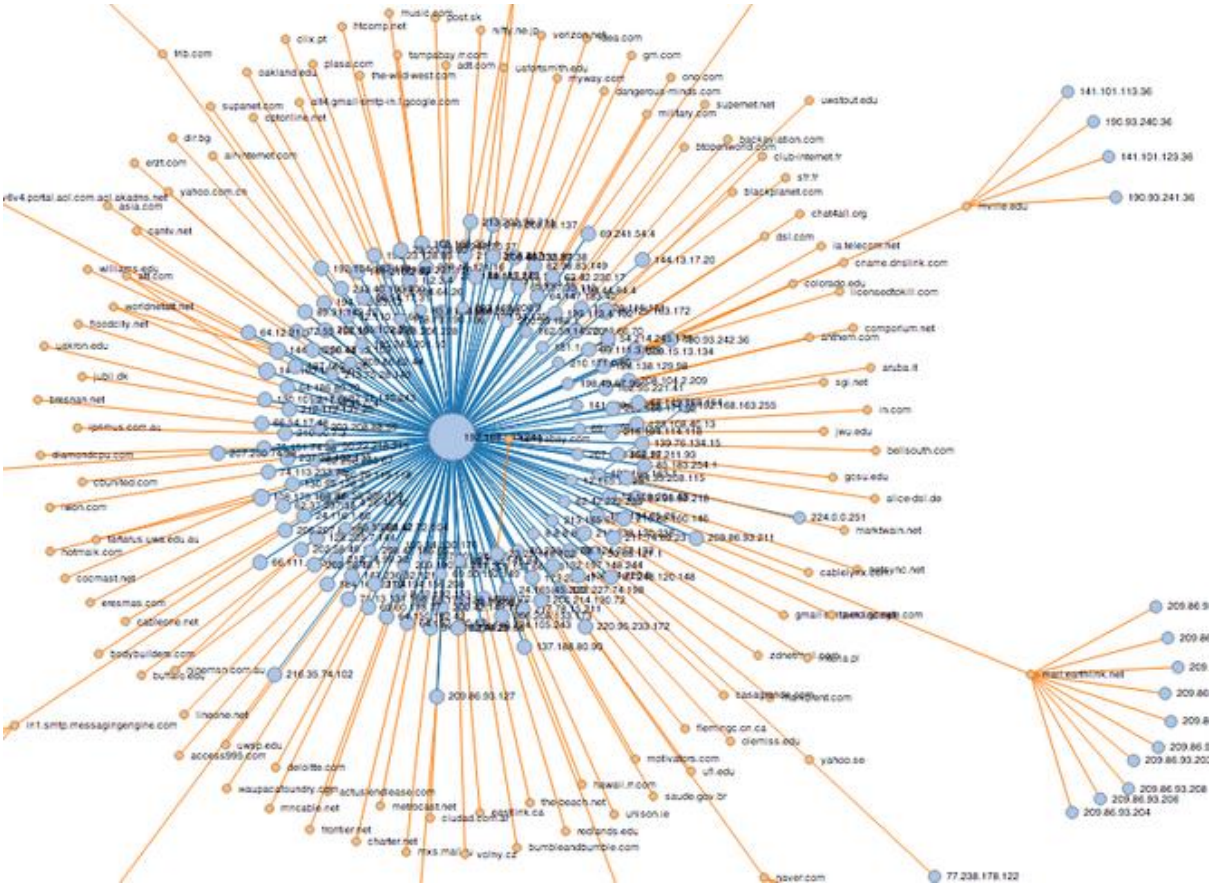
Long term analysis is concerned with comprehensive monitoring of a botnet behavior over time, tracking its configuration changes, like new C&C or drop zone servers, and monitoring its communication protocols and actions. Such analysis requires the implementation of a network monitoring solution to observe traffic generated by malware samples. To fulfill this task the CERT chose MalCom,[81] a tool designed to help analysts map out botnet infrastructure. The tool enables an analyst to display graph visualizations based on the network traffic generated by malware samples. MalCom was used to extract additional information, like domain names and IP addresses of contacted C&C servers, together with relations between them.

