Global Information Assurance Certification Paper

The Conductor Role in Security Automation and Orchestration

GIAC (GCIH) Gold Certification

Author: Murat Cakir, murat.cakir@clf.com.tr
Advisor: Adam Kliarsky
Accepted: August 21, 2017

Abstract

Security Operations Centers (SOCs) are trying to handle hundreds of thousands of events per day and automating any part of their daily routines is considered helpful. Ultimately fast creation of malware variants produces different Indicators of Compromise (IOCs) and automated tasks should adapt themselves accordingly. This paper describes the possible use of automation at Threat Hunting, Identification, Triage, Containment, Eradication and Recovery tasks and phases of Incident Handling along with practical examples. Also describes how they can fail or can be systematically forced to fail when orchestration is missing. Orchestration should not only cover dynamic selection of proper paths for handling of specific tasks, but should also provide circumstantial evidence while doing that. Finally, there should be a Conductor who should know "when and how to use the baton" to accept, modify or reject any part of the automated flow.

1. Introduction

Security Information and Event Management (SIEM) market leaders like LogRhythm, Nitro Security, ArcSight, QRadar, and Splunk create good visibility for Incident Handlers who need to take fast and accurate actions. To get benefit from machine learning algorithms to detect malicious activities, some Security Operation Centers (SOCs) use software from User Behaviour Analytics (UBA) vendors like Exabeam, Forcepoint, Fortscale, Niara, Securonix, and Sqrrl (Oltsik, 2016). Better approach is to use targeted prevention tools and develop detection mechanisms based in threat intelligence (Crowley, 2017).

To differentiate good or bad, decision algorithms might be enough. But identify the origin of the attacks and track the trails the responders need to deal with paths or links. Dealing with links can be complicated for conventional database but not for graph based database (e.g. Neo4j) where everything is stored as displayed. Tools powered with database can easily detect and display activity paths in terms of nodes, properties and relationships. Sqrrl using such a strategy can inspire future products which will use similar components like big data (e.g. Apache Hadoop), key/value stores (e.g. Apache Accumulo), graph database (e.g. Neo4j or similar). As such, initiatives like The Integrated Adaptive Cyber Defense (IACD) project (by the Department of Homeland Security (DHS) and the National Security Agency (NSA)) seek to adapt a traditional control and decision approach from the physical world to apply it in cyberspace.

To drive cyber operations timelines from months to minutes to milliseconds, there is ongoing work of implementing The Observe-Orient-Decide-Act (OODA) Loop at speed and scale. (Johns Hopkins University Applied Physics Laboratory, 2017) One product that was born and raised along with IACD project is Phantom Cyber's Phantom.

Phantom, being active in IACD project from its early days is a promising architecture with a very intuitive interface for creating playbooks and can run its playbooks on a wide range of security products. It works at the Decide and Act stages of Observe, Orient, Decide, and Act (OODA) decision cycle. While creating and running

playbooks can be interpreted as ‘automation’, the way they are created and being executed is closer to the concept of ‘orchestration’.

According to Schneier, it is possible to automate only what you're certain about, and to deal with uncertainty in cybersecurity ‘focus’ needs to be on making the people effective and not on replacing them; i.e. focus needs to be on security orchestration rather than automation (Schneier, 2017). This paper discusses the state of art at Security Orchestration and Automation through the statement above.

2. Incident Response

CSIRT defines six identifiable stages of response to an INFOSEC incident as; Preparation, Identification, Containment, Eradication, Recovery and Follow-up” ("COMPUTER INCIDENT RESPONSE GUIDEBOOK", 1996) Almost identical and widely accepted stages defined by SANS are; Preparation, Identification, Containment, Eradication, Recovery and Lessons Learned (Kral, 2011).

Famous conductor and professor, Gustav Meier mentions that conductors spend most of their preparation time for a performance without access to their ‘instrument,’: the orchestra (Meier, 2009). With a good set of indicators ‘Identification of an incident’ may take remarkably short amount of time. Investing time here will be beneficial for speeding up daily work. Considering the hash of a malicious file exists at an accessible database already, checking for that hash value takes less than a second whereas analyzing binary through your own Viper + Cuckoo setup takes approximately two minutes. Thus, setting up the infrastructure, tools, playbooks, work flows, recruiting people with appropriate skill sets and training them for the target they will be monitoring are all tasks taking place during preparation time. This is the time spent without a single real incident.

If there is a difference between the duration required by the task and the duration that is allocated to the task by the handler then loss of quality is inevitable or the work will be ineffective. With proper automation and correct assignments, we can achieve both quality and effective working.

Incident Handlers dealing with thousands of incidents per day will appreciate any proactive effort that can help them at any of Incident Handling stages. At Preparation

Stage Incident Handlers setup tools, techniques and procedures for each stage of Incident Handling.

At a SANS Survey "Incident Response: How to Fight Back" conducted with 259 respondents (88% working in an IR role) Alissa Torres highlights two of the findings as;

1. There is a need for tools that increase visibility into threats and how they apply to their environment, including scoping and remediation capabilities
2. and a need for more automation and integration with SIEM technology (Torres, 2014)

The term 'automation' here corresponds to automating and/or simplifying some or all the works of system administrators, network administrators, security analysts, incident handlers, threat hunters and forensic analysts. What we can automate is the procedures; they are fixed and can be automated. Techniques are systematic procedures which can also be good candidates for automation. Methods which are again systematic and simply the usage of tools can be automated as well.

2.1. Methodologies

At Incident Response and Handling, using well-established methodologies not only simplifies the operations but reduce mistakes as well. There are several guiding methods that decomposes complete process into smaller fractions and define what to do at each of those. These methodologies don't change the way we name the stages of Incident Handling as Identification, Containment, Eradication, Recovery and Lessons Learned, they rather assure quality of performance at each of those stages.

2.1.1. Observe Orient Decide Act (OODA)

"OODA stands for observe-orient-decide-act, and it's what people responding to a cybersecurity incident do constantly, over and over again. We need tools that augment each of those four steps. These tools need to operate in a world of uncertainty, where there is never enough data to know everything that is going on. We need to prioritize understanding, execution, initiative, decentralization and command (Schneier, 2017)."

The OODA loop defined by Colonel John Richard Boyd created a significant breakthrough on modern warfare theory. At his research project, Lieutenant Colonel

Jeffrey N. Rule discusses Boyd's work in detail and provides the missing pieces of information about how such an elegant approach considered tactical only (Rule, 2013). It is easier to understand OODA as a continuous sequence of acts which feed each other both forward and backward.

2.1.2. Cyber Kill Chain, Deception Chain, Moving Target Defense

Created by Lockheed Martin, the "Cyber Kill Chain" defines adversaries' attack routines in seven stages: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control (C&C), and Action on Targets (Scarfone, 2016). Whenever possible, the target needs to stop the adversary at an early stage as much as possible.

A modified model "Deception Chain" (Heckman, Stech, Thomas, Schmoker, & Tsow, 2015) uses deception techniques to slow down the attack or to change the course of action. Deception Chain provides a more appropriate model for understanding, analyzing and collecting the techniques and tools of the adversary although it requires more planning and a better infrastructure to remain intact while monitoring the attacks.

Another technique that can provide a defensive advantage is Moving Target Defense. "Moving Target Defense (MTD) is a security approach used in many common computer systems to help make them less easily compromised. A MTD seeks to provide additional protection to all protected programs even if those programs have known vulnerabilities. It does not seek to fix any particular software vulnerability but, instead, seeks to make any such vulnerability more difficult to exploit (Davidson & Andel, 2017)."

2.1.3. Diamond Model

The Diamond Model establishes the basic atomic element of any intrusion activity, the event, composed of four core features: adversary, infrastructure, capability, and victim (Caltagirone, Pendergast, & Betz, 2013). This model provides benefits in terms of threat intelligence gathering while being easier to understand compared to the use cases of STIX (Barnum, 2014). It is also possible to create diamonds for each separate stage of the Kill Chain to create more granular intelligence.

2.2. Indicators of Compromise (IoC)

The OpenIOC project, an open source initiative founded by Mandiant, defines IOCs as "specific artifacts left by an intrusion, or greater sets of information that allow for the detection of intrusions or other activities conducted by attackers (The OpenIOC Framework, 2017)." According to Chris Sanders, any piece of information that can be used to objectively describe a network intrusion, expressed in a platform-independent manner is an indicator. It might be the IP address of a command and control (C2) server or a complex set of behaviors. (Sanders, & Smith, 2014)

He further categorizes them into two groups as Static and Variable indicators where Static indicators consist of Atomic, Computed, and Behavioral indicators while variable indicators don't have known values but rather appear as a sequence of events (Sanders, & Smith, 2014). Playbooks taking actions depending on loose indicators are prone to errors. Increasing the number and reliability of indicators may create opportunities for taking solid and successful actions.

Behavioral indicators explain 'how' and atomic/computed indicators explain 'what'. If any of those remain the same during the course of infection (or attack) those are considered 'static', otherwise, changing indicators are called 'variable indicators'. Some indicators from Petya ransomware can be listed as follows ("Petya Ransomware Fast Spreading Attack", 2017):

Atomic indicators	
email	wowsmith123456@posteo.net
FilePath	dllhost.dat
Computed indicators	
FileHash-SHA1	34f917aaba5684fbe56d3c57d48ef2a1aa7cf06d
FileHash-SHA1	38e2855e11e353cedf9a8a4f2f2747f1c5c07fcf
FileHash-SHA1	56c03d8e43f50568741704aee482704a4f5005ad

Behavioral indicators	
Execution of “ezvit.exe”	“ezvit.exe” executes two child processes “rundll32.exe” and “UniCrypt.exe”

("New ransomware, old techniques: Petya adds worm capabilities", 2017)

At WannaCry ("WannaCry Infos", 2017), we had seen combined cases of static and variable indicators such as:

Static atomic indicator

FileName mssecsvc.exe

With variable computed indicators (for different file under the same name):

Variable computed indicators

FileHash-SHA1 e14f1a655d54254d06d51cd23a2fa57b6ffdf371cf6b828ee483b1b1d6d21079

FileHash-SHA1 a50d6db532a658ebbebe4c13624bc7bdada0dbf4b0f279e0c151992f7271c726

FileHash-SHA1 9b60c622546dc45cca64df935b71c26dcf4886d6fa811944dbc4e23db9335640

Searching for similarities at various and often disregarded parts of code can provide new insights to malware analysts. CodexGigas named such a work as malware DNA profiling ("codexgigassys (CodexGigas)", 2017). Using deliverables of a work like that (i.e. yara signatures) it becomes easier to detect variants of malware or new works of APTs looking at common practices of producers, such as the tools or naming schemes they are using or even the mistakes they are making.

2.3. Threat Intelligence

Indicators are a subset of ‘observables’. From what you have observed you can filter out the parts of the evidence that indicate undesired activity. Therefore, focusing on observables is more important than indicators, and standards like STIX and TAXII (now

being managed by OASIS) are becoming more important ("OASIS | Advancing open standards for the information society", 2017). The more accurate the observation, the better the decision, which, in turn, will lead to a more effective action (Rule, 2013).

Threat Intelligence deals with observables and indicators. Having detailed information about the tools, techniques and procedures of adversaries, it is easier to profile them. It also makes it easier to deploy protective measures for specific adversaries provided that the threat intelligence data is relevant.

Threat Intelligence can be further categorized into four groups:

Strategic	high-level information, consumed at board level or by other senior decision-makers
Operational	information about specific impending attacks against the organization
Tactical	Tactics, Techniques, and Procedures (TTPs) and information about how threat actors are conducting attacks
Technical	information (or, more often, data) that is normally consumed through technical means

Table 1: Threat Intelligence Categories (Chismon & Ruks, 2016)

2.4. Automation

In Incident Handling, we require ways to automate different tasks. This includes the collection of threat intelligence, mining data for indicators, correlate events and taking appropriate actions in limited time with maximum accuracy. As automation deals with certain steps with mostly well-defined information at hand, it is possible to perform series of actions in batch mode. One can gather that well-defined information from threat intelligence sources (open source and commercial for non-government entities) and automate parts depending on atomic indicators.