[70] See http://www.linux-kvm.org/page/Main_Page
[71] See https://www.virtualbox.org
[72] See http://virt-manager.org
[73] See http://libvirt.org
[74] See http://drbl.sourceforge.net
[75] See http://en.wikipedia.org/wiki/Remote_Desktop_Protocol
[76] See http://en.wikipedia.org/wiki/KVM_switch
[77] See http://www.cuckoosandbox.org
[78] See http://anubis.iseclab.org
[79] See https://malwr.com
[80] See http://plusvic.github.io/yara/
[81] See https://github.com/tomchop/malcom

Long term analysis also allowed the CERT team to observe updates in botnet configuration and identify new versions of bots, which were installed in the laboratory to enable additional monitoring. In this case, bot configuration updates were delivered using encrypted files shared via peer-to-peer communication between bots. Analyzing these sorts of files requires expert knowledge and advanced technical skills, but by performing such analysis an expert will gain invaluable insight into botnet actions. There is no single approach to decrypting such files, as algorithms used to encrypt the information contained within them differ from one botnet type to another and sometimes change between the bot versions. Regardless of the particular encryption scheme, the key problem faced by the analyst is obtaining an encryption key, which is generally only available at runtime and only for a brief moment stored in a computer's RAM. Memory analysis tools like Volatility Framework[82] prove to be extremely helpful with this task. In this case the extracted encryption key was an important piece of forensic information as it was used by in-house developed software for decrypting the botnet's configuration files. Information within these files provided insight into active campaigns using this botnet infrastructure that included a campaign that targeted online bank accounts.

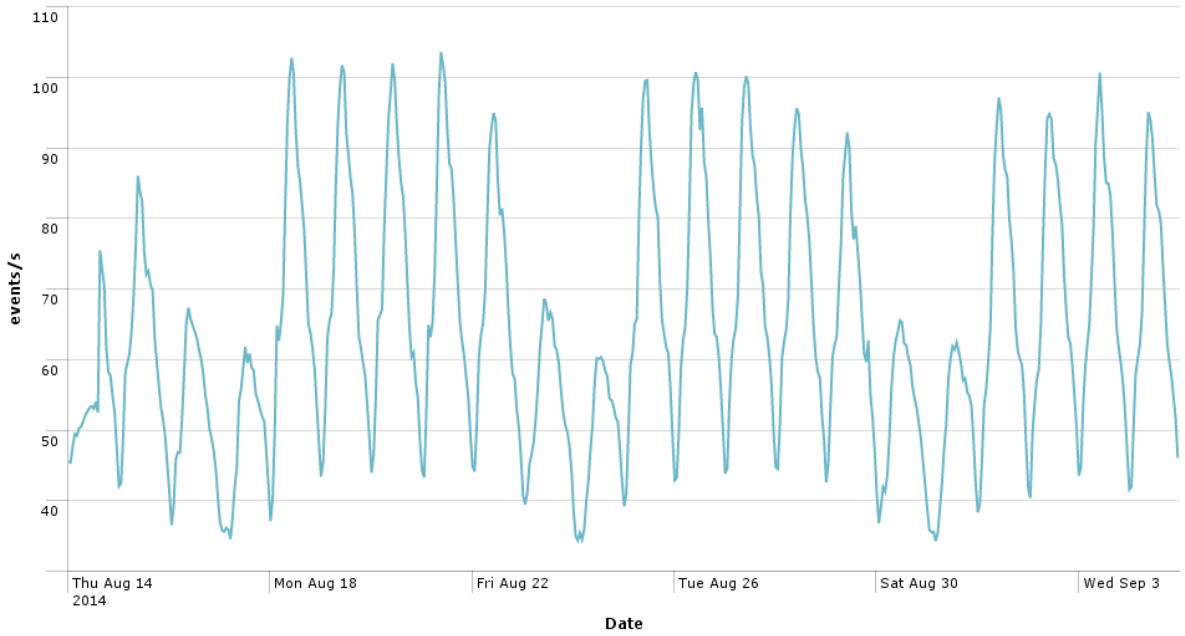
[82] See https://github.com/volatilityfoundation/volatility

Figure 7. Example visualization from MalCom, from Github page of the project.



The monitoring of this botnet infrastructure, aided by the MalCom tool, led to the discovery of novel techniques used to maintain constant control over the entire botnet network. Researchers were able to observe that samples were making a lot of connections with other infected bots, as presented in the Figure. The findings were consistent with studies of the ZeuS Gameover botnet [49][48] and showed that its author created a network of zombie computers almost entirely independent from the typical backend infrastructure, and relied heavily on a peer-to-peer(P2P)network to transport stolen user data, and deliver new configurations and updates to the bot software. The P2P nature of the botnet inspired the implementation of a “drone” bot whose sole purpose was to monitor these peer-to-peer communications directly by participating in the network. The drone gathered additional information about botnet structure that would have been impossible to obtain using just passive observation of network traffic.

Figure 8. Rate of bot connection attempts with the sinkhole.



The raw data gathered in the course of an investigation like this usually cannot be easily understood directly by analysts without some level of tool support. The analysis and visualization features of tools like Kibana[83] and Splunk[84] help analysts discover new insights. Both tools allow analysts to query datasets to identify and focus on key details and to interpret patterns of activity. The visualization of security data helps analysts make sense of the vast amount of processed information and makes it easier for them to spot changes that would be difficult to detect using automated analysis.

3.2.5 Distribution

The CERT's analysis efforts created a rich body of knowledge about the botnet, including large datasets of malware samples and their YARA signatures, lists of IP addresses for infected hosts and C&C servers, and Snort[85] signatures based on analysis of bot communication protocols. This data was shared with relevant parties in the CERT constituency that included ISPs and institutions affected directly by the malware, in order to allow them to detect and notify users about infected hosts.

The CERT also implemented preventive actions against botnet activities. In order to limit the threat posed by botnets, sinkholes were set up to catch malicious traffic directed to known botnet C&C servers. Such action required cooperation and information exchange with DNS providers and other national CERTs in order to take over C&C domain names and redirect traffic from infected computers to sinkholes. This setup prevented users from contacting the real C&C servers and at the same time allowed the discovery of more IP addresses of zombie machines, including hosts located outside of CERT's constituency.

Starting from the time that the CERT started actively monitoring botnets in its constituency networks, it gained invaluable knowledge about botnet actions and infrastructure, which otherwise would be unavailable or hard to obtain. Data collected from running sinkholes and from botnet analysis

[83] See http://www.elasticsearch.org/overview/kibana/
[84] See http://www.splunk.com
[85] See https://www.snort.org

performed in the malware laboratory proved to be especially valuable. This work has enabled the CERT to maintain an up-to-date view on the botnet threat landscape. Ongoing monitoring of constituency networks with tools like spamtraps and honeypots, and leveraging visualizations of gathered datasets, allowed the team to track the evolution of botnets and to identify new threats posed by new botnet infrastructure and new campaigns.

3.3 Effective data exchange on a national level

As part of their information sharing mission, national-level CERTs must make sense of rapidly growing volumes of network security incident data. The faster and more completely a CERT can make actionable data available to network owners, administrators, ISPs, and hosting providers the greater the chances of reducing the impact of attacks.

In February 2012, CERT Polska launched the first generation of n6, a platform designed for the reliable and fast delivery of large volumes of network incident data to affected parties. The initial version of n6 was based on a filtering engine that processed lists of indicators one by one, attempting to map each indicator to the appropriate recipients (based on CIDRs, ASNs and other features). n6 took files with these indicator lists as input and rewrote them to a flat file repository using a representation that preserved that mapping.

Development of the second version of the n6 platform began in mid-2013. The new version offers faster data processing, a more sophisticated unified data model that can be efficiently implemented in both relational and document databases, and a RESTful API.

Access to n6 data feeds is provided as a free service for all registered organizations, but the software itself is closed source. However, a large part of the platform, including the frontend (API server), will be released under an open source license by the end of 2014. This case study is based on CERT Polska's use of the2nd generation of the n6 platform for data exchange (also described in "Standards and tools for exchange and processing of actionable information").

CERT Polska chose to develop n6 from the ground up, but similar results could be obtained by building on top of existing open source tools. This was the approach initially taken by the IntelMQ project. The first versions of the IntelMQ system leveraged AbuseHelper, Logstash and Elasticsearch, although the latest release replaces AbuseHelper and Logstash with a custom implementation. Although building a system from existing components will significantly reduce the development effort, it is important to appreciate that integration (and development to add missing functionality) still entails a substantial amount of effort. For a CERT that does not require advanced functionality, a distribution system based on a simpler tool like Megatron can be set up quickly and with minimal effort.

3.3.1 Collection

The n6 platform integrates with many sources of indicator-level information, and includes extensive support for consuming feeds from automated monitoring and analysis systems including sandboxes, honeypots, sinkholes, WAFs, and IDSes. Most of the data feeds in n6 originates from the operational activities of CERT Polska, including monitoring systems it manages and data received from partner organizations (e.g., other CERTs). These feeds are combined with data obtained from publicly available sources. The data collection process is automated for most feeds but the level of automation depends on the data origin. Most of data is either received by email, or downloaded as static files over HTTP from the web sites of cooperating organizations. Data coming from one-time (non-recurring) sources are inserted into the system manually. Collected data comes in different formats, including JSON, CSV, IODEF, text files, which are in some cases compressed and encrypted.

3.3.2 Preparation

Collected data are unpacked, parsed and normalized immediately after receiving. In the next stage normalized data is enriched with IP addresses, AS numbers and country codes using a caching DNS resolver and a GeoIP system. Enrichment is implemented early in the processing pipeline to keep up with the short validity timeline for some indicators. This is especially important when monitoring activity for websites associated with phishing which may only be active for several hours.