In *Rise of the Robots. Technology and the Threat of a Jobless Future*, Ford discusses which jobs will be handled by robots in the future. He mentions “Once one of the industry’s major players begins to gain significant advantages from increased

automation, the others will have little choice but to follow suit. Automation will also offer the ability to compete on dimensions beyond lower labor costs (Ford, 2016).” Those significant advantages are expected to come from ‘actor’ data, playbooks and techniques. If they are not available to public and become available unexpectedly due to a leak such as ShadowBroker’s ("misterch0c/shadowbroker", 2017) negative impact can be high for the unprepared as seen with Wannacry ransomware ("WannaCry/WannaCrypt Ransomware Summary - SANS Internet Storm Center", 2017). Luckily, when threat level gets higher, IOCs, patches and rules are published earlier (if not available already) and if the systems are automated to digest those, they are up to date as soon as feeds are available. This is the significant advantage gained over a competitor; while a competitor assigns resources and money for establishing defensive or recovery mechanisms those who have invested time and money (only for defense) in advance will continue their business uninterrupted.

During the past decade attackers have improved efficiency at a greater rate than defenders (Fonash & Schneck, 2017) and as a result this deficiency needs to be completed with human efforts at defenders’ side. That’s why U.S Bureau of Labor Statistics pointed to hundreds of thousands of unfilled cybersecurity-related jobs in recent years (Olyaei, 2016) and there is no time to train and get them on board. Thus, automation is a need not only for providing the OODA loop feeds correctly but for closing the gap for missing workforce also.

3. Tools and Techniques

“If you know what you’re looking for and where to look, nothing is hidden (Rob Lee cited from Davidoff, & Ham, 2012).”

3.1. A Security Analyst’s Toolset

For an attacker or Pen Tester, a single distro might be enough. The most popular distribution at the time of this writing was ‘Kali’ from Offensive Security.

Unfortunately, Security Analysts don’t have a single tool or distro yet for handing incidents from beginning to the end, and this may never be the case. Doug Burk’s ‘Security Onion’ can be considered as the most beneficial bundled solution for spotting

the incidents quickly as it comes with Snort/Suricata, Bro, OSSEC, Sguil, Squert, ELSA, Xplico, NetworkMiner, and many other security tools already installed. Different tools are available depending on the stage of incident handling, such as Security Onion, CapAnalysis, SOF-ELK for Identification, Phantom for automating tasks at Containment, Eradication and Recovery stages, FIR, OpenSOC or TheHive for most of the stages, and others (see Appendix B for a more detailed list).

Security Onion (Security Onion Solutions, LLC), SANS Investigative Forensic Toolkit (SIFT) Workstation (SANS), SELKS (Stamus Networks), HoneyDrive. Lenny Zeltser's REMnux, Phantom Community (Phantom Cyber), Network Monitor Freemium (LogRhythm), Response Operation Collection Kit (ROCK NSM) (MOCYBER), OSSIM (AlienVault), SOF-ELK (Phil Hagen/SANS FOR572), CapAnalysis (Gianluca Costa), Flowbat (Applied Network Defense), The Hive (Thomas Franco, Saâd Kadhi, Jérôme Leonard), MazeRunner Community Edition (Cymmetria), FIR (Fast Incident Response), Silk on a Box (NetSA), Moloch and OpenSOC (OpenSOC). And this list can be expanded.

3.2. Flows, conversations, normal vs. abnormal

A SOC analyst needs clear understanding of what is considered as normal at a monitored network. When company policies related to the usage of information systems provide extensive privileges to end users there may be no black or whites.

Contrarily, at some networks, usage of the same resources might have been highly restricted. From SOC point of view it is easier to spot anomalies at infrastructures of companies with well-defined policies even when 'noise' or extensive traffic for real world applications are present.

With a software like Network Monitor Freemium (LogRhythm) (despite the license restrictions of the free product,) one can easily spot what is happening at network level and up (Appendix C/Task #2: Installation of Network Monitor Freemium (LogRhythm) for setting up a VM for Freemium).

While there are few things you can do with 1 Gbps data processing and 1 GB packet capture storage limits, you can still enable (from Rules tab) some or all DPA

(Deep Packet Analytics)) rules and after applying them you can quickly find interesting sessions like 'Protocol is not on its appropriate port' (Flow_ProtoMismatchApp) type of traffic. You can also select a field (from Analyze tab) and visualize data for that field easily. Via Diagnostics tab you can check the packet/data/flow rates to figure out expected ranges. (It is also possible to upload pcap and replay it, but for most network forensic investigation cases, it may exceed the limit free license permits.)

Wireshark, tshark or tcpdump can be alternatives. First one will fail to open large pcaps, the second and third will fail to visualize communication. CapAnalysis is a good alternative (although feeding data into it will still require splicing) (Appendix C/Task #4: Installation of CapAnalysis and Appendix C/Task #5 for splitting large pcap into smaller files).

Once uploads are completed, clicking on the dataset name at CapAnalysis will provide several options for viewing, categorizing and filtering data.

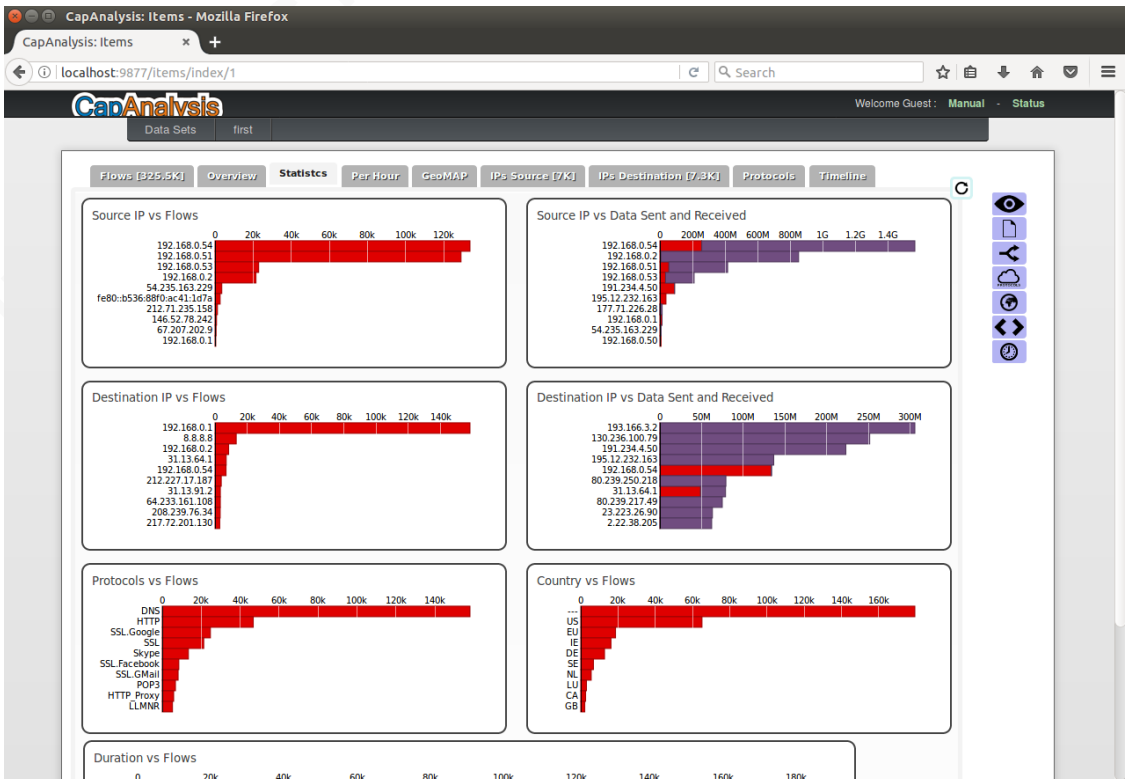

Figure 1: Analyzing packet capture files with CapAnalysis

Murat Cakir, murat.cakir@clf.com.tr

Knowing what to look for (i.e. indicators) rather than trying to understand what really happened is easier for an analyst. Further, it is very easy to filter out special protocols to eliminate background traffic (or noise). Focusing on specific flows and knowing the packet capture (pcap) files that store them, looking inside those specific files is possible. Likewise, it can be done for specific timeframes, countries, source or destination IPs etc. It is an overview with wealthy information. Drilling down up to packet level pcap segment can be exported (through pressing the information icon left to the connection) and investigated with tools like Wireshark.

CapAnalysis is good for performing 'Rinse-Repeat Intrusion Detection' technique (Hjelmvik, 2015) (which is basically removing all 'normal' or whitelisted protocols, traffic, files, IPs etc. and repeating the process from the beginning) partially. Missing features at CapAnalysis are negating applied filters and defining lists or groups.

Tools like CapAnalysis may not explicitly help automating things, but they can help defining the search patterns of Incident Handlers. Once this is done, automation can be provided through a set of defined steps; mostly for drilling down to flows for what is not expected and searching deeper for indicators in what is left.

3.3. Searching for traffic patterns

When we are working with a packet capture file (i.e. in out of band mode) we can use several tools for investigation. Converting a pcap to a flow allows us to search for particular conversations as well as quickly eliminating what we don't care for (Appendix C/Task #7: Installation of SiLK, Appendix C/Task #8: Use sets from scans.io with SiLK, Appendix C/Task #9: Using lists from Malware Domain List, Appendix C/Task #10: Using suspicious domains list from SANS).

When lists are used for protection and blocking, it is preferred to use whitelists (such as Alexa, Umbrella etc.) merged with your lists of trusted IPs. For threat hunting and blocking blacklists, common usage is to combine malicious IPs/domains, suspicious IPs/domains and your lists of (potential adversaries') IPs collected from previous attacks.

Notice that there will be gap between not whitelisted nor suspicious/blacklisted. Thus, at threat hunting and network forensics, it is wiser to discard fully trusted part only and look for the rest.

For practicing different SiLK commands and generating visual representations of them, FlowBat is an appropriate choice. Once, the narrowing process is clear and it is working as expected, it can be automated.

We still need to be careful with using lists as Whitelisted domain lists can still contain malicious domains (Hjelmvik, 2017) or shared hosts can be mistakenly considered as malicious or clean (from Sanders, & Smith, 2014), or domain/IP lookups can return private IPs which can end up blocking your own private IP blocks if they match (Appendix C/Task #8: Use sets from scans.io with SiLK). Thus, correctly sanitizing data is very important. Herman Slatman provides a list of Threat Intelligence sources (Slatman, 2017).

3.4. Searching for signatures and patterns

Extracting parts of conversation, files, attachments and similar observables and checking for indicators can be automated rather easily. “Fortunately, there are many tools that can extract files from network packet captures. Unfortunately, no tool performs perfectly in all situations (Deck, 2015).”

Similarly, checking for hash values for known bads, virus/malware or yara signatures need different tools. We can use tools like foremost, tcpxtract, tcpextract, tshark_extractor, NetworkMiner etc. and then will need to look up for the hashes at Team Cymru, Virustotal, Comodo Valkyrie, Virusshare, etc. either online or via downloaded hash sets. We can script this checking process or can use tools like FileLookup.py.

3.5. Performing search, extract, check near real-time and using of Threat Intelligence feeds

Another straightforward way for extract files, compute and check their hashes process is using Bro (Appendix C/Task #11: Installing Bro) with default configuration. The detect-MHR script will detect file downloads and check corresponding hash values at

Team Cymru's Malware Hash Registry. This will provide a near real-time detection through the interface Bro is monitoring. You can also work on a captured file.

Relying on hash signatures has at least one clear disadvantage: they may not be registered into the Threat Intelligence database you are querying. It might be even possible to craft a file with the same MD5 value of a clean or malicious file (McHugh, 2015). Thus, allowing or blocking based on MD5 hashes should not be considered reliable anymore and sha1 or sha256 should be used instead.

Luckily, Bro is capable of consuming Threat Intelligence Data ("Intelligence Framework — Bro 2.5-152 documentation", 2017) (Appendix C/Task #12 and Appendix C/Task #13 for Installing CIF and Integrating Bro and CIF).

Another way to access and consume (at least query) data from an up to date threat intelligence service is using STAXX from Anomali. (Appendix C/Task #14: STAXX) With that, an Incident Handler can get access to STIXX Threat Intelligence provided via a free service (such as Hail a Taxii) or Anomali's own service.