The n6 data model is based on a normalized form where objects are represented as discrete *events*. An event includes a set of predefined required and optional attributes. During normalization data is normally converted to a representation based on JSON. The most important event attributes are listed below:

- malicious IP address related to the threat,
- destination IP address (e.g., of a sinkhole or honeypot),
- ASN and country code,
- category of the event (bots, C&C, phish, etc.),
- source and destination port used in TCP or UDP communication,
- fully-qualified domain name (FQDN) and URL related to the threat,
- hash of the binary related to the threat (MD5, SHA-1),
- method used to obtain the data (data obtained from sinkhole, results from behavioral analysis, interaction with honeypots, reports from IDS/IPS/WAF, etc.),
- one of three levels of confidence that the information is accurate.

As soon as data is enriched, the n6 engine links the events to affected organizations using a built-in contact database by using multiple matching criteria, including associated IP ranges, autonomous systems, country codes and domain names.

The confidence level assigned to events will vary based on the origin and the technical methods used to derive the data. For example data obtained from a CERT Polska sinkhole would be deemed more reliable than unverified data from an external party about a phishing campaign. Data stored in the n6 platform is marked with one of three levels of confidence. A high level of confidence is assigned to data from trusted and fully verified channels; this includes several of the monitoring systems operated by CERT Polska. A medium level is assigned to generally reliable channels. Finally, a low confidence level is used for unverified data sources. The guidance CERT Polska gives to its constituents is to verify medium level data and generally exercise a cautious approach to data with a low level of confidence.

3.3.3 Storage

When data is received by n6 it is archived for reference in its original form in a document-oriented NoSQL database – TokuMX[86]– which is replicated on 2 physical servers. The database and other key components of the n6 platform are replicated in order to provide redundancy and high availability.

Once normalized, data is stored in a structured form in a relational(SQL) database. The n6 platform uses the MariaDB database with the TokuDB[87] storage engine. TokuDB was chosen for its scalability and data compression. As part of the data preparation, some events are aggregated, including connections to sinkholes. To adequately characterize the activity of an infected host, it is sufficient to

[86] TokuMX – a variant of MongoDB with built-in compression and improved scalability. See http://www.tokutek.com/tokumx-for-mongodb/

[87] TokuDB – an open source storage engine for MySQL and MariaDB, which features built-in compression and high insertion performance for large indexed tables. See http://www.tokutek.com/tokudb-for-mysql/

store timestamps for the first and the last connection, and to group by the type of detected malware and botnet instance (if known).

3.3.4 Analysis

Although there are no built-in automated analyses in the n6 platform, because data is stored in a normalized form in a SQL database it is relatively easy to develop custom queries and analyzes, especially ad-hoc statistical queries. For example, the annual reports published by CERT Polska contain statistical data obtained from the n6 database. The n6 database is also coupled with other internal systems to support other typical operational work.

3.3.5 Distribution

Data stored in the n6 database is made available to subscribed organizations via a REST API over TLS with mandatory authentication using X.509 certificates. Organizations receive only the data that is relevant for their infrastructure. For example, national CERTs can access information that is related to IP addresses in their countries or constituency. Some types of collected data that can be used to detect and block malicious activity (e.g., addresses of C&C servers), are distributed to all organizations in Poland without filtering. A comprehensive permission model allows the definition of arbitrary sharing policies for data based on type, source, IP addresses, and domains. The system also performs on-the-fly anonymization of selected elements, including the IP addresses associated with victims and honeypots.

An organization can query the REST API to request an arbitrary subset of the available data, using a combination of event attributes as the selection criteria. The API can output events in three formats: JSON, CSV and IODEF. The CSV form of the data contains only the most important attributes, making it lightweight, human-readable and easy to process. The IODEF output is intended to support the tools based on that standard, such as CIF and MANTIS. Finally, JSON is the native format of n6 platform and events in the JSON-based output format contain all of the attributes stored in the n6 database.

Different categories of events will be more or less valuable to an organization based on its type and size. For example, in most cases a list of bots will be useful to a corporate security team to detect infected machines in its company's' infrastructure and to take a course of action in order to mitigate the effects of the compromises. The lists of malicious and phishing websites are useful to companies providing hosting, which are able to block these websites. The lists of misconfigured services such as DNS, NTP, SNMP are actionable information for Internet Service Providers, which may be able to take actions based on n6 data to reduce the number of misconfigured services used in reflection and amplification attacks. Finally, data distributed by n6 is broadly useful as a source for indicators that can be used to create rules for firewalls, IDS/IPS systems and proxies. Currently there are more than 250 organizations subscribed to the platform.

4 Gaps and recommendations

This section is intended as an overview of the gaps commonly found in CERT processes for handling actionable information, and provides a set of general recommendations for organizations with information-dissemination responsibilities. More specific recommendations related to actionable information can be found in the detailed discussion of the information processing pipeline in section 2. Many of the issues pointed out in the ENISA report on proactive detection of security incidents [13] remain relevant today. Where appropriate, key findings are referenced from this report, and readers are encouraged to read the previous study again.

A persistent gap can be observed in approaches for to the assessment of the quality of information received from external sources. The gap was originally described in the 2011 and 2012 ENISA studies. [13][45] It is still the case that often the original source and collection method of information are not known. This frequently puts a national CERT in a position where it is unable to effectively evaluate an external dataset, and their only option is to either forward it to the affected parties, hoping that it is actionable or discard it. The development of validation approaches remains an active area of research, and the quality feedback mechanisms in information sharing communities remain primitive.

Creating actionable information based on external information received by a national CERT remains a challenge. The information available is still frequently incomplete, requiring a lot of additional analysis skills and effort to do effective enrichment. Few national CERTs are able to consistently generate new actionable information, whether through external sources, or by themselves or acting as forwarders of information. Technical solutions for processing and forwarding indicators are reaching maturity, but analytical tools that can be used to analyze data and provide additional intelligence lag behind.

Based on this study, it is recommend CERTs abide by the following three general principles when building an information-sharing capability:

1. Establish a doctrine to set expectations among the CERT community. Define clear sharing rules and labels on the data exchanged, as well as expectations for handling and any specific actions that should be taken by the recipient.
2. Try not to start from scratch. Consider what has already been developed and can be leveraged immediately.
3. Explore the possibility of applying additional processes that can provide more context and make the information more actionable.

The goal of any CERT is to obtain some level of true situational awareness for its constituents. That process requires a good knowledge of the assets to be protected and an intimate understanding of the threats faced. Attaining a high level of situational awareness remains a challenge, especially for CERTs with a large scope of responsibility facing large volumes of not-yet-actionable data. Furthermore, as it was pointed out in the study:

> *"Long term trend analysis of data remains a problem for CERTs for various reasons: there is often a lack of resources (both in terms of manpower and financially) and tools necessary to carry out such research. These tools usually have to be developed from scratch by a CERT. Furthermore, to make the research results more useful, it would be worthwhile comparing trend reports published by various CERTs. This is problematic, however, as there is no common understanding of the term 'incident,' not to mention incident types. Other issues, such as different collection methods and capabilities, may mean that statistics are skewed in some manner."[13]*

These words are still valid today, although some attempts to remediate this situation through efforts like. [38] where observed. The suggested remedy in the previous studies still holds promise:

> *"As an incentive to CERTs, ENISA could publish collective trend reports based on data that national, government and other CERTs collect about their constituencies, naming participating teams. These trend reports could influence the future objectives of ENISA, giving first-hand objective information about state of security and saving resources on costly research."[13]*

CERTs face similar challenges when attempting to extract actionable information from the large amounts of data available. Methods that can be used for this purpose are still being developed and rarely see widespread deployment. Visualization can be a very useful tool for analysts when tackling large datasets, however as it was previously noted:

> *"There is no single good way to visualize security data. The knowledge needed to build a proper solution is interdisciplinary and often comes not only from a security expert but is extended with input from social studies and research about human perception. (...) Creating a meaningful visualization of information usually takes considerable resources. More research into providing easy tools to visualize actionable information, especially with focus on providing situational awareness, would be beneficial."[13]*

Many standard formats for the exchange and processing of actionable information have been proposed. Few have been widely adopted, apart from some of the simplest "standards." During this study it was noted that STIX has begun to be adopted and there appears to be some growing enthusiasm for STIX, at least greater than any other comparable standard the industry has seen thus far. As much as we applaud this effort and hope for its success it is still worth pointing out that in practice this footprint has not been visible in the national CERT community, and the tools that support STIX remain immature. The standard itself also introduces significant overhead in regards to most types of indicators processed by CERTs: IPs, DNS names, and URLs. A drive by the STIX/TAXII creators and ENISA to develop tools (both for sharing and analysis) and foster quicker adoption in the CERT community would be welcome.

As a set of general recommendations to CERTs and the following are suggested:

- If possible, standard data formats and transports mechanisms should be used. The accompanying inventory document contains a reference to standards that are currently in use within the incident handling community.
- For some recipients, standard formats may be less helpful for distributing actionable information since they lack the capability to process them. Simpler methods should be used in these cases (e.g., human-readable text). Alternatively, a CERT may consider providing automatically-generated, human-readable reports along with the original data in a structured standard format.
- Adjust the way the information is processed and distributed based on the requirements and constraints for each data type. Be sensitive to the overhead of data formats for large volumes of data, and use more elaborate formats for less frequent reports.

It was also observed that basic indicator data (e.g., IPs, Domains, URLs) as it is exchanged today is frequently insufficient to effectively address threats. This is because these indicators inherently have a very short life expectancy, as attackers can change them frequently. What is needed is increased exchange of higher-level information that describes the methods used by attackers (TTPs) generalized in such a way that they can be applied to another network's environment, detecting intrusions even as the basic indicators change. These TTPs would include queries that analysts can use for exploratory analysis using SIEMs or similar systems. The MISP and CRITs systems are initial steps in that direction but they still lack the advanced analysis, decision-support and collaboration features that would be needed to develop and maintain more sophisticated models of adversary and behavior. It was concluded that ENISA should support work in this area through funding of further research and development.