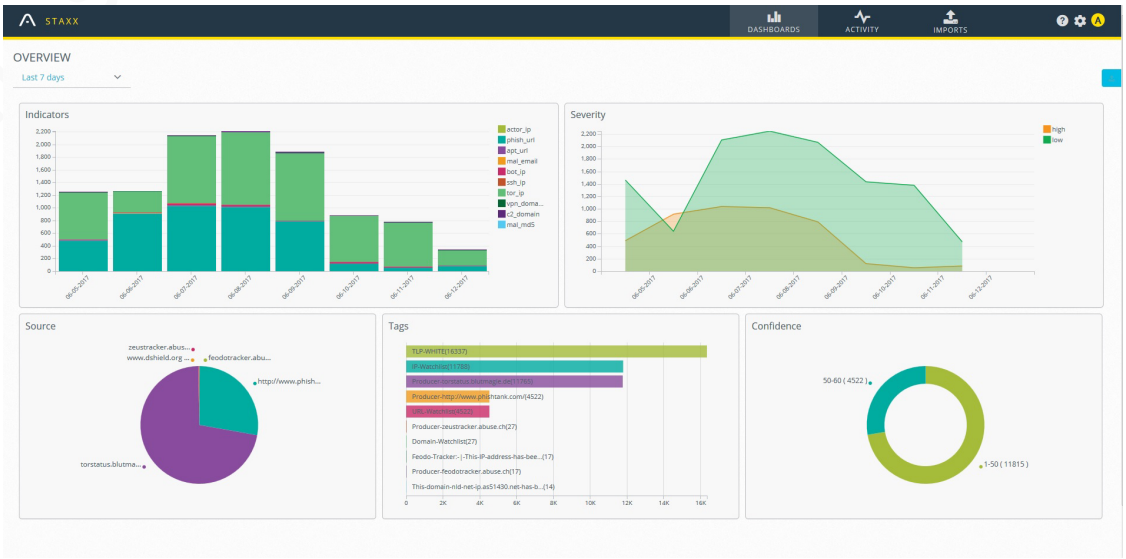

Figure 2: STAXX Dashboard

Murat Cakir, murat.cakir@clf.com.tr

3.6. Towards NSM with built-in Threat Intelligence

There are several projects related to Network Security Monitoring (NSM) architectures with built-in threat intelligence data consumption like Open-NSM (("An NSM Group without borders", 2017) or Response Operation Collection Kit (ROCK NSM) by MOCYBER ("ROCK NSM", 2017). It is possible to evaluate each although (especially latter) few tweaks might be needed for starting services at run time. It might be better to keep an eye on these promising projects and revisit them once they are more mature.

For demonstration purposes, a potential candidate was AlienVault's OSSIM. ("OSSIM: The Open Source SIEM | AlienVault", 2017) It is a very well-known product which we didn't want to re-visit and a similar alternative was OpenSOC ("Open Security Operations Center", 2017). Due to high hardware requirements of OpenSOC we didn't used that either but noted as a possible choice. Also note that a good reference for using end point logs in Splunk can be found at SANS reading room (Eddy, 2017).

An outstanding alternative for us was SELKS ("Open Source | Stamus Networks", 2017). SELKS was built with components like Suricata, ELK (Elastic, Logstash, Kibana) and Scirius from the very beginning (2014) and a similar setup now, inspires a parallel work by Doug Burks for enhancing Security Onion (Burks, 2017) with contribution from works of Phil Hagen (Hagen, 2017).

The good thing with those products is accessing additional interfaces to use with automation; like Kafka, Elastic and Logstash. Having new 'data sources' which we can use to access data we can use those products along with others via automation products like Phantom (Appendix C/Task #15: Evaluate SELKS, Appendix C/Task #16: Security Onion or Security Onion with ELK and Appendix C/Task #17: Phantom).

3.7. Using Threat Intelligence with IPS

Threat Intelligence not only provides information about tools, techniques, procedures, but solid information in terms of atomic IOC data like IP addresses, computed IOC like file hashes and DNS addresses etc. Those can be fed into defense mechanisms using automation. We showed how lists can be imported into IPS for alerting or blocking using Suricata's reputation features (Appendix C/Task #18: Suricata

on Ubuntu) but again becoming a single point of failure is possible when hardware requirements are not considered or lists are not fully trusted.

4. Working with presented tools at real-time Incident Handling, Threat Hunting and Network Forensics Analysis

Incident response is your organization's reaction to any unauthorized, unlawful, or unacceptable activity that occurs on one of your networks or computer systems. Computer forensics is the unearthing of evidence from computer media to support a legal proceeding (Prosise & Mandia, 2003). Network forensics refers to the scientific study of network-based evidence, commonly applied to legal questions (Davidoff & Ham, 2012).

Threat hunting is the analysis for detecting intruders' signs through data and systems (Bejtlich, 2017). It is about placing an appropriate, dedicated focus on the effort by analysts who purposely set out to identify and counteract adversaries that may already be in the environment (Lee & Lee, 2016).

The path from Incident response followed by threat hunting to network and computer forensics involves increasing human interaction at both offense and defense sides. From threat hunting perspective, logs and alerts created by a basic malware, which can obscure the real data needed in a hunt can even be considered as 'noise' (Lee & Lee, 2016).

We selected a case from Forum of Incident Response and Security Teams' 2015 conference ("FIRST.org / 27th Annual FIRST Conference / Program", 2015). Training documentation for the forensic analysis is already provided (Hjelmvik, 2015) (For detailed answers (Davis, 2016))(See Appendix D).

It was challenging for an incident handler to find out all the answers because of the background traffic (in ~4.8 GB (combined) pcap) and automation was only possible with up to date and accessible threat intelligence data. Obfuscated javascript and backdoors were additional problems for the handler.

Apart from finding the answers for the challenge questions, we wanted to verify the following:

a. Was real time prevention possible? (True/False/n.a.)

b. Could we find the answer of this question via an automate investigation? (True/False/n.a.)

c. Considering efforts and costs needed for setting up automation vs. human incident analysis which one would be cost and time beneficial? (A/H)

Even for a case like this, with tools presented in this paper it was possible to reach following conclusions:

- Although it might be hard for specific cases, almost all the investigation work could have been carried out via automation
- For 11 of 14 situations (78,5%) prevention was possible
- Considering the recurrence likelihood of similar situations are high; investing time and effort into automation of investigation work could be beneficial for 12 of 17 (70,5%) of presented situations

Malware and automated attacks and less involvement of humans at the offensive side help us use automation at defensive side more extensively and more reliably. Similarly, when an attack vector converges to 'normal behavior' of a legitimate process, possibility of detection is less likely. Method of using standard system commands and legitimate tools like powershell etc. is very effective, yet, raises less alerts.

"Despite common misconceptions, threat hunting cannot be fully automated. Much of the process and any repeatable steps -searching for known signs of a threat on the network, reusing new threat data and performing other machine-learning tasks, for example- can and should be automated, but there will always be a need for analysts who have instincts and inquisitive minds (Lee & Lee, 2016)." When "threats manifest themselves (SANS Institute, cited from Cima, 2001)" differently, ways to respond to them differ.

5. Orchestration

5.1. Cases where automation is possible but might not be reliable

Adding to many control mechanisms to playbooks not only make them more complicated and slow, but may create single point of failures. Consider cases below which may fail or may be forced to fail:

Case 1: A playbook consuming threat intelligence data from a single service and needs a valid response to continue

Issue(s): Failure in communication will hang the process. A compromised TI server can let the blocked in and may keep the legitimate out.

Case 2: A playbook consuming threat intelligence data containing domain names, runs an automated job to resolve their IPs and uses those to block connections

Issue(s): Private IP addresses returned at DNS queries can match to internal hosts and legitimate connections to internal hosts can be blocked.

Case 3: A playbook using a service which keeps track of exposed credentials for given domain and blocks remote access requests with those username

Issue(s): Pollute monitored sites and forums with real usernames and fake passwords and force playbook to block legitimate users

Case 4: A playbook checking MD5 hash of a malware and depending on the identification it blocks the CnC server connections

Issue(s): Create two different variants of malware of which will have the same MD5 hashes. Let the first variant remain dormant if a connection to CnC is blocked and let the second variant become active if the connection is blocked. Playbook relying on the methods of the first variant will mistakenly activate the second variant.

Case 5: Playbooks feeds Alexa, Umbrella top 1M lists to IPS and it allows no communication but communication with those hosts only.

Issue(s): In case resolved IP address are used by shared hosting sites, hacking into another service served by the same host will allow access.

Murat Cakir, murat.cakir@clf.com.tr

Case 6: Automated mechanism blocks access for IPs that create Slowloris DoS attacks (Nikolic, 2017)

Issue(s): An attacker with ability to intercept and relay communication between two parties one of which is a web server strips the final CRLF from requests. Organization at Web Server side interprets those as attacks and blocks the IP address(es) of legitimate users.

5.2. Role of Products like Phantom

Phantom comes with available playbooks through 'git' as well as two 'case templates'; 'Response Template 1' and 'NIST 800-61' template. Both define Incident Handling properly in slightly different ways while giving the user ability to add tasks and to assign specific tasks to specific users. Given the opportunity, one could track incidents in an almost fully automated way according to valid methodologies. From technology point of view, it is a great benefit for information security administration and management staff but deploying each of the products to be integrated for a specific task is not only expensive but might not be feasible. Community version has limitations, such as performing a maximum of 100 actions per day. From technology point of view, organizations with enough budget can use community know-how and can automate most of their incident response and handling work with a technology like Phantom's.

5.3. A case study with SELKS, Phantom and OTRS

At Appendix C/Task #19 we demonstrated a simple flow that queries alerts from SELKS and creates tickets for each at OTRS. Presented playbook can be improved to handle specific cases only, or OTRS part can be replaced with another asset depending on what the real requirements are. It can also be customized to work with ELK version of Security Onion.

5.4. Artificial Intelligence (AI), Machine Learning and Cognitive Computing

Darktrace's The Enterprise Immune System is using a method inspired by the self-learning intelligence of the human immune system. As human immune system works by learning about what is normal for the body, identifying and neutralizing outliers that

do not fit that evolving pattern of normality, Darktrace uses machine learning and AI to implement the same at cyber defense ("Darktrace | Technology", 2017).

IBM Resilient Incident Response Platform Enterprise uses dynamic action plans and simulations through their Incident Response Platform ("IBM Resilient Incident Response Platform", 2016) and embeds Watson cognitive computing into a product called IBM Cognitive SOC ("IBM Cognitive Security - Watson for Cyber Security", 2017).

DarkLight interprets the data like a human analyst and supports evidence-based decision-making and course of action selection. It also enables organizations to share Programmable Reasoning Objects^{SM} (PRO^{SM}) for analytics and automated courses of action they create for threat hunting, insider threats, false positive reduction ("DarkLight Artificial Intelligence Expert System", 2017).

"Triaging refers to the sorting, categorizing, and prioritizing of incoming incident reports or other CSIRT requests. It can be compared to triage in a hospital, where patients who need to be seen immediately are separated from those who can wait for assistance (Killcrece, Kossakowski, Ruefle & Zajicek, 2003)."

Thus, these tools and platforms will be beneficial -at least- for 'triaging' and incident responders and handlers can work on what tools cannot cope with.

Elastic now introduces a new extension called X-Pack which eases the job of adding machine learning to existing structures. As we have introduced how we can create a flow from suricata event to an actionable item, we can also add machine learning capabilities to SELKS or SecurityOnion with ELK and monitor our infrastructure for anomalies and add new features as future work.

6. Conclusion

"Automation, too, is fixed. Incident response needs to be dynamic and agile, because you are never certain and there is an adaptive, malicious adversary on the other end (Schneier, 2017)". Adaptive offense needs adaptive defense. As Schneier further emphasizes as "You need a response system that has human controls and can modify

itself on the fly. Automation just doesn't allow a system to do that to the extent that's needed in today's environment (Schneier, 2017).” Small changes at offensive side can easily circumvent defensive mechanisms unless observables are investigated at a deeper level for behaviors and/or ‘the DNA’ ("codexgigassys (CodexGigas)", 2017) through properly implemented automation.

Cyber warfare is unconventional and asymmetric warfare (Geers, 2011). The adversary having access to the same threat intelligence resources knows which attacks are more likely to fail and change tactics, tools or procedures. They can be aware of the targets’ tools and techniques used for automation. This means they can into get their OODA loop which creates friction (Rule, 2013). To prevent that, continuous adaptation is needed. It can be implemented in terms of using several different ways and playbooks at automation and by comparing the results. Or deception techniques can be used to change the course of activities and playbooks can run with spoofed indicators.