Finally, our overall conclusion is that information exchanges have not yet reached maturity and the sharing environment will need to develop further before the benefits of these exchanges is fully achieved. As the environment evolves, and mechanisms for the information exchange becomes established, consumers of security data will face new challenges related to data management, analysis and integration of new systems with existing security controls.

5 Conclusion

To our knowledge, this best practice guide is the first study of its kind. This guide aims to provide a picture of the challenges faced by national-level CERTs and other incident response organizations as they build infrastructure for the processing and exchange of actionable information. Our goal was to provide a broad overview of the current information-sharing landscape in the context of actionable information, and to identify existing tools and standards, best practices, gaps and provide recommendations for improvement.

Information exchanges have not yet reached maturity, and the sharing environment will need to develop further before the benefits of these exchanges is fully realized. As the environment evolves, and mechanisms for the information exchange become better established, consumers of security data will face new challenges related to data quality evaluation, data management, and the automation of analysis and mitigation actions.

Our hope is that the contents of this report will provide a collection of useful resources for CERT teams that wish to improve their own information processing and sharing capabilities, and contribute to the larger goal of improving the state of information sharing more generally.

6 References

[1] Georgia Killcrece, Klaus-Peter Kossakowski, Robin Ruefle, & Mark Zajicek. Organizational Models for Computer Security Incident Response Teams (CSIRTs) (CMU/SEI-2003-HB-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.

[2] Sean Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX™)," February 2014, Version 1.1. Available from: http://stix.mitre.org/about/documents/STIX_Whitepaper_v1.1.pdf

[3] David Bianco, "The Pyramid of Pain," http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html (Retrieved on July 22nd 2014).

[4] Carlos Blanco, et al., "A Systematic Review and Comparison of Security Ontologies," 2012 Seventh International Conference on Availability, Reliability and Security, pp. 813-820, 2008 Third International Conference on Availability, Reliability and Security, 2008

[5] Curt Monash, "Examples of machine-generated data," April 2010, http://www.dbms2.com/2010/04/08/machine-generated-data-example/ (Retrieved on 20th July 2014)

[6] David Bianco, On the Misuse of Indicators, July 2013. http://detect-respond.blogspot.com/2013/07/on-misuse-of-indicators.html

[7] Eric M. Hutchins, Michael J. Cloppert, Rohan M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," 2011. Available from: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf

[8] Verizon, 2014 Data Breach Investigations Report, April 2014. Available from: http://www.verizonenterprise.com/DBIR/2014/reports/rp_Verizon-DBIR-2014_en_xg.pdf

[9] OECD, "Improving the Evidence Base for Information Security and Privacy Policies: Understanding the Opportunities and Challenges related to Measuring Information Security, Privacy and the Protection of Children Online," OECD Digital Economy Papers, No. 214, OECD Publishing, 2012. Available from: http://dx.doi.org/10.1787/5k4dq3rkb19n-en

[10] Gavin Reid, "Security Logging in an Enterprise, Part 2 of 2," 2013, http://blogs.cisco.com/security/security-logging-in-an-enterprise-part-2-of-2/ (Retrieved November 2014)

[11] ENISA, "Proactive Detection of Security Incidents – Honeypots," November 2012, Available from: http://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots

[12] Brian Krebs, Espionage Hackers Target 'Watering Hole' Sites, September 2012, http://krebsonsecurity.com/tag/watering-hole-attack/

[13] ENISA, "Proactive Detection of Network Security Incidents," December 2011, Available from: http://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-report

[14] David Mann, JoAnn Brooks, Joe DeRosa, "The Relationship between Human and Machine-Oriented Standards and the Impact to Enterprise Systems Engineering," MITRE Technical Report, 2011, Available from: https://www.mitre.org/publications/technical-papers/the-

relationship-between-human-and-machineoriented-standards-and-the-impact-to-enterprise-systems-engineering

[15] John D. Howard, Thomas A. Longstaff, "A Common Language for Computer Security Incidents," Sandia National Laboratories, 1998. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.4289

[16] "Data Harmonization Ontology," https://bitbucket.org/clarifiednetworks/abusehelper/wiki/Data%20Harmonization%20Ontology, (Retrieved November 2014)

[17] Django DINGOS Developers' Overview, 2013. http://django-dingos.readthedocs.org/en/latest/_downloads/dingos_data_model.pdf (Retrieved November 2014)

[18] Weimer, Florian. "Passive DNS replication." FIRST Conference on Computer Security Incidents, 2005. Available from: http://www.enyo.de/fw/software/dnslogger/first2005-paper.pdf

[19] CERT Polska, Annual Report 2013, May 2014, http://www.cert.pl/PDF/Report_CP_2013.pdf

[20] Curt Monash, "Dataset management," March 18, 2013, http://www.dbms2.com/2013/03/18/dataset-management-revelytix-cloudera-navigato/ (Retrieved on 21st September 2014).