“The Orient and Decide portions of the OODA loop are internal processes, whereas the Observe and Act portions interact with the external world. While the orient phase is at the heart of the OODA loop, it is the dialectic engine of analyses and synthesis (understanding and creativity) that is at the heart of orientation. It is here where the creative nature of the individual or organization makes it unpredictable (Rule, 2013).” This “orient” phase is where adaptation takes place (at both ends) due to involvement of humans and it is the most complicated part to automate.

Tools used by incident handlers can be considered as musical instruments. They can use different techniques to create the desired output but the output can be the same. Two different instruments from brass and string families can generate the same note using different techniques just like the threat intelligence sources, intrusion detections systems, SIEMs etc. can. But not all instruments can cover the same octaves. Further there is a distinguisher called “timbre” which differentiates the perception of the same note played by different instruments (Constantinsen, 2015).

“The leader in traditional chamber groups is usually the player of the highest-pitched instrument: the first violin in a string quartet, the flutist in a woodwind quintet, or

the first trumpeter in a brass quintet. In a mixed ensemble, the player whose part has priority from a compositional point of view takes over the leadership.

When the task of signaling and cuing becomes complex, or when too many musicians must be led from within an ensemble, a separate person -the conductor- is designated to direct the group and make musical and technical decisions (Meier, 2009)."

Automation triggers the need for orchestration since, parts affect each other and should behave in coordination. Even an adaptive orchestration mechanism should be driven correctly. Conducting security orchestration is a new skill set Incident Handlers and Responders should learn.

To share the transcendent beauty of music the conductor should connect to the composer's thought and emotion (Meier, 2009). Not all composers can be conductors or vice versa. Composers should know how to write the score and conductors should know at least how to read them.

At the automation and orchestration of Incident Handling, the score is the methodology. Incident handler in composer role selects the necessary instruments (i.e. tools) and writes the score (i.e. applies the methodology). During performance, it is again an Incident Handler (this time in conductor role) who ensures the score is played correctly.

Thus, both roles should be handled by Incident Handlers. It may be possible to have external parties who can select methodology, techniques and implement tools for automating and orchestrating the cycle, but at the end the conductor will be a human – at least for another while.

Murat Cakir, murat.cakir@clf.com.tr

References

An NSM Group without borders. (2017). *Open-nsm.net*. Retrieved 12 June 2017, from http://www.open-nsm.net/

Barnum, S. (2014). *STIX Whitepaper | STIX Project Documentation*. *Stixproject.github.io*. Retrieved 5 June 2017, from http://stixproject.github.io/getting-started/whitepaper/

Bejtlich, R. (2017). *The Origin of Threat Hunting*. *Taosecurity.blogspot.com.tr*. Retrieved 13 June 2017, from https://taosecurity.blogspot.com.tr/2017/03/the-origin-of-threat-hunting.html

Burks, D. (2017). *Towards ELK on Security Onion: A Technology Preview*. *Blog.securityonion.net*. Retrieved 12 June 2017, from http://blog.securityonion.net/2017/03/towards-elk-on-security-onion.html

Chismon, D., & Ruks, M. (2016). *Threat Intelligence: Collecting, Analysing, Evaluating*. *Centre for the Protection of National Infrastructure*. Retrieved 13 June 2017, from https://www.ncsc.gov.uk/content/files/protected_files/guidance_files/MWR_Threat_Intelligence_whitepaper-2015.pdf

Cima, S. (2001). *Vulnerability Assessment*. *Sans.org*. Retrieved 5 June 2017, from https://www.sans.org/reading-room/whitepapers/basics/vulnerability-assessment-421

codexgigassys (CodexGigas). (2017). *GitHub*. Retrieved 24 July 2017, from https://github.com/codexgigassys

COMPUTER INCIDENT RESPONSE GUIDEBOOK. (1996). *Csirt.org*. Retrieved 5 June 2017, from http://www.csirt.org/publications/navy.htm

Constantinsen, B. (2015). *What Music Really Is | The Physics of Sound. What Music Really Is*. Retrieved 13 June 2017, from http://whatmusicreallyis.com/research/physics/#timbre

Crowley, C. (2017). *Future SOC: SANS 2017 Security Operations Center Survey*. *Sans.org*. Retrieved 5 June 2017, from https://www.sans.org/reading-room/whitepapers/analyst/future-soc-2017-security-operations-center-survey-37785

Murat Cakir, murat.cakir@clf.com.tr

DarkLight Artificial Intelligence Expert System. (2017). *Darklightcyber.com*. Retrieved 13 June 2017, from https://www.darklightcyber.com/differentiators

Darktrace | Technology. (2017). *Darktrace.com*. Retrieved 13 June 2017, from https://www.darktrace.com/technology/

Davidoff, S., & Ham, J. (2012). *Network forensics* (1st ed.). Upper Saddle River, NJ: Prentice Hall.

Davidson, C., & Andel, T. (2017). Feasibility of Applying Moving Target Defensive Techniques in a SCADA System. In *11th International Conference on Cyber Warfare & Security* (p. 363). Boston: Academic Conferences and Publishing International Limited.

Davis, C. (2016). *Web Defacement and Spear Phishing*. Presentation, Louisville, KY.

Deck, S. (2015). *Extracting Files from Network Packet Captures*. *Sans.org*. Retrieved 8 June 2017, from https://www.sans.org/reading-room/whitepapers/forensics/extracting-files-network-packet-captures-36562

Eddy, E. (2017). *Intrusion detection through traffic analysis from the endpoint using Splunk Stream*. *Sans.org*. Retrieved 13 June 2017, from https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-traffic-analysis-endpoint-splunk-stream-37800

FIRST.org / 27th Annual FIRST Conference / Program. (2015). *First.org*. Retrieved 5 June 2017, from https://www.first.org/conference/2015/program

Fonash, P., & Schneck, P. (2017). Cybersecurity: From Months to Milliseconds. The Johns Hopkins University Applied Physics Laboratory LLC.. Retrieved 3 July 2017, from https://secwww.jhuapl.edu/IACD/Resources/Reference_Materials/07030229_Cybersecurity_From_Months_to_Milliseconds_2015-2.pdf

Ford, M. (2016). *Rise of the robots* (1st ed.). New York: Basic Books.

Hagen, P. (2017). *SOF-ELK*. *GitHub*. Retrieved 12 June 2017, from https://github.com/philhagen/sof-elk

Heckman, K., Stech, F., Thomas, R., Schmoker, B., & Tsow, A. (2015). *Cyber denial, deception and counter deception* (1st ed.). Virginia: Springer.

Murat Cakir, murat.cakir@clf.com.tr

Hjelmvik, E. (2015). Hands - on Network Forensics. First. Retrieved 6 June 2017, from https://www.first.org/resources/papers/conf2015/first_2015_-_hjelmvik-_erik_-_hands-on_network_forensics_20150604.pdf

Hjelmvik, E. (2015). Rinse-Repeat Intrusion Detection - NETRESEC Blog. Netresec. Retrieved 6 June 2017, from https://www.netresec.com/?page=Blog&month=2015-08&post=Rinse-Repeat-Intrusion-Detection

Hjelmvik, E. (2017). *Domain Whitelist Benchmark: Alexa vs Umbrella - NETRESEC Blog*. *Netresec*. Retrieved 8 June 2017, from http://www.netresec.com/?page=Blog&month=2017-04&post=Domain-Whitelist-Benchmark%3A-Alexa-vs-Umbrella

IBM Cognitive Security - Watson for Cyber Security. (2017). *IBM*. Retrieved 13 June 2017, from https://www.ibm.com/security/cognitive/

IBM Resilient Incident Response Platform. (2016). *Www-01.ibm.com*. Retrieved 13 June 2017, from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WGD03098USEN

Intelligence Framework — Bro 2.5-152 documentation. (2017). *Bro.org*. Retrieved 9 June 2017, from https://www.bro.org/sphinx-git/frameworks/intel.html

Johns Hopkins University Applied Physics Laboratory. (2017). *March 2017 Integrate d Adaptive Cyber Defense (IACD) Orchestration Thin Specification. Johns Hopkins University Applied Physics Laboratory*. Retrieved 5 June 2017, from https://secwww.jhuapl.edu/IACD/Resources/Specifications/IACD_Orchestration_Thin_Specification.pdf

Killcrece, G., Kossakowski, K., Ruefle, R., & Zajicek, M. (2003). *Organizational Models for Computer Security Incident Response Teams (CSIRTs)*. *Resources.sei.cmu.edu*. Retrieved 13 June 2017, from http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=6295

Kral, P. (2011). *Incident Handler's Handbook*. *Sans.org*. Retrieved 5 June 2017, from https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

Murat Cakir, murat.cakir@clf.com.tr

Lee, R., & Lee, R. (2016). *The Who, What, Where, When, Why and How of Effective Threat Hunting*. *Sans.org*. Retrieved 13 June 2017, from https://www.sans.org/reading-room/whitepapers/analyst/who-what-where-when-effective-threat-hunting-36785

McHugh, N. (2015). *Create your own MD5 collisions*. *Natmchugh.blogspot.com.tr*. Retrieved 9 June 2017, from https://natmchugh.blogspot.com.tr/2015/02/create-your-own-md5-collisions.html

Meier, G. (2009). *The score, the orchestra and the conductor* (1st ed.). Oxford: Oxford University Press.

misterch0c/shadowbroker. (2017). GitHub. Retrieved 3 July 2017, from https://github.com/misterch0c/shadowbroker

Merriam-Webster dictionary (2017). *Merriam-webster.com*. Retrieved 5 June 2017, from https://www.merriam-webster.com/

New ransomware, old techniques: Petya adds worm capabilities. (2017). *Windows Security*. Retrieved 24 July 2017, from https://blogs.technet.microsoft.com/mmpc/2017/06/27/new-ransomware-old-techniques-petya-adds-worm-capabilities/

Nikolic, A. (2017). *http-slowloris-check NSE Script*. *Nmap.org*. Retrieved 13 June 2017, from https://nmap.org/nsedoc/scripts/http-slowloris-check.html

OASIS | Advancing open standards for the information society. (2017). *Oasis-open.org*. Retrieved 5 June 2017, from https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti

Oltsik, J. (2016). *Anticipating the RSA Security Conference*. *Network World*. Retrieved 5 June 2017, from http://www.networkworld.com/article/3036160/security/anticipating-the-rsa-security-conference.html

Olyaei, S. (2016). *The Cybersecurity Talent Shortage.... is a myth? - Sam Olyaei*. *Gartner*. Retrieved 5 June 2017, from http://blogs.gartner.com/sam-olyaei/2016/12/21/cybersecurity-talent-shortage-myth/

Online Business Dictionary - BusinessDictionary.com. (2017). *Businessdictionary.com*. Retrieved 5 June 2017, from http://www.businessdictionary.com

Murat Cakir, murat.cakir@clf.com.tr

Open Security Operations Center. (2017). *Opensoc.github.io*. Retrieved 12 June 2017, from http://opensoc.github.io/

Open Source | Stamus Networks. (2017). *Stamus-networks.com*. Retrieved 12 June 2017, from https://www.stamus-networks.com/open-source/

OSSIM: The Open Source SIEM | AlienVault. (2017). *Alienvault.com*. Retrieved 12 June 2017, from https://www.alienvault.com/products/ossim

Petya Ransomware Fast Spreading Attack. (2017). *AlienVault Open Threat Exchange*. Retrieved 24 July 2017, from https://otx.alienvault.com/pulse/59525e7a95270e240c055ead/

Prosise, C., & Mandia, K. (2003). *Incident Response* (1st ed.). Emeryville, USA: McGraw-Hill Professional Publishing.

ROCK NSM. (2017). *Rocknsm.io*. Retrieved 12 June 2017, from http://rocknsm.io/

Rule, J. (2013). *A Symbiotic Relationship: The OODA Loop, Intuition, and Strategic Thought*. *Handle.dtic.mil*. Retrieved 5 June 2017, from http://handle.dtic.mil/100.2/ADA590672

Sanders, C., & Smith, J. (2014). *Applied network security monitoring* (1st ed.). Waltham, MA: Syngress, an imprint of Elsevier.

Scarfone, K. (2016). *The Hunter's Handbook: Endgame's Guide to Adversary Hunting* (1st ed.). Annapolis: CyberEdge Group, LLC.

Schneier, B. (2017). *Essays: Security Orchestration for an Uncertain World - Schneier on Security*. *Schneier.com*. Retrieved 5 June 2017, from https://www.schneier.com/essays/archives/2017/03/security_orchestrati.html

Slatman, H. (2017). A curated list of Awesome Threat Intelligence resources. GitHub. Retrieved 8 June 2017, from https://github.com/hslatman/awesome-threat-intelligence

The OpenIOC Framework. (2017). *Openioc.org*. Retrieved 5 June 2017, from http://www.OpenIOC.org

Torres, A. (2014). *Incident Response: How to Fight Back*. *Sans.org*. Retrieved 5 June 2017, from https://www.sans.org/reading-room/whitepapers/analyst/incident-response-fight-35342

Murat Cakir, murat.cakir@clf.com.tr

WannaCry Infos. (2017). *Docs.google.com*. Retrieved 24 July 2017, from https://docs.google.com/spreadsheets/d/1XNCCiiwpIfW8y0mzTUdLLVzoW6x64hkHJ29hcQW5deQ/pubhtml

WannaCry/WannaCrypt Ransomware Summary - SANS Internet Storm Center. (2017). SANS Internet Storm Center. Retrieved 3 July 2017, from https://isc.sans.edu/forums/diary/WannaCryWannaCrypt+Ransomware+Summary/22420/

Murat Cakir, murat.cakir@clf.com.tr

Appendix A
Terms and Definitions

Here are the definitions for the terms used in this paper:

Technique : A systematic procedure, formula, or routine by which a task is accomplished (Online Business Dictionary, 2017).

Procedure : A fixed, step-by-step sequence of activities or course of action (with definite start and end points) that must be followed in the same order to correctly perform a task. Repetitive procedures are called routines (Online Business Dictionary, 2017).

Method : An established, habitual, logical, or prescribed practice or systematic process of achieving certain ends with accuracy and efficiency, usually in an ordered sequence of fixed steps (Online Business Dictionary, 2017).

Methodology : A system of broad principles or rules from which specific methods or procedures may be derived to interpret or solve different problems within the scope of a particular discipline. Unlike an algorithm, a methodology is not a formula but a set of practices (Online Business Dictionary, 2017).

Tool : An item or implement used for a specific purpose. A tool can be a physical object such as mechanical tools including saws and hammers or a technical object such as a web authoring tool or software program (Online Business Dictionary, 2017).

Incident : Untoward event which (depending on the circumstances) may lead to a damage, disaster, or loss (Online Business Dictionary, 2017).

Event : Occurrence happening at a determinable time and place, with or without the participation of human agents. It may be a part of a chain of occurrences as an effect of a preceding occurrence and as the cause of a succeeding occurrence (Online Business Dictionary, 2017).

Indicator : Measurable variable used as a representation of an associated (but non-measured or non-measurable) factor or quantity. For example, consumer price index (CPI) serves as an indicator of general cost of living which consists of many factors some of which are not included in computing CPI (Online Business Dictionary, 2017).

Observable : capable of being observed, discernible (Merriam-Webster dictionary, 2017).

Murat Cakir, murat.cakir@clf.com.tr

Appendix B

Below are some tools and distros that can be used at Incident Response:

Security Onion (Security Onion Solutions, LLC): Ubuntu based linux distro that contains Snort, Suricata, Bro, OSSEC, Sguil, Squert, ELSA, Xplico, NetworkMiner, and many other security tools already installed. Security Onion is mostly used for Network Security Monitoring. Website: https://securityonion.net/

SELKS (Stamus Networks): Debian based distro from Stamus Networks bundles Suricata, Logstash, Kibana, ElasticSearch together. It uses Scirius for Suricata ruleset management and EveBox for alert and event management. Similar to SecurityOnion it provides an easy to setup Network Security Monitoring environment. Website: https://www.stamus-networks.com/open-source/#selks

Bro (The Bro Project): Bro Network Security Monitor is a network analysis framework originally been developed by Vern Paxson. It's ultimate aim is to provide clear visibility for what is happening on the network with the help of its event-driven scripting language. Website: https://www.bro.org/

SANS Investigative Forensic Toolkit (SIFT) Workstation (SANS): SANS FOR 508 (Advanced Incident Response) training's official distro SIFT is Ubuntu based and contains various tools like: log2timeline (Timeline Generation Tool), Rekall Framework (Memory Analysis), Volatility Framework (Memory Analysis), Autopsy (GUI Front-End for Sleuthkit), PyFLAG (GUI Log/Disk Examination) along with many others. Website: https://digital-forensics.sans.org/community/downloads

HoneyDrive (Ioannis Koniaris): HoneyDrive comes with components like Kippo SSH honeypot, Kippo-Graph, Kippo-Malware, Kippo2MySQL, Dionaea malware honeypot, DionaeaFR, Amun, Glastopf, Wordpot, Conpot SCADA/ICS honeypot and related scripts. It can be used at unallocated and properly monitored IPs to collect threat intelligence. Website: http://bruteforcelab.com/honeydrive

REMnux (Lenny Zeltser): REMnux is the malware analysts' favorite distro. It contains every tool needed for analysis, examination and investigation of malware and memory images. Website: https://remnux.org/

Murat Cakir, murat.cakir@clf.com.tr

Phantom Community (Phantom Cyber): Phantom Cyber's Community edition is a limited edition of Phantom Platform which automates and orchestrates incident handling and response using vendor+community provided playbooks. Website: https://www.phantom.us/community/

Network Monitor Freemium (LogRhythm): Network Monitor provides fast and clear network visibility to network and security administrators including search-based alerting features. Website: https://logrhythm.com/network-monitor-freemium/

Response Operation Collection Kit (ROCK NSM)(MOCYBER): ROCK, along with Suricata and Bro, uses Stenographer and performs full packet capture. It also has features like recursive file scanning. Website: http://rocknsm.io/

OSSIM (AlienVault): Open source SIEM OSSIM is used widely. It differentiates itself with community support and behavioral monitoring capability. Website: https://www.alienvault.com/products/ossim

SOF-ELK (Phil Hagen/SANS FOR572): Another distro from SANS courses, SOF-ELK is the inspiration behind Security Onions' direction into ELK usage. It is a network forensics software, also provided as a virtual image. Website: https://github.com/philhagen/sof-elk

CapAnalysis (Gianluca Costa): CapAnalysis is one of the best tools that can visually explain what is in packet capture files, especially when they are large. Website: http://www.capanalysis.net/ca/

Moloch: To store large scale packet capture data, index and serve them Moloch is one of the best tools available. It can also create views from selections which make it easier for the analyst to quickly pin point weird connections. Website: http://molo.ch/

Flowbat (Applied Network Defense): Flowbat is a tool for working with flow data to query and analyze them. It also works with packet captures in the same manner and makes it easier to execute complex SiLK commands. Website: http://www.flowbat.com/

SiLK on a Box (NetSA): Carnegie Mellon's CERT provides System for Internet Level Knowledge (SiLK) and YAF together to create an environment for network

flow data collection and analysis. Website: https://tools.netsa.cert.org/confluence/pages/viewpage.action?pageId=3571714

The Hive (Thomas Franco, Saâd Kadhi, Jérôme Leonard): The Hive is powerful at analysis due to usage of Cortex, but it also helps collaboration inside IR teams. Website: http://thehive-project.org/

FIR (Fast Incident Response): Commonly, CERTs always try to develop tools for their own needs, as well as the needs of the environment. FIR is developed by CERT Société Générale for easier creation, tracking, and reporting of cybersecurity incidents. Website: https://github.com/certsocietegenerale/FIR

OpenSOC (OpenSOC): OpenSOC uses special network cards, Apache Flume, Kafka and Hadoop to deal with extensive packet captures and network traffic creating a big data environment. Luckily it is scalable although even the very basic setup has extensive requirements. Website: http://opensoc.github.io/

MazeRunner Community Edition (Cymmetria): MazeRunner uses deception based kill chain method for tracking and isolating attackers and it mainly depends on careful planting of breadcrumbs as part of a deception campaign. It has STIX/TAXII integration and (at Community Edition) decoys run on Linux only. Website: https://community.cymmetria.com/

Appendix C
Tasks

Be warned that the steps below are created for demonstrating what we want to achieve as simple as possible. They may even cause harm as we work on older versions of some software and operating systems. Follow official guidelines and best practices for implementing those techniques.

TASK #1: Installation of Oracle VirtualBox

All virtual machines described in this paper are created with Oracle VirtualBox unless they are downloaded as already made virtual machines.

To download and install VirtualBox please follow the instructions located at https://www.virtualbox.org/ as well as the VirtualBox User Manual (https://www.virtualbox.org/manual/UserManual.html)

Most of the machines will need 4-8 GB of RAM and several cores, it is preferred to use a host machine with 16 GB or more RAM and several CPU cores.

TASK #2: Installation of Network Monitor Freemium (LogRhythm)

(Complete Task #1 and install VirtualBox)

Download CentOS 7 based Network Monitor Freemium from https://logrhythm.com/network-monitor-freemium/ Extract FreemiumNetMon.zip you've downloaded and double click on the vbox file to import it to VirtualBox. Tune settings down if default 8GB RAM is too much for you. Log in as `logrhythm` with a password of `changeme`. Use the command: `ip address` to display the IP address obtained from DHCP. Connect to web interface using https and the displayed IP address with credentials: `admin/changeme`.

TASK #3: Installation a Ubuntu Server

(Complete Task #1 and install VirtualBox)

From, http://releases.ubuntu.com/14.04/ download the iso (http://releases.ubuntu.com/14.04/ubuntu-14.04.5-server-amd64.iso), create a virtual machine with 3 GB RAM and 30 GB disk space (disk type: vmdk), select network as

NAT, mount iso image as Live CD, and when prompted select '`Install Ubuntu Server`', '`Use entire disk and setup LVM`', '`Install security updates automatically`', add '`OpenSSH server`' as an additional service, follow the screens and when prompted, eject the installation media and it should be ready. You can copy this vm as your bare Ubuntu server VM without window manager.

To continue, execute:

```
sudo apt-get update
```

```
sudo apt-get install --no-install-recommends ubuntu-desktop
```

for Unity desktop installation without extras. Alternatives can be Lubuntu or LXDE desktop depending on your preference.

```
sudo reboot
```

Login again, choose '`Install VirtualBox additions`' from VirtualBox menu, click on run (enter root password) and you can configure bidirectional clipboard usage as well as shared folders.

Press `Ctrl-Alt-T` and your gnome-terminal should be available. To have additional utilities, you can run:

```
sudo apt-get install konsole
```

which will provide another terminal (konsole) with better copy-paste capabilities, and

```
sudo apt-get install firefox
```

for having firefox browser. Issue;

```
sudo poweroff
```

to complete installation.

For newer versions, process is similar. E.g. http://releases.ubuntu.com/17.04/ubuntu-17.04-server-amd64.iso is the installation media for 17.04.

TASK #4: Installation of CapAnalysis

(Complete Task #3 and install an Ubuntu Server)

Follow: http://www.capanalysis.net/ca/how-to-install-capanalysis and start with:

```
sudo apt-get install gdebi wireshark
```

(wireshark is for further analysis)

Download ubuntu 64-bit package from http://www.capanalysis.net/ca/#download

run `gdebi-gtk` , find downloaded package, install

run

```
sudo /etc/init.d/capanalysis restart
```

and visit http://localhost:9877

Share a folder with host and locate our big pcap file at that folder.

Go to Datasets at CapAnalysis, create a new dataset, and click on files, browsing the folder where you mounted the shared folder, select and upload the pcap.

If pcap file is > 500 MB follow TASK #5 to split it into smaller files and upload those files.

TASK #5: Installation of PcapPlusPlus

We will use PcapSplitter from PcapPlusPlus packate. To do that, issue the commands below at your Ubuntu server:

```
apt-get install git libpcap0.8-dev make g++
cd /opt
git clone https://github.com/seladb/PcapPlusPlus.git
cd PcapPlusPlus
./configure-linux.sh
```

(answer '`n`' for PF_RING and DPDK support since we won't use those)

```
make
cd Examples/PcapSplitter
make
cd Bin
```

Murat Cakir, murat.cakir@clf.com.tr

(Usage: `PcapSplitter -f large_pcap_file -o output_dir -m file-size -p size_in_bytes`)

```
./PcapSplitter -f /mnt/large.pcap -f pcap_file -o /mnt -m file-size -p 500000000
```

TASK #6: The Hive

(Complete Task #3 and install an Ubuntu Server)

(Ref: https://github.com/CERT-BDF/TheHiveDocs/blob/master/installation/deb-guide.md)

```
echo 'deb https://dl.bintray.com/cert-bdf/debian any main' | sudo tee -a /etc/apt/sources.list.d/thehive-project.list

sudo apt-key adv --keyserver hkp://pgp.mit.edu --recv-key 562CBC1C

sudo apt-get update

sudo apt-get install thehive

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key D88E42B4

echo "deb https://packages.elastic.co/elasticsearch/2.x/debian stable main" | sudo tee -a /etc/apt/sources.list.d/elasticsearch-2.x.list

sudo apt-get update && sudo apt-get install elasticsearch

cat <<EOT >> /etc/elasticsearch/elasticsearch.yml

network.host: 127.0.0.1

script.inline: on

cluster.name: hive

threadpool.index.queue_size: 100000

threadpool.search.queue_size: 100000

threadpool.bulk.queue_size: 1000

EOT
```

Murat Cakir, murat.cakir@clf.com.tr

```
service elasticsearch restart

sudo mkdir /etc/thehive

(cat << _EOF_

# Secret key

# ~~~~~

# The secret key is used to secure cryptographics functions.

# If you deploy your application to several instances be sure to
use the same key!

play.crypto.secret="$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' |
fold -w 64 | head -n 1)"

_EOF_

 ) | sudo tee -a /etc/thehive/application.conf

cd /opt/thehive

bin/thehive -Dconfig.file=/etc/thehive/application.conf &
```

Browse to http://localhost:9000

Update database

Create admin user

Login with those credentials

TASK #7: Installation of SiLK

(Complete Task #2 and install Ubuntu)

For a complete SiLK suite installation visit
https://tools.netsa.cert.org/confluence/pages/viewpage.action?pageId=23298051

```
cd /tmp/

sudo apt-get -y install libglib2.0-dev dc glib2

sudo apt-get -y install libpcap-dev

wget http://tools.netsa.cert.org/releases/silk-3.11.0.1.tar.gz
```

Murat Cakir, murat.cakir@clf.com.tr

```
wget http://tools.netsa.cert.org/releases/libfixbuf-1.7.1.tar.gz
wget http://tools.netsa.cert.org/releases/yaf-2.7.1.tar.gz
cd ~/tmp
tar -zxvf libfixbuf-1.7.1.tar.gz
cd libfixbuf-1.7.1
./configure && make
sudo make install
cd ~/tmp
tar -zxvf yaf-2.7.1.tar.gz
cd yaf-2.7.1
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
./configure --enable-applabel
make
sudo make install
cd ~/tmp
tar -xvzf silk-3.11.0.1.tar.gz
cd silk-3.11.0.1
./configure  --with-libfixbuf=/usr/local/lib/pkgconfig/  --with-
python  --enable-ipv6
make
sudo make install
cat <<EOF >>silk.conf
/usr/local/lib
/usr/local/lib/silk
EOF
sudo mv silk.conf /etc/ld.so.conf.d/
```

Murat Cakir, murat.cakir@clf.com.tr

```
sudo ldconfig
```

With this bundle installed, you can use conversion tools like rwp2yaf2silk (from pcap to flow) as:

```
rwp2yaf2silk --in=forensics.snort.pcap --out=forensics.flow
```

TASK #8: Use sets from scans.io with SiLK

(Complete Task #2 and install Ubuntu)

(Complete Task #7 and install SilK)

```
apt-get install grepcidr liblz4-tool
```

Script below will download a (daily) set from scans.io, uses grepcidr to filter out private IP blocks and creates a set to use with SiLK. As this is considered as a whitelist, use appropriately such as filtering out white traffic to have a smaller flow file (i.e. `--anyset= alx1mip.set --fail=newflow_wo_whitelist_traffic`)

```
#!/bin/bash
#
# Data is provided via https://scans.io/series/alexa-dl-top1mil
# Please agree their tems for usage
#
set -x
locate grepcidr > /dev/null || { echo "grepcidr missing, install grepcidr package" ; exit 1; }
locate lz4 > /dev/null || { echo "lz4 missing, install liblz4-tool package" ; exit 2; }
locate rwsetbuild > /dev/null || { echo "rwsetbuild missing, install SiLK" ; exit 3; }
top1mwwwurl=`curl -s https://scans.io/series/alexa-dl-top1mil | grep -m 1 a-www | cut -d\" -f2`
top1mwww=`echo $top1mwwwurl | cut -d/ -f5`
```

Murat Cakir, murat.cakir@clf.com.tr

```
test -f $top1mwww ||  wget $top1mwwwurl

tempfile="alexa-www-$RANDOM"

test -f "gf.txt" || cat <<EOF >> gf.txt

0.0.0.0

127.0.0.1

10.0.0.0/8

172.16.0.0/12

192.168.0.0/16

EOF

lz4 -d $top1mwww

top1mwwwjson=`echo ${top1mwww} | sed s/.lz4//`

cat $top1mwwwjson  | grep -o '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-
9]\{1,3\}\.[0-9]\{1,3\}' > $tempfile

cat $tempfile | sort -u | grepcidr -v -f gf.txt > alx1mip.txt

test -f "alx1mip.set" && rm alx1mip.set

rwsetbuild alx1mip.txt alx1mip.set

rm $tempfile

echo alx1mip.set built from Alexa Top 1M IP addresses data from
scan.io
```

TASK #9: Using lists from Malware Domain List

(Complete Task #7 and install SiLK)

```
apt-get install dos2unix
```

(from Sanders, & Smith, 2014)

```
curl http://www.malwaredomainlist.com/hostslist/ip.txt | dos2unix
> mdl.iplist

rwsetbuild mdl.iplist mdl.iplist.set
```

Then we can check which communications with malware domains took place as;

```
rwfilter /mnt/forensics.flow --anyset=mdl.iplist.set --
pass=stdout | rwcut
```

(or pass it to a file)

TASK #10: Using suspicious domains list from SANS

(Complete Task #7 and install SiLK)

(Via https://isc.sans.edu/suspicious_domains.html)

```
apt-get install cpanminus
```

```
/usr/bin/perl -MCPAN -e 'install Net::CIDR'
```

```
wget http://www.cpan.org/authors/id/R/RA/RAYNERLUC/cidr2range-
0.9.pl
```

```
for i in `cat block.txt | grep -v '\#\|Start' | awk -F$'\t'
'{print $2 "/" $3}'`; do perl cidr2range-0.9.pl -l $i >>
suspicious.txt; done
```

Then you can convert this to a SiLK set as above and use similarly.

TASK #11: Installing Bro

(Complete Task #2 and install Ubuntu 17.04)

(Follow https://www.bro.org/download/beta-packages.html)

```
sudo sh -c "echo 'deb
http://download.opensuse.org/repositories/network:/bro/xUbuntu_17
.04/ /' > /etc/apt/sources.list.d/bro-beta.list"
```

(For Ubuntu 14.04 line will look like;

```
sudo sh -c "echo 'deb
http://download.opensuse.org/repositories/network:/bro/xUbuntu_14
.04/ /' > /etc/apt/sources.list.d/bro.list"
```

)

```
sudo apt-get update
```

```
sudo apt-get install bro-beta
```

Murat Cakir, murat.cakir@clf.com.tr

(For Ubuntu 14.04 line will look like;

```
sudo apt-get -y install bro
```

)

Edit

```
sudo vi /opt/bro/etc/node.cfg
```

and update interface name to reflect the interface you are listening.

Run

```
sudo /opt/bro/bin/broctl
```

and at "`[BroControl] >`" prompt, execute,

```
install
start
status
```

It should be up and running.

(https://www.bro.org/bro-exchange-2013/exercises/faf.html)

Download https://www.bro.org/static/exchange-2013/faf-exercise.pcap and run:

```
cd /tmp
/opt/bro/bin/bro -r faf-exercise.pcap
/opt/bro/share/bro/policy/frameworks/files/hash-all-files.bro
```

and Bro logs should be available in current working directory where files.log shows the file types and their hashes along with the communication endpoints related to these conversations.

```
cat <<EOF > local.bro
# Detect SHA1 sums in Team Cymru's Malware Hash Registry.
@load /opt/bro/share/bro/policy/frameworks/files/detect-MHR.bro
/opt/bro/bin/bro -Cr /mnt/forensics.snort.pcap local.bro
"Site::local_nets += { 10.0.0.0/24 }"
```

Murat Cakir, murat.cakir@clf.com.tr

This will populate entries at `notice.log` when Cymru database contain hashes for them.

TASK #12: CIF Installation

(Complete Task #2 and install Ubuntu 14.04)

(Ref: https://github.com/csirtgadgets/massive-octo-spice/wiki/PlatformUbuntu)

```
curl -Ls https://raw.githubusercontent.com/csirtgadgets/massive-octo-spice/master/hacking/platforms/easybutton_curl.sh | sudo bash -
```

(command above will take time from 10 minutes up to 30 minutes depending on your machine's hardware configuration)

```
sudo chown `whoami`:`whoami` ~/.cif.yml
sudo service monit stop
sudo service cif-smrt stop
sudo -u cif /opt/cif/bin/cif-smrt --testmode
```

(again, above will take time)

```
sudo service cif-smrt start
sudo service monit start
```

test as;

```
cif --cc US
```

TASK #13: Bro - CIF Integration

(Complete Task #11 and install Bro for Ubuntu 14.04)

(Complete Task #12 and install CIF)

Please note that even the small instance requires 16 GB of RAM and 8 cores (Ref: https://github.com/csirtgadgets/massive-octo-spice/wiki)

(Ref: Ismael Valenzuela, http://blog.opensecurityresearch.com/2014/03/identifying-malware-traffic-with-bro.html)

Murat Cakir, murat.cakir@clf.com.tr

```
sudo vi /opt/bro/share/bro/site/local.bro

(append the lines below to the end of the file)

@load frameworks/intel/seen

@load frameworks/intel/do_notice

@load policy/integration/collective-intel

redef Intel::read_files += {

      "/opt/bro/feeds/domain-malware.intel",

};

sudo mkdir /opt/bro/feeds

cif --feed --otype fqdn --confidence 65 --tags malware --format
bro > domain-malware.intel

sudo cp domain-malware.intel /opt/bro/feeds/

sudo /opt/bro/bin/broctl
```

(issue '`restart`' command)

Get a sample file; (via the same reference page above)

```
wget https://raw.githubusercontent.com/aboutsecurity/Bro-
samples/master/sample4/sample4.pcap

/opt/bro/bin/bro -r sample4.pcap
/opt/bro/share/bro/site/local.bro
```

And at intel.log should contain entries related to malware domains. (if due to confidence level or removal of the domain, log entry is missing, you can try an older file located with the example at https://raw.githubusercontent.com/aboutsecurity/Bro-samples/master/CIF/domain-malware.intel)

TASK #14: STAXX

Get, extract and run Anomali_STAXX_v2.3.ova

Note the URL for UI and connect with the credentials admin changeme

(or `anomali/anomalistaxx` for shell access)

Murat Cakir, murat.cakir@clf.com.tr

Use http://hailataxii.com/taxii-discovery-service

Anonymous connections are accepted but you can also use; HTTP-Basic authentication with the credentials: user=`guest`, password=`guest`.

Configure the feeds and accept EULA – done

By default, the Anomali uses Anomali Limo site (https://limo.anomali.com/api/v1/taxii/taxii-discovery-service/) to get Anomali's repository of threat intelligence feeds in STIX format.

One remark: 'free text search' retrieves the items beginning with search term only

TASK #15: Evaluate and install SELKS

Visit https://www.stamus-networks.com/open-source/#selks, and download

http://dl.stamus-networks.com/selks/SELKS-3.0-desktop.iso

Configure a VM as: Debian 64, 4 GB RAM, 30 GB disk and 2 ethernet cards

Check live mode, running from iso image. (Then you can clone from https://github.com/StamusNetworks/SELKS)

Default OS user credentials are user: `selks-user` / password: `selks-user` (password in Live mode is live). The default root password is `StamusNetworks`

Run `Scirius` and add new sources easily (from sources at top menu)

For installation to disk, choose graphical install, follow the prompts.

When installation is finished, edit `/etc/elasticsearch/elasticsearch.yaml` (as root) and replace,

`network.bind_host: 127.0.0.1 to network.bind_host: 0.0.0.0` to make elastic listen on all interfaces. Issue;

```
service elasticsearch restart
```

to make the change effective. (do not use in production environments)

TASK #16: Security Onion or Security Onion with ELK

(Visit http://blog.securityonion.net/2017/03/towards-elk-on-security-onion.html

Murat Cakir, murat.cakir@clf.com.tr

for Security Onion with ELK)

Suggested hardware is:

2 CPU cores, 4GB RAM, 20GB virtual hard drive, (1) management interface with full Internet access, (1) sniffing interface (separate from management interface)

Download https://github.com/Security-Onion-Solutions/security-onion/releases/download/v14.04.5.2/securityonion-14.04.5.2.iso

And follow; https://github.com/Security-Onion-Solutions/security-onion/wiki/Installation

After first reboot, click setup and complete network interface configuration: Reboot and poweroff

clone the virtual machine since usual SO installation will continue in Production mode and ELK version will be based on Evaluation mode.

For usual Securityonion installation:

Chose Production, Standalone create the user

Select Suricata as the engine, with ET ruleset

Install VBOX additions if you like

For Securityonion ELK, start with cloned vm

Install VBOX additions if you like

Reboot

Setup for evaluation

(reboot not necessary)

Download the script:

```
wget https://raw.githubusercontent.com/Security-Onion-Solutions/elastic-test/master/securityonion_elsa2elastic.sh
```

Run the script with sudo privileges:

```
sudo bash securityonion_elsa2elastic.sh
```

and access the GUI via browser at: https://localhost/app/kibana

Murat Cakir, murat.cakir@clf.com.tr

TASK #17: Phantom

Download Phantom from https://www.phantom.us/community/ and follow https://my.phantom.us/docs/quickstart

Default administrative user is '`admin`' with a password of '`password`'.

Note that Community version had several limitations and one of them is 100 licensed actions per day at the time of writing of this paper.

TASK #18: Suricata on Ubuntu

(Complete Task #1 and install VirtualBox)

(Complete Task #2 and install Ubuntu)

Ubuntu machine should have two interfaces, the first one at `NAT` network and the second at Internal network (`intnet`). Configure them first then boot the system.

`/etc/network/interfaces` should look like

```
# The loopback network interface

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet dhcp

auto eth1

iface eth1 inet dhcp
```

(Ref: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Ubuntu_Installation)

```
sudo apt-get -y install libpcre3 libpcre3-dbg libpcre3-dev build-essential autoconf automake libtool libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libcap-ng-dev libcap-ng0 make libmagic-dev libjansson-dev libjansson4 pkg-config

sudo apt-get -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0
```

Murat Cakir, murat.cakir@clf.com.tr

```
VER=3.2.2

wget "http://www.openinfosecfoundation.org/download/suricata-
$VER.tar.gz"

tar -xvzf "suricata-$VER.tar.gz"

cd "suricata-$VER"

./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --
localstatedir=/var

make

sudo make install-full

sudo ldconfig
```

(check your interface name)

```
ifconfig

sudo suricata -c /etc/suricata/suricata.yaml -i eth0 --init-
errors-fatal
```

(if runs successfully)

```
sudo suricata -D -c /etc/suricata/suricata.yaml -q 0
```

(ip forwarding)

```
vi /etc/sysctl.conf and uncomment

net.ipv4.ip_forward = 1
```

(add iptables and suricata startup rules to rc.local)

```
vi /etc/rc.local

# By default this script does nothing.

iptables -A INPUT -j NFQUEUE

iptables -A OUTPUT -j NFQUEUE

iptables -A FORWARD -j NFQUEUE

iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0.0.0.0/0 -j
MASQUERADE
```

Murat Cakir, murat.cakir@clf.com.tr

```
suricata -D -c /etc/suricata/suricata.yaml -q 0

exit 0
```

Reboot and your IPS should be ready.

As a client at internal network, create a new VM with 1 GB RAM, no disk, and use Kali image (http://cdimage.kali.org/kali-2017.1/kali-linux-2017.1-amd64.iso) as a live CD image. It will have a single network interface connected to internal network. This will act our machine at internal network

Boot the machine,

Add the line below (Google's DNS server) to `/etc/resolv.conf`

```
nameserver 8.8.8.8
```

and add default route

```
route add default gw [internal IP address of suricata box]
```

and Kali should be able to access internet.

Test Suricata:

At a console at suricata, tail the log file:

```
tail -f /var/log/suricata/fast.log
```

At Kali machine issue a request with the agent name 'BlackSun':

```
curl -A "BlackSun" www.google.com
```

and you should be able to see the alert, like below:

```
06/14/2017-07:41:05.251122  [**] [1:2008983:6] ET USER_AGENTS
Suspicious User Agent (BlackSun) [**] [Classification: A Network
Trojan was detected] [Priority: 1] {TCP} 10.0.2.15:44100 ->
172.217.17.196:80
```

Likewise, you can visit http://www.testmyids.com/ (which is not harmful) to trigger alerts at Suricata logs.

Reputation:

(Ref: http://suricata.readthedocs.io/en/latest/reputation/index.html)

Murat Cakir, murat.cakir@clf.com.tr

Edit `/etc/suricatat/suricata.yaml` and uncomment reputation related lines

```
# IP Reputation
reputation-categories-file: /etc/suricata/iprep/categories.txt
default-reputation-path: /etc/suricata/iprep
reputation-files:
 - reputation.list
```

And add:

```
 - reputation.rules
```

to the end of existing enabled rules (ending with .rules)

Then create your reputation list;

```
cd /etc/suricata/
mkdir iprep
cd iprep
echo 1,KnownBads,Hosts we do not want to communicate >
categories.txt
nslookup www.exploit-db.com
echo 192.124.249.8,1,80 > reputation.list
echo 192.124.249.9,1,80 >> reputation.list
```

Create the rule to trigger the alert;

```
cat<<EOF > /etc/suricata/rules/reputation.rules
alert ip any any -> any any (msg:"Communication with bad hosts";
flow:to_server; iprep:dst,KnownBads,>,50; sid:3000001; rev:1;)
EOF
```

Let suricata re-read the configuration;

```
kill -USR2 `cat /var/run/suricata.pid`
```

While monitoring the log file;

Murat Cakir, murat.cakir@clf.com.tr

```
tail -f /var/log/suricata/fast.log
```

Visit http://www.exploit-db.com at Kali;

And the rule should be triggered and alerts should be written to logs.

Change the rule at `/etc/suricata/rules/reputation.rules` and replace '`alert`' to '`drop`'

```
kill -USR2 `cat /var/run/suricata.pid`
```

Re-visit http://www.exploit-db.com at Kali and it should be unreachable.

TASK #19: SELKS, Phantom, OTRS integration

Phantom Part

(Complete Appendix C/Task #17 and Install Phantom)

Phantom Elasticsearch Integration

(Complete Appendix C/Task #15 Evaluate and Install SELKS)

Go to Phantom web interface, select Apps from left hand dropdown, go to Unconfigured Apps tab, find Elasticsearch

Select `Configure new asset`

Fill in Asset Name, fill in necessary fields and at next screen define groups (stats, users RW), submit

At Asset settings, set the URL;

URL `http://[ip of the SELKS box]`

`Save` and `Test Connectivity`

Integration part of Phantom with SELKS is completed

Install CENTOS for OTRS

Create a vm with a network interface (bridged), 4 GB RAM, 40 GB disk space

Install CENTOS from the DVD image (https://www.centos.org/download/)

Select the software as `Server with GUI`

Enable the network interface

Set the password for root and add a user (such as centos) and set a password for that user

Select disk (for automatic partitioning)

When installation is completed, remove the CD (image), reboot, at first start, accept the license agreement

Installation of OTRS on CENTOS

(Ref: http://doc.otrs.com/doc/manual/admin/stable/en/html/installation.html#installation-on-centos)

```
Disable SELINUX and reboot

yum install mariadb-server epel-release dkms kernel-devel kernel-headers

sudo su

cat<<EOF >>  /etc/my.cnf.d/zotrs.cnf

[mysqld]

max_allowed_packet   = 20M

query_cache_size     = 32M

innodb_log_file_size = 256M

EOF

systemctl start mariadb

/usr/bin/mysql_secure_installation

cd /tmp/

wget http://ftp.otrs.org/pub/otrs/RPMS/rhel/6/otrs-5.0.20-01.noarch.rpm

yum install --nogpgcheck otrs-5.0.20-01.noarch.rpm

service httpd restart

yum -y install "perl(Text::CSV_XS)"
```

Murat Cakir, murat.cakir@clf.com.tr

Visit `http://localhost/otrs/installer.pl` via your browser

Follow the steps at http://doc.otrs.com/doc/manual/admin/stable/en/html/web-installer.html

Note down generated password for '`otrs`' and generated password for '`root`'

```
su - otrs
/opt/otrs/bin/otrs.Daemon.pl start
/opt/otrs/bin/Cron.sh start
```

go to start page `http://localhost/otrs/index.pl`

OTRS installation is completed

Enabling OTRS Rest Interface

From Top Menu, select Admin

From Agent Management Group, select Agents

Select Add agent

From Top Menu, select Admin

From System Administration Group, select Web Services

From Action, Add web service

Give it a name (e.g. `otrsrest`)

OTRS will serve as a Provider

Select Network Transport : `HTTP::REST`

From Add Operation Dropdown, select each operation one by one and give them names, network transport settings will be done at the end (after defining all) by pressing `Configure` button next to Network Transport box

Operation	Name	Incoming Mapping	Outgoing Mapping	Network Transport
`Ticket::TicketGet`	`TicketGet`	`Simple`	`XSLT`	`/TicketGet`

Murat Cakir, murat.cakir@clf.com.tr

Sample test (for ticketID=1, username=agent, password=agentpw, rest service name=otrsrest:)

```
curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/otrsrest/TicketGet?TicketID=1&UserLogin=agent&Password=agentpw"
```

Operation	Name	Incoming Mapping	Outgoing Mapping	Network Transport
Ticket::TicketSearch	TicketSearch	Simple	XSLT	/TicketSearch

Sample test (for username=agent, password=agentpw, rest service name=otrsrest:)

```
curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/otrsrest/TicketSearch?UserLogin=agent&Password=agentpw"
```

Operation	Name	Incoming Mapping	Outgoing Mapping	Network Transport
Ticket::TicketCreate	TicketCreate	Simple	XSLT	/TicketCreate
Ticket::TicketUpdate	TicketUpdate	Simple	XSLT	/TicketUpdate

Save and finish

Phantom OTRS Integration

Go to Phantom web interface, select Apps from left hand dropdown, go to Unconfigured Apps tab

(If OTRS is not there, download it from https://my.phantom.us/2.1/apps/ and install via `Install App`)

Select `Configure new asset`

Fill in Asset Name, fill in necessary fields and at next screen define groups (stats, users RW), submit

Murat Cakir, murat.cakir@clf.com.tr

At Asset settings:

URL `http://[IP address of the OTRS box]`

username as you have entered for agent at OTRS side

password is the password of the user: '`agent`'

OTRS Web Service name as you entered at OTRS side (e.g. otrsrest)

you can leave rest as they are since we created accordingly.

`Save` and `Test Connectivity`

From `Customers > Customer Management` add a customer (i.e. customer)

`Customers > Customer User Management`, add a user to that customer (i.e. customer)

Integration part of Phantom with OTRS is completed

Adding the Playbook to Phantom

From Phantom `Home>Sources>Events` create an Event and note down the Id

Create a playbook via `Home>Playbooks> +Playbook`

Give a name to your playbook, select all for "`Operates on`" part

Click on Python Playbook Editor, copy/paste the code (provided at the section below)

Confirm the "`Edit Full Playbook?`" prompt and press `Save`

Enter a comment and `save` again

Now, you can click on the debugger and run the playbook, as the Container ID, enter the `Id number` you have noted down at "`Create an Event`" step and click on "`TEST`".

Subjects of OTRS tickets contain event IDs and they can be queried via visiting `http://[SELKS IP]/evebox/#/event/[Event ID]`

Phantom Playbook

```
"""
Sample playbook code for SELKS <> OTRS Integration by M.Cakir
(2017)
```

Murat Cakir, murat.cakir@clf.com.tr

```
"""
import phantom.rules as phantom
import json
from datetime import datetime, timedelta

def on_start(container):
      phantom.debug('on_start() called')
      get_config_1(container=container)
      return

def get_config_1(action=None, success=None, container=None,
results=None, handle=None, filtered_artifacts=None,
filtered_results=None):
      phantom.debug('get_config_1() called')
      parameters = []
      phantom.act("get config", parameters=parameters,
assets=['selks'], callback=run_query_1, name="get_config_1")
      return

def run_query_1(action=None, success=None, container=None,
results=None, handle=None, filtered_artifacts=None,
filtered_results=None):
      phantom.debug('run_query_1() called')
      parameters = []
      currentlog = datetime.now().strftime("%Y.%m.%d")
      parameters.append({
          'index': "logstash-alert-%s" % currentlog,
          'type': "SELKS",
          'routing': "",
          'query': "{ \"query\": { \"match_all\": {} } }",
      })
      phantom.act("run query", parameters=parameters,
assets=['selks'], callback=create_ticket_1, name="run_query_1",
parent_action=action)
      return

def create_ticket_1(action=None, success=None, container=None,
results=None, handle=None, filtered_artifacts=None,
filtered_results=None):
      phantom.debug('create_ticket_1() called')
      results_data_1 = phantom.collect2(container=container,
datapath=['run_query_1:action_result.data.*.hits.hits.*._id',
'run_query_1:action_result.data.*.hits.hits.*._source.src_ip',
'run_query_1:action_result.data.*.hits.hits.*._source.dest_ip',
'run_query_1:action_result.data.*.hits.hits.*._source.alert.signa
ture_id',
'run_query_1:action_result.parameter.context.artifact_id'],
action_results=results)
      formatted_data_1 = phantom.get_format_data(name='format_9')
      parameters = []
      for results_item_1 in results_data_1:
          if results_item_1[0]:
              parameters.append({
                  'Ticket:Title': "Action Required",
```

```
                'Ticket:CustomerUser': "customer",
                'Ticket:Priority': "3 normal",
                'Ticket:State': "new",
                'Ticket:Queue': "Raw",
                'Article:Subject': results_item_1[0],
                'Article:Body': "There has been an alert
regarding source %s and destination %s with an alert ID: %s" %
(results_item_1[1], results_item_1[2], results_item_1[3]),
                'context': {'artifact_id': results_item_1[4]},
            })
     phantom.act("create ticket", parameters=parameters,
assets=['otrs'], name="create_ticket_1", parent_action=action)
     return

def on_finish(container, summary):
     phantom.debug('on_finish() called')
     return
```

Murat Cakir, murat.cakir@clf.com.tr

Appendix D
Case Study

For the case presented at Section 4 from Forum of Incident Response and Security Teams' 2015 conference ("FIRST.org / 27th Annual FIRST Conference / Program", 2015). References (Hjelmvik, 2015) and (Davis, 2016) can be used for detailed information and for some of the answers.

Challenge questions were like following:

(1) Q1.1: What IP address did the attackers use?

(2) Q1.2: How did the attacker get the fr.jpg file to the webserver?

(3) Q1.3: Show how the web page looked after the defacement for URL

(4) Q1.4: List all commands FrogSquad sent using the cm0 backdoor on March 12

(5) Q1.5: Did FrogSquad come back at a later time from the same class C IP network (217.195.49.0/24)?

(6) Q 2.1: From which three "odd" (nonlegitimate) domain names the largest downloads made?

(7) Q2.2: Are the files downloaded from www.mybusinessdoc.com (68.164.182.11) malicious?

(8) Q2.4: Did the download from 1.webcounter.info (148.251.80.172) use HTTP,SSL or something else?

(9) Q2.5: How was this piece of malware delivered to Ned's computer? (HTTP / E-mail / Chat / Other)

(10) Q2.6: What domains does the JavaScript download additional malware from?

(11) Q2.7: What binaries were dropped by Delivery_Notification_00000529832.doc.js on April 7? MD5 sums wanted!

(12) Q3.1: From what IP and TCP port was the C2 software downloaded?

(13) Q3.2: What type of C2 channel was established from Krusty's computer to a server in Hong Kong after the C2 software was downloaded and executed?

Murat Cakir, murat.cakir@clf.com.tr

(14) Q3.3: Krusty's computer (.54) has been infected with some “badware”, when did this happen and how?

(15) Q3.4: Extract all emails sent with SMTP (NetworkMiner)

(16) Q3.5: List all long running sessions (Argus)

(17) Q3.6: Look for data exfiltration, i.e. large amounts of outbound data transfers (Argus)

We looked for the answers of the following questions:

a. Was real time prevention possible? (True/False/n.a.)
b. Could we find the answer of this question via an automate investigation? (True/False/n.a.)
c. Considering efforts and costs needed for setting up automation vs. human incident analysis which one would be cost and time beneficial? (A/H)

We didn’t provide the answers and techniques used, but the tools presented in this paper are sufficient for finding the answers. As a brief list, those are demonstrated at:

Task 7 (SiLK), Task 9 (MDL), Task 13 (Bro&CIF), Task 14 (Staxx), Task 15 (SELKS), Task 16 (Security Onion), Task 18 (Suricata)

Evaluation of the case for our questions:

(1) What IP address did the attackers use?
    a. False (we couldn’t block the attacker) It could have been possible to permit communication with whitelisted IPs only, and it might not be feasible
    b. True. Searching backwards, getting the alert from an integrity checker, looking for the uploaded file(s), searching for uploader’s IP
    c. Human handler
(2) How did the attacker get the fr.jpg file to the webserver?
    a. True. The vulnerability CVE-2014-1683 had a snort rule and an IPS could have prevented the attack

Murat Cakir, murat.cakir@clf.com.tr

b. True. (similar as above)

c. Human handler

(3) Show how the web page looked after the defacement for URL

a. True. Web site updates can be tracked

b. True

c. Automated

(4) List all commands FrogSquad sent using the cm0 backdoor on March 12

a. True. (backdoor implanting could have been avoided as it was possible to identify it since it was installing itself for autorun at Windows startup)

b. True. (though it was harder, it was still possible with tools like Viper)

c. Human

(5) Did FrogSquad come back at a later time from the same class C IP network (217.195.49.0/24)?

a. False. (could we prevent communication from the same C-Class?)

b. True (argus, silk etc. would suffice)

c. Automated

(6) From which three "odd" (nonlegitimate) domain names the largest downloads made?

a. Not applicable

b. True

c. Automated

(7) Are the files downloaded from www.mybusinessdoc.com (68.164.182.11) malicious?

a. True (can be checked and downloading can be prevented)

b. True

Murat Cakir, murat.cakir@clf.com.tr

c. Automated

(8) Did the download from 1.webcounter.info (148.251.80.172) use HTTP,SSL or something else?

a. True

b. True (Bro)

c. Automated (for small captures Human handler)

(9) How was this piece of malware delivered to Ned's computer? (HTTP / E-mail / Chat / Other)

a. True (it was possible to block the infection as the malware hash (1f5a31b289fd222e2d47673925f3eac9) could have been detected (https://virustotal.com/tr/file/e2d8b48dbd699edd3ccac2211567699cecf2f99b0d288e74458c8e303da4853d/analysis/)

b. True

c. Automated

(10) What domains does the JavaScript download additional malware from?

a. False. (could we block connection to those domains?) At least one of the domains (209.59.156.160 carina-paris-hotel.com) was not blacklisted, it was possible for communications with whitelisted only, but was not very likely

b. True

c. Human

(11) What binaries were dropped by Delivery_Notification_00000529832.doc.js on April 7? MD5 sums wanted!

a. True

b. True

c. Automated

Murat Cakir, murat.cakir@clf.com.tr

(12) From what IP and TCP port was the C2 software downloaded?

a. True (it could have been avoided)

b. True (could have been detected via automation)

c. Automated

(13) What type of C2 channel was established from Krusty's computer to a server in Hong Kong after the C2 software was downloaded and executed?

a. True. (103.10.197.187 was Blacklisted and we could deny communication on non-standard ports. Metasploit shells could have been identified although it could have been encoded differently)

b. True

c. Automated

(14) Krusty's computer (.54) has been infected with some "badware", when did this happen and how?

a. True (a Cryptolocker (https://virustotal.com/tr/file/45329e9da1af744b5d57cbb92d78cf884b1ae7c4390f2d5372dee462fa6c97f8/analysis/) type of malware could have been blocked via tools at several layers of defense)

b. True

c. Human

(15) Extract all emails sent with SMTP (NetworkMiner)

a. Not applicable

b. True

c. Automated

(16) List all long running sessions (Argus)

a. Not applicable

b. True

c. Automated

(17) Look for data exfiltration, i.e. large amounts of outbound data transfers (Argus)

a. True (large amount of data transfers can be blocked)

b. True

c. Automated

Table of Results:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Was real time prevention possible? (True/False/n.a.)	False	True	True	True	False	n.a.	True	True	True	False	True	True	True	True	n.a.	n.a.	True
Could we find the answer of this question via an automate investigation? (True/False/n.a.)	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
Regarding efforts and costs we used for automation vs. human incident analysis which one would be cost and time beneficial? (A/H)	H	H	A	H	A	A	A	A	A	H	A	A	A	H	A	A	A

Summary

For a specific case like presented, findings can be summarized as;

- Although it might be hard for specific cases, almost all the investigation work could have been carried out via automation
- For 11 of 14 situations (78,5%) prevention was possible
- Considering the recurrence likelihood of similar situations are high; investing time and effort into automation of investigation work could be beneficial for 12 of 17 (70,5%) of presented situations

Murat Cakir, murat.cakir@clf.com.tr

Appendix E
Mentioned Software and Tools

Phantom (Phantom Cyber)	https://www.phantom.us/
Kali (Offensive Security)	https://www.kali.org/
Security Onion (Security Onion Solutions, LLC)	https://securityonion.net/
SANS Investigative Forensic Toolkit (SIFT) Workstation (SANS)	https://digital-forensics.sans.org/community/downloads
SELKS (Stamus Networks)	https://www.stamus-networks.com/open-source/#selks
HoneyDrive	http://bruteforcelab.com/honeydrive
Lenny Zeltser's REMnux	https://remnux.org/
Phantom Community (Phantom Cyber)	https://www.phantom.us/community/
Network Monitor Freemium (LogRhythm)	https://logrhythm.com/network-monitor-freemium/
Response Operation Collection Kit (ROCK NSM) (MOCYBER)	http://rocknsm.io/
OSSIM (AlienVault)	https://www.alienvault.com/products/ossim
SOF-ELK (Phil Hagen/SANS FOR572)	https://github.com/philhagen/sof-elk
CapAnalysis (Gianluca Costa)	http://www.capanalysis.net/ca/
Flowbat (Applied Network Defense)	http://www.flowbat.com/
The Hive (Thomas Franco, Saâd Kadhi, Jérôme Leonard)	https://thehive-project.org/
MazeRunner Community Edition (Cymmetria)	https://community.cymmetria.com/

Murat Cakir, murat.cakir@clf.com.tr

FIR (Fast Incident Response)	https://github.com/certsocietegenerale/FIR/
Silk on a Box (NetSA)	https://tools.netsa.cert.org/confluence/pages/viewpage.action?pageId=23298051
OpenSOC (OpenSOC)	http://opensoc.github.io/
Moloch	http://molo.ch/
Wireshark	https://www.wireshark.org/
tcpdump	http://www.tcpdump.org/tcpdump_man.html
MISP	http://www.misp-project.org/
SiLK	https://tools.netsa.cert.org/silk/index.html
foremost	http://foremost.sourceforge.net/
grepcidr (Jem Berkes)	http://www.pc-tools.net/unix/grepcidr/
tcpxtract	http://tcpxtract.sourceforge.net/
tcpextract	https://github.com/faust/tcpextract
tshark_extractor	https://github.com/rangercha/tshark_extractor
Filelookup	https://github.com/hiddenillusion/FileLookup/blob/master/FileLookup.py
Bro Network Security Monitor	https://www.bro.org/index.html
DarkTrace	https://www.darktrace.com/technology/
Suricata	https://suricata-ids.org/
Viper binary management and analysis framework	http://viper.li/
Das Malwerk	http://dasmalwerk.eu/
Elastic X-Pack	https://www.elastic.co/products/x-pack

Murat Cakir, murat.cakir@clf.com.tr