[21] Chang, Fay, et al. "Bigtable: A distributed storage system for structured data," Google, 2006. http://research.google.com/archive/bigtable-osdi06.pdf

[22] Carson Zimmerman, "Ten Strategies of a World-Class Cybersecurity Operations Center," MITRE, 2014, Available from: http://www.mitre.org/publications/all/ten-strategies-of-a-world-class-cybersecurity-operations-center

[23] Moira West Brown, Don Stikvoort, Klaus-Peter Kossakowski, Georgia Killcrece, Robin Ruefle, & Mark Zajicek. Handbook for Computer Security Incident Response Teams (CSIRTs) (CMU/SEI-2003-HB-002), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. Available from: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=6305

[24] Matthew Rosenquist, Cyber security Hunter teams are the next advancement in network defense. Intel IT Peer Network, November 2012. Retrieved: 2014-10-31. Available from: https://communities.intel.com/community/itpeernetwork/blog/2012/11/28/cyber-security-hunter-teams-are-the-next-advancement-in-network-defense

[25] George M. Jones, John Stogoski, ALTernatives to Signatures (ALTS), CERT Coordination Center, Software Engineering Institute, April 2014. Available from: http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=296146

[26] Brandon Enright, Using a "Playbook" Model to Organize Your Information Security Monitoring Strategy, 2013, http://blogs.cisco.com/security/using-a-playbook-model-to-organize-your-information-security-monitoring-strategy/ (Retrieved November 2014)

[27] Committee on National Security Systems, "CNSS Instruction No. 4009: National Information Assurance (IA) Glossary", 2010. Available from: http://www.ncix.gov/publications/policy/docs/CNSSI_4009.pdf

[28] US-CERT Security Trends Report: 2012 in Retrospect, Department of Homeland Security, November 2013. Available from: https://www.us-cert.gov/sites/default/files/US-CERT_2012_Trends-In_Retrospect.pdf

[29] Austin Whisnant, Sid Faber, Network Profiling Using Flow, Software Engineering Institute, Carnegie Mellon University, August 2012. Available from: http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=28115

[30] U.S. Department of Homeland Security, "Privacy Impact Assessment for EINSTEIN 2," 2008, Available from: http://www.dhs.gov/xlibrary/assets/privacy/privacy_pia_einstein2.pdf

[31] Bayer, U., Comparetti, P. M., Hlauschek, C., Kruegel, C., & Kirda, E. (2009, February). Scalable, Behavior-Based Malware Clustering. In NDSS (Vol. 9, pp. 8-11).

[32] Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys (CSUR), 44(2), 6.

[33] Rafique, M. Z., & Caballero, J., "Firma: Malware clustering and network signature generation with mixed network behaviors". In Research in Attacks, Intrusions, and Defenses (pp. 144-163). Springer Berlin Heidelberg, 2013. Available from: http://software.imdea.org/~juanca/papers/firma_raid13.pdf

[34] Tufte, Edward R. "The visual display of quantitative information.," ISBN: 978-0961392147, Graphics Press USA, 2nd edition, 2001.

[35] Ben Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations." Visual Languages, 1996. Proceedings., IEEE Symposium on.

[36] Raffael Marty, "Applied Security Visualization", ISBN: 978-0-321-51010-5, Addison Wesley Professional, 2008.

[37] Jay Jacobs, Bob Rudis, "Data-Driven Security: Analysis, Visualization and Dashboards", ISBN: 978-1-118-79372-5, Wiley, April 2014.

[38] The Cyber Green Initiative: Improving Health Through Measurement and Mitigation, JPCERT/CC, Released October 2014. Available from: http://cybergreen.net/CyberGreenInitiativeConceptPaper.pdf

[39] Kührer, Marc, Christian Rossow, and Thorsten Holz. "Paint it Black: Evaluating the Effectiveness of Malware Blacklists." Research in Attacks, Intrusions and Defenses. Springer International Publishing, 2014. 1-21. Available from: http://syssec.rub.de/media/emma/veroeffentlichungen/2014/07/23/raid14-blacklists.pdf

[40] Alexandre Pinto, Kyle Maxwell, "Measuring the IQ of your Threat Intelligence Feeds", DefCon 22, 2014. Available from: http://www.slideshare.net/AlexandrePinto10/defcon-22-measuring-the

[41] Chris Johnson, Lee Badger, David Waltermire, Guide to Cyber Threat Information Sharing (Draft), NIST Special Publication 800-150 (Draft), October 2014. http://csrc.nist.gov/publications/drafts/800-150/sp800_150_draft.pdf

[42] "APT1: Exposing One of China's Cyber Espionage Units," Mandiant, 2013. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf

[43] ENISA, "Detect, SHARE, Protect - Solutions for Improving Threat Data Exchange among CERTs", November 2013. Available from: https://www.enisa.europa.eu/activities/cert/support/data-sharing

[44] Andrew Cormack, "Incident Response and Data Protection," version 2, TERENA. Available from: http://www.terena.org/activities/tf-csirt/publications/data-protection-v2.pdf

[45] Piotr Kijewski, Paweł Pawliński, "Proactive Detection and Automated Exchange of Network Security Incidents," STO Information Systems and Technology Panel (IST) Symposium, Koblenz, 2012, Available from: http://www.cso.nato.int/Pubs/rdp.asp?RDP=STO-MP-IST-111

[46] U.S. Department of Defense, "Joint Publication 3-13 Information Operations," November 2012. Available from: http://www.dtic.mil/doctrine/new_pubs/jp3_13.pdf

[47] Jose Nazario, "Estonian DDoS Attacks – A summary to date", 2007-05-17, http://www.arbornetworks.com/asert/2007/05/estonian-ddos-attacks-a-summary-to-date/ (Retreived November 2014)

[48] cert.pl Technial Report "Zeus-P2P monitoring and analysis," June 2013, Available from: http://www.cert.pl/PDF/2013-06-p2p-rap_en.pdf

[49] Dennis Andriesse, Christian Rossow, Brett Stone-Gross, Daniel Plohmannm Herbert Bos. "Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus." Available from: http://www.syssec-project.eu/m/page-media/3/zeus_malware13.pdf

Annex A: Abbreviations

The table below presents the list of abbreviations used in the document.

Abbreviation	Explication
ADS	Alternate Data Stream
API	Application Programming Interface
APWG	Anti-Phishing Working Group
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CAPEC	Common Attack Pattern Enumeration
CERT	Computer Emergency Response Team
CIDR	Classless Inter-Domain Routing
CIF	Collective Intelligence Framework (software)
CNSS	Committee on National Security Systems
COTS	Commodity Off-the-Shelf
CRITs	Collaborative Research Into Threats (software)
CSIRT	Computer Security Incident Response Team
CSV	Character-Separated Values or Comma-Separated Values
CVE	Common Vulnerabilities and Exposures
DBIR	Data Breach Investigations Report
DNS	Domain Name System
DRBL	Diskless Remote Boot in Linux (software)
ELK	Elastic Search, Logstash, Kibana (software stack)
FIRST	Forum of Incident Response and Security Teams
HDFS	Hadoop Distributed File System
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IFAS	Information Feed Analysis System (software)
IOC	Indicator of Compromise
IODEF	Incident Object Description Exchange Format
IPS	Intrusion Prevention System
ISP	Internet Service Provider

JSON	JavaScript Object Notation
KVM	Keyboard, Video and Mouse
KVM	Kernel-based Virtual Machine (software)
MISP	Malware Information Sharing Platform (software)
NAT	Network Address Translation
NECOMA	Nippon-European Cyberdefense-Oriented Multilayer threat Analysis
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology (USA)
NOC	Network Operations Center
NTP	Network Time Protocol
OECD	Organisation for Economic Co-operation and Development
OSS	Open Source Software
P2P	Peer-to-Peer
PCAP	Packet CAPture (file format)
PDF	Portable Document Format
PII	Personally Identifiable Information
RAM	Random Access Memory
RAT	Remote Administration Tool
RDBMS	Relational Database Management System
REST	Representational State Transfer
RTIR	Request Tracker for Incident Response (software)
SHA	Secure Hash Algorithm
SIEM	Security Information and Event Management
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
STIX	Structured Threat Information Expression
TAXII	Trusted Automated eXchange of Indicator Information
TCC	Team Cymru Console
TCP	Transmission Control Protocol
TERENA	Trans-European Research and Education Networking Association
TLD	Top-level Domain
TLP	Traffic Light Protocol
TLS	Transport Layer Security

TTP	Tactics,Techniques and Procedures
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
VSRoom	Virtual Situation Room (software)
WAF	Web Application Firewall
XML	Extensible Markup Language
YARA	Yet Another Regex Analyzer (software)

ENISA
European Union Agency for Network and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office
1 Vass. Sofias & Meg. Alexandrou
Marousi 151 24, Athens, Greece

ISBN: 978-92-9204-107-6
doi: 10.2824/38111
Catalogue number: TP-05-14-107-EN-N

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu