Automated Detection of the Early Stages of Cyber Kill Chain

Ian Herwono and Fadi Ali El-Moussa
Security Futures Practice, Research & Innovation, BT, Ipswich IP5 3RE, U.K.

Keywords: Cyber Security, Kill Chain, Security Events Monitoring, Attack Detection Plan, Automated Process.

Abstract: Early detection of cyber threats is critical for proactive network defence and protection against data, financial and reputation loss that could be caused by large-scale security breach. Continuous monitoring and in-depth analysis of related system and network events are required to achieve the objective. However cyber threat hunting activities are both time-consuming and labour-intensive; the prospect of being able to automate them effectively is thus worth exploring. In this paper we introduce the prototype of our attack detection tool for automating the process of discovering and correlating security events towards early threat detection. Its main objective is to facilitate continuous event monitoring and to alert security analysts whenever a series of detected events and activities may indicate early stages of a cyber kill chain. The process automation will reduce the load of human analysts and spare them valuable time to investigate more sophisticated, unknown attacks. We provide two use cases which describe the chain of tasks a security analyst would have to perform when investigating cyber incidents and trying to identify the systems targeted by potential attack. We then show how to create attack detection plans for those use cases and apply them on relevant datasets. We present the results produced by the tool and discuss our future work on context-aware classification of security events which aims to make the detection process more efficient.

1 INTRODUCTION

Today's cyber attackers will try to find any method to subvert the computer network and system of companies and government agencies with the main intents of collecting confidential and private data, disrupting critical Internet services, or denying access to user data. With the modern complexity of such attacks the only way to be reliably alerted if a system has been compromised is by reviewing both the system and network events and correlating them across those two layers to develop a thorough view into a potential attack. The problem gets even worse as every system in the network has the ability to log events, and in a busy network the network traffic per day could be very large. Hence the volume of log data generated by the end systems and network devices will be so large that it is impractical for system administrators and security analysts to review every data record in the log and correlate those events at system and network level to detect a potential attack.

Security solutions such as the Security Information and Event Management (SIEM) tools try to make the analyst's life easier by collecting the events from various systems and network devices into one place, grouping them into categories, and providing the analyst with easy, centralised access interface to any of the alerts or logs. SIEM solution usually provides an interactive dashboard to help analysts drill down into the data and correlate series of system and network events in order to make sense if they indicate a potential security breach. The analysts should have cyber threat hunting skill and experience in order to be able to spot (malicious) activities and discover connections between seemingly unrelated events that may have been seen in the network over long period of time but could be part of a cyber kill chain with high-impact security breach. The analyst's task basically comprises actions of discovering the footprints left behind by the attacker on systems all over the network, tracing relevant network traffic, and trying to figure out the attacker's intent, i.e. to know where the attacker is heading over time and which service or data is being targeted. Being able to detect threats earlier in the kill chain is essential to defend the network proactively and prevent against data, financial and reputation loss caused by large-scale security breach. This means that continuous monitoring and in-depth

Herwono, I. and El-Moussa, F.
Automated Detection of the Early Stages of Cyber Kill Chain.
DOI: 10.5220/0006543301820189
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 182-189
ISBN: 978-989-758-282-0

analysis of relevant system and network events are required to achieve the objective.

Cyber threat hunting activities are both time-consuming and labour-intensive when carried out manually; the prospect of being able to automate them effectively is thus worth exploring. In this paper we introduce the prototype of our attack detection tool to automate the process of observing and correlating security events towards early threat detection. The objective is to allow continuous event monitoring and alert the analysts whenever a series of events that were spread over time may indicate early stages of a known attack pattern. Such automation will reduce the load of human analysts and spare them valuable time to investigate more sophisticated, unknown attacks.

The remainder of this paper is structured as follows. Section 2 presents some related works in the area of attack patterns and multi-stage attack detections. In Section 3 we introduce two example use cases that will later be applied to our attack detection tool. Section 4 briefly describes the tool. In Section 5 we show how to use the tool and specify the parameters for automating the detection and correlation process described in the use cases. Section 6 discusses our future work on context-aware classification of security events in order to improve the automated detection process. Finally we conclude our work in Section 7.

2 RELATED WORK

(Bhatt et al., 2014) presented the design principles of a framework that model multi-stage attacks in a way that both describes the attack methods as well as the anticipated effects of attacks. Their foundation to model behaviours is by combining the *Intrusion Kill Chain* attack model (Hutchins et al., 2011) and defence patterns, i.e. a hypothesis-based approach of known patterns. (Barnum, 2007) introduced the concept of attack patterns as a mechanism to capture and communicate the attacker's perspective. He described a typical process for generating the attack patterns and using them at different phases of software development. The work was related to the *Common Attack Pattern Enumeration and Classification (CAPEC)* initiative of the U.S. Department of Homeland Security (DHS) on providing a publicly available catalogue of attack patterns with a comprehensive schema and classification taxonomy created to assist in the building of secure software (https://capec.mitre.org). (Scarabeo et al., 2015) developed a data mining framework that employs text mining techniques to dynamically relate the information between the security-related events and CAPEC attack patterns. It aims to reduce analysis time and increase the quality of attack reports, as well as to automatically build correlated scenarios.

3 USE CASES

This section presents the two use cases for our attack detection tool (Herwono et al., 2017). They describe the typical time-consuming tasks human analysts need to perform when analysing large amount of logs to investigate specific type of attacks. Both use cases are based on cyber incident analysis using a dataset that contains one day of Snort IDS alert logs with over two million records (MACCDC, 2012). Table 1 lists the top 10 names of alert classification with the most numbers of detected events/alerts in the dataset along with the number of associated alert signatures.

Table 1: Snort alert classification names with the most detected events (Top 10).

Alert Classification	# events	# signatures
Web Application Attack	1,345,864	517
Attempted Information Leak	244,334	232
Misc activity	226,245	51
Potential Corporate Privacy Violation	141,403	27
Unsuccessful User Privilege Gain	32,431	5
Generic Protocol Command Decode	26,129	41
Access to a Potentially Vulnerable Web Application	25,898	340
Potentially Bad Traffic	24,312	43
Misc Attack	12,189	17
Attempted Administrator Privilege Gain	9,777	44

3.1 Use Case 1: Identify Compromised Web Server

In this use case an Internet company has a number of web services running on its network and its Security Operations Centre (SOC) team needs to be aware if one or more of their web servers has been targeted by an attacker or worse it has been compromised. Their Snort IDS system has produced a huge number of alert logs which need to be examined to differentiate between real threats and false positives. A typical scenario, using the SIEM's interactive dashboard a SOC analyst will first isolate all the alerts that were classified as *Web Application Attack* (i.e. 1,345,864 events). Then he/she will check how many destination hosts (i.e. destination IP addresses) were involved under that alert category (i.e. 30 hosts).

As the analyst has now found out that there is potential web attack traffic targeting some of the company's web services, the next step would be to narrow down the search and verify if any of the web servers has been successfully compromised. To do this the analyst searches for *Access to a Potentially Vulnerable Web Application* alerts among those 30 destination IP addresses. Eventually, by carefully examining the search result (i.e. 1,672 events in total) the analyst found that a *Cybercop Scan* activity (i.e. 2 events) was detected for a particular destination IP address; the analyst has now successfully identified the web server that has been potentially compromised.

3.2 Use Case 2: Determine Type of Attack

In the second use case the SOC analyst wants to determine the actual type of attack or security breach on web servers that he/she suspected may have been compromised. This is important in order to assess the damage that could have been caused by successful attack and to allow the administrators take suitable mitigation actions such as blocking inbound web traffic or shutting down the server completely.

The analyst will first look for all Snort IDS alerts that relate to *Detection of a Network Scan* classification (i.e. 694 events). This query results in a list of matching alerts with different attack signature names, one of which catches the analyst's attention, i.e. *WEB_SERVER IIS 8.3 Filename With Wildcard (Possible File/Dir Bruteforce)* with 22 events. The analyst then retrieves the list of destination IP addresses reported with that attack signature which indicates directory traversal activity (i.e. 16 IP addresses).

The analyst now needs to know if the attacker was able to successfully compromise any of the IIS (Internet Information Services) web servers deployed in the network. He/she starts looking for alerts classified as *Web Application Attack* and spots a set of alerts with signature name *WEB-IIS Directory Traversal Attempt* (i.e. 874 events). The analyst collects the involved 29 destination IP addresses and checks them against the list of destination IP addresses obtained earlier (i.e. for *WEB_SERVER IIS 8.3 Filename With Wildcard*) in order to determine the actual servers that have been targeted by the attacker.

This is a typical scenario, where the analyst has to compare alerts collected from one event possibly at network level with events at host level to verify that the network scanning activity detected as attacks towards web server's directory traversal matches the attack at host with directory traversal attempt. Following the comparison the analyst identifies 15 destination IP addresses that were involved in both attack signatures which may lead to root access to the web server using the *Directory Traversal* method. To narrow down the search further, the analyst may need to investigate the system logs to check who managed to log into the system and compare the login time with the collected Snort alerts and gather more evidence of successful attack.

4 ATTACK DETECTION TOOL

4.1 Introduction

Our attack detection tool allows cyber-defence analysts to automate the process of correlating specific security, system and network events that may have been logged at different places in the network over short or long period of time. In order to correlate two separate events correctly an analyst normally needs to verify whether the same machine or device is involved in both events. For example, if there are some IDS alerts involving a device with particular IP address, the analyst may then want to check whether there is any suspicious outbound traffic coming from that device in subsequent periods of time. Regular checks with DHCP (Dynamic Host Configuration Protocol) records are thus needed to make sure that the IP address still matches the same device. The tool helps automate this type of checks and correlates past events with subsequent series of new events to save analyst a lot of time in monitoring the progress of potential attacks.

4.2 Attack Detection Plan

Attack detection plans are used for triggering and automating the process of detecting various types of cyber-attacks. Each plan indicates a series of events that collectively may form logical steps and phases of cyber kill chain; the plan sets the conditions under which the events could be correlated with each other, leading to certain conclusions, e.g. preparation for implanting a backdoor within the target system. Essentially the plan represents the step-by-step process of investigating cyber incidents that otherwise a human analyst would have to go through in order to verify the attack and narrow down the search for the intent of the malicious activities.

5 AUTOMATED ATTACK DETECTION

In this section we will show how we can utilise our attack detection tool to automate the sequence of tasks an analyst would have to perform based on the use cases described earlier in Section 3. The benefits of such automation are summarised as follows:

- The analyst can redo the same type of analysis on either different set of data gathered from different network, or same type of data collected at different periods of time without having to keep repeating the same tasks every single time.
- The attack detection plans can be shared with other analysts to be used for analysing different set of data in their own network.
- Members of the analyst team can validate each other's finding in an automated way.
- Different security analysts can add or edit the automated detection steps to cover other type of attack or more complex attack pattern.

For each use case we will create its corresponding attack detection plan by specifying the set of events to be detected along with their correlation parameters. Then we will show the results produced by the tool using the Snort IDS dataset described in Section 3.

5.1 Use Case 1: Identify Compromised Web Server

5.1.1 Detection Plan

In this use case the analyst's task is to identify any company's web server that has been compromised. We can split this task into two sub-tasks, i.e. *Detect potential attack on web applications*, and *Verify scan activities on web server*. The sub-tasks are then connected to each other to form the attack detection plan, as shown in Figure 1.

The next step is to specify which events to be monitored or actions to be taken within each sub-task. Figure 2 shows the details for the first sub-task. Its objective is to gather alerts classified as *Web Application Attack* within one-hour time interval and then group them either by the source or destination IP address. It will trigger the next sub-task once any of such alerts has been detected from the logs, i.e. threshold is set to 1.

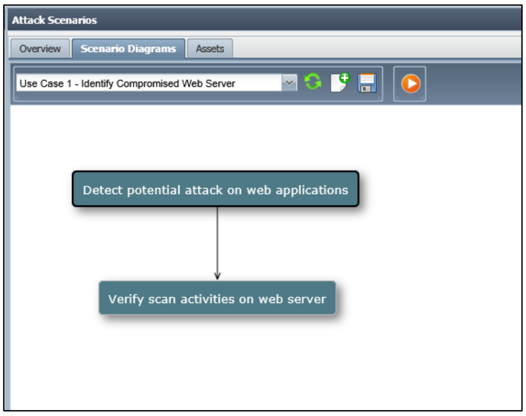
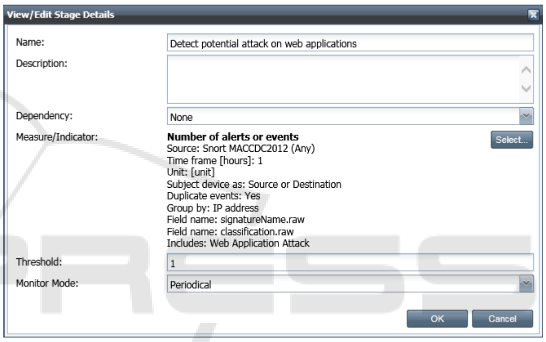

Attack Scenarios
Overview
Scenario Diagrams
Assets
Use Case 1 - Identify Compromised Web Server
Detect potential attack on web applications
Verify scan activities on web server

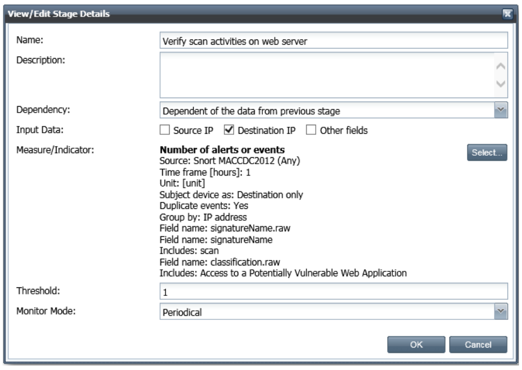
Figure 1: Attack detection plan for Use Case 1.



Name: Detect potential attack on web applications
Description:
Dependency: None
Measure/Indicator: **Number of alerts or events**
Source: Snort MACCDC2012 (Any)
Time frame [hours]: 1
Unit: [unit]
Subject device as: Source or Destination
Duplicate events: Yes
Group by: IP address
Field name: signatureName.raw
Field name: classification.raw
Includes: Web Application Attack
Threshold: 1
Monitor Mode: Periodical

Figure 2: Details of the first sub-task.

The second sub-task should verify if any of the destination IP addresses identified in the first sub-task has also been involved in other Snort alerts categorised as *Access to a Potentially Vulnerable Web Application* and if some malicious scan activities have been detected on those affected machines. Figure 3 shows how the second sub-task is configured where the dependency field is now set to *Dependent of the data from previous stage* and *Destination IP* is selected as input data. Also note that only alerts with signature name containing the word "scan" will be considered (see the corresponding *Includes* field).



Name: Verify scan activities on web server

Description:

Dependency: Dependent of the data from previous stage

Input Data: □ Source IP ☑ Destination IP □ Other fields

Measure/Indicator:
Number of alerts or events
Source: Snort MACCDC2012 (Any)
Time frame [hours]: 1
Unit: [unit]
Subject device as: Destination only
Duplicate events: Yes
Group by: IP address
Field name: signatureName.raw
Field name: signatureName
Includes: scan
Field name: classification.raw
Includes: Access to a Potentially Vulnerable Web Application

Threshold: 1

Monitor Mode: Periodical

Figure 3: Details of the second sub-task.

5.1.2 Result

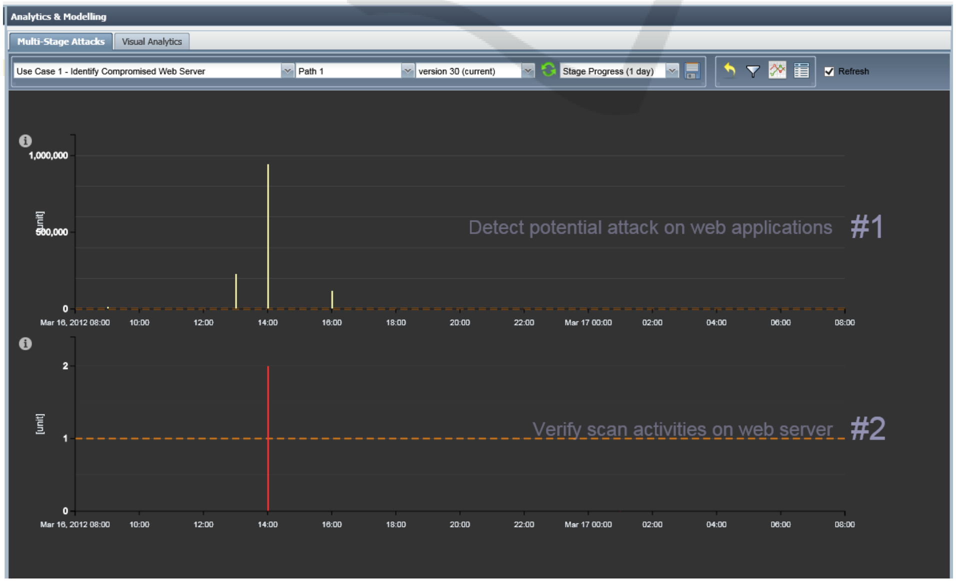
The tool provides a capability to ‘replay’ the detection plan over relevant datasets stored in a database-like system. This can either be used to verify the detection and correlation tasks on known datasets, or to discover evidence of attack activities in historical logs during forensic analysis. Our prototype implementation makes use of the *Elasticsearch* system for alert logs storage.

Before the replay can go ahead the user analyst may add or modify some details regarding the time interval and specific time period of the logs. The time interval should normally be aligned with the time frame configured for each sub-task, e.g. hourly, daily, etc. For our particular use case the time period is set between “16 March 2012 at 00:00 UTC” and “17 March 2012 at 00:00 UTC”.

Once the replay has started, the analyst can follow the progress of each sub-task in detecting and correlating the events within the time period via the tool’s visualisation interface. As depicted in Figure 4 the vertical lines (or bars) indicate the occurrence of events matching the filter parameters of the corresponding sub-task at particular time point and the line length represents the (maximum) examined value, e.g. maximum number of detected alerts per destination IP address. Each time the threshold in the first sub-task is exceeded, the second sub-task is triggered and provided with the list of identified destination IP addresses, i.e. the first time already happened on “16 March 2012 at 08:00 UTC”. While the second sub-task now starts its own detection activity, the first sub-task resumes. This process goes on until it reaches the end of the time period.

Figure 4 shows that the second sub-task has detected related events on “16 March 2012 at 14:00 UTC”. The analyst can then drill-down into the data and see the details of the matching events. As shown in Figure 5 a server with the IP address “192.168.229.101” has been targeted by attack with the signature *WEB-MISC cybercop scan*.

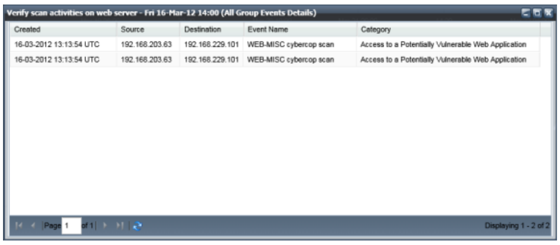
The whole detection process took about 10 seconds to complete. The processing time should however be seen as indicative only as there is a number of factors affecting the system performance such as type of operating system, server configuration, CPU utilisation, network load, software optimisation, etc. We deployed our attack detection tool using *Apache Tomcat* application



Use Case 1 - Identify Compromised Web Server | Path 1 | version 30 (current) | Stage Progress (1 day) | ☑ Refresh

Figure 4: Visualisation of detected events in Use Case 1.

server running on a Windows 2012 server with 32GB RAM. The *Elasticsearch* server application also runs on the same machine.



Verify scan activities on web server - Fri 16-Mar-12 14:00 (All Group Events Details)

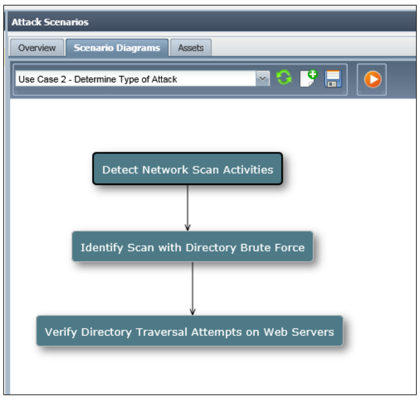
Created	Source	Destination	Event Name	Category
16-03-2012 13:13:54 UTC	192.168.203.63	192.168.229.101	WEB-MISC cybercop scan	Access to a Potentially Vulnerable Web Application
16-03-2012 13:13:54 UTC	192.168.203.63	192.168.229.101	WEB-MISC cybercop scan	Access to a Potentially Vulnerable Web Application

Figure 5: Detected events in the second sub-task.

5.2 Use Case 2: Determine Type of Attack

In the second use case the aim is to automate detection of directory traversal attack and to identify the servers that have been targeted. Figure 6 shows the corresponding detection plan which consists of three sub-tasks, i.e., *Detect network scan activities*, *Identify scan with directory brute force*, and *Verify directory traversal attempts on web servers*. Table 2 summarises the detection parameters for each sub-task.

After replaying the detection plan over historical logs as we did on the first use case, our tool produced the results shown in Figure 7. The directory traversal attempts were detected four times during the day (split in one-hour intervals) on “16 March 2012” between “08:00 UTC” and “14:00 UTC”. Figure 7 also shows that on “16 March 2012 at 09:00 UTC” five destination IP addresses were involved in the detected events. In total 15 servers were identified by the detection plan. The detection process took around 10 seconds to complete.



Attack Scenarios
Overview
Scenario Diagrams
Assets
Use Case 2 - Determine Type of Attack
Detect Network Scan Activities
Identify Scan with Directory Brute Force
Verify Directory Traversal Attempts on Web Servers

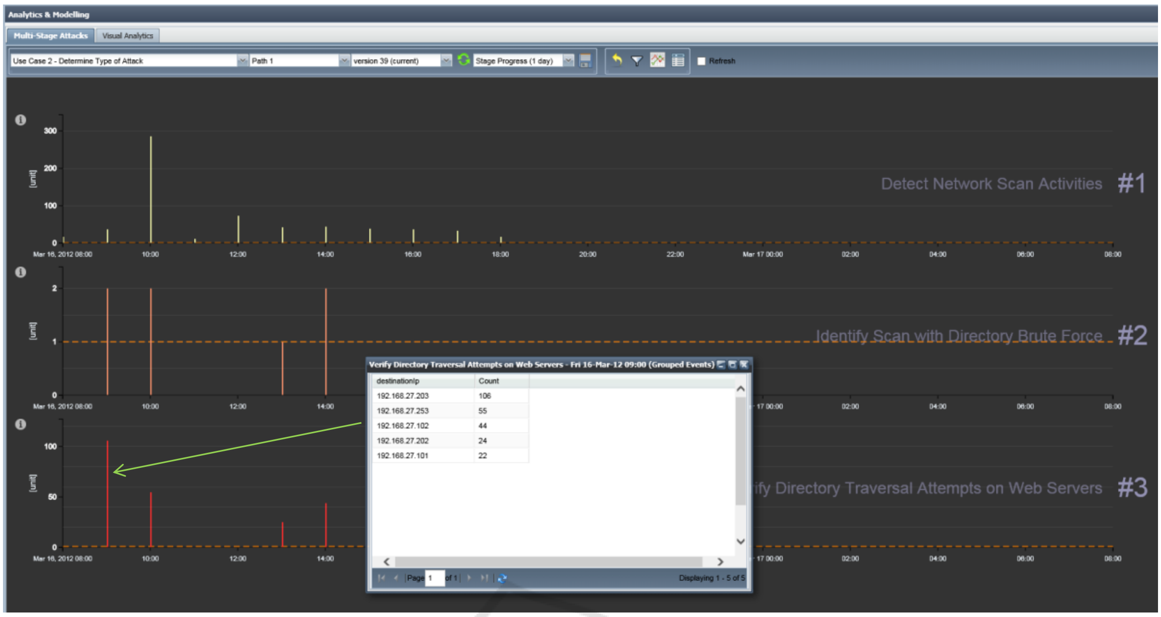
Figure 6: Attack detection plan for Use Case 2.

6 FUTURE WORKS

Basically the previous set of uses cases have been achieved on security events gathered from the same security system, i.e. Snort IDS. Large enterprises may have a number of different security systems deployed in their network supplied by multiple vendors such as McAfee, Trend, FortiGate, etc. Each of these systems will be reporting security events based on the vendor’s own naming and classification system. For example, as we see in Table 3 a number of reported events are showing different names and the SIEM’s security context database may group them into different categories. However many of them actually belong to the same malicious activity, e.g. port scanning.

Table 2: Sub-task detection parameters (Use Case 2).

Sub-task	**Detect Network Scan Activities**	**Identify Scan with Directory Brute Force**	**Verify Directory Traversal Attempts on Web Servers**
Input Data	N/A	Destination IP	Destination IP
Measure	**Number of alerts or events** Source: Snort MACCDC2012 (Any) Time frame [hours]: 1 Subject devices as: Source or Destination Duplicate events: Yes Group by: destinationIp Field name: classification.raw Includes: Detection of a Network Scan	**Number of alerts or events** Source: Snort MACCDC2012 (Any) Time frame [hours]: 1 Subject devices as: Source or Destination Duplicate events: Yes Group by: destinationIp Field name: classification.raw Includes: Detection of a Network Scan Field name: signatureName Includes: Bruteforce	**Number of alerts or events** Source: Snort MACCDC2012 (Any) Time frame [hours]: 1 Subject devices as: Destination only Duplicate events: Yes Group by: destinationIp Field name: classification.raw Includes: Web Application Attack Field name: signatureName Includes: WEB-IIS Directory transversal attempt
Threshold	1	1	1



Verify Directory Traversal Attempts on Web Servers - Fri 16-Mar-12 09:00 (Grouped Events)

destinationIp	Count
192.168.27.203	106
192.168.27.253	55
192.168.27.102	44
192.168.27.202	24
192.168.27.101	22

Page 1 of 1

Displaying 1 - 5 of 5

Figure 7: Visualisation of detected events in Use Case 2.

Table 3: Vendor-specific event classification.

Events	Classification	Vendor
TCP: SYN Port Scan	Denial of Service (DOS)	McAfee
UDP: Port Scan	Scan	McAfee
icmp_flood	ICMP Flood Attack	FortiGate
TCP.Invalid.Packet.Size	Spoofing	FortiGate
tcp_port_scan	Scan	FortiGate

It is therefore important that our detection tool can be vendor-agnostic so that the same attack detection plans can be used irrespective of which security system was used to generate the security events. Such flexibility would further boost the advantage of having such automation tool since a security analyst does not need to be familiar with different vendor's naming and classification systems in order to identify particular type of attack.

In order to achieve this objective we currently work on context-aware activity classification component for grouping security events from different vendors based on their semantics. This is due to the fact that vendor's classification system is rather static and usually does not take the surrounding context of the event into account, such as the source/destination IP address or the type of infrastructure being targeted. Depending on the context, a security event with specific signature may indicate different types of activity. On the other hand two or more security events with different signatures may indicate the same type of activity.

The context-aware activity classification component may later be integrated with our attack detection tool. We envisage the following input data:

- Security event or alert data
- Security context data which contains information about the classification assigned by the system vendor to particular event/alert (e.g. see Table 3)
- Domain knowledge data which contains some pre-defined knowledge and metrics to be used for the classification process

The context-aware classification process will be executed as follows:

1. First we build baseline models for the normal protocol (TCP, UDP, ICMP, etc.), services (Web server, database, etc.) and infrastructure (DNS, DHCP, etc.) behaviours. The models become part of *domain knowledge* data and will be used later to compare the reported security events with the baseline and determine how far away the security events are from the baseline and assign a weight to the distance.
2. We then extract the relevant information from the security events. As shown in Figure 8 the *Events Attributes Extraction* sub-component will first extract the following event attributes:
    - Event or alert name (this is normally the signature name given by the vendor)
    - Date and time when the event was detected

   - Source and destination IP addresses and port numbers
   - Application and network protocols (e.g. HTTP, TCP, UDP, etc.)
   - Infrastructure or device type of the source or destination (e.g. web server, database server, authentication server, etc.)

3. We then combine the extracted attributes and process them together with security context data, *domain knowledge* and metrics (i.e. baseline behaviours) in order to determine the context of the event, and eventually to decide whether an event/alert is classified as false positive or into one of pre-defined (malicious) activities, e.g.:
   - Port scanning
   - Denial of service
   - Code injection
   - Gaining access
   - Probing
   - Privilege escalation
   - Code execution
   - Data exfiltration

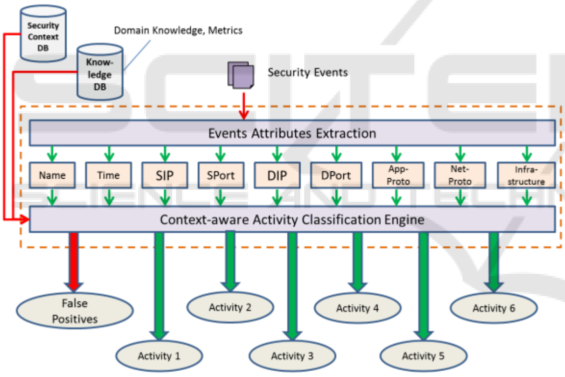


Figure 8: Context-aware activity classification.

Table 3 already showed an example of context-aware activity classification outcome where all the listed events may be classified as *Port Scanning* activity.

7 CONCLUSIONS

We have explored the possibility of automating the process of monitoring and correlating security events to help analysts detect potential attacks earlier in the kill chain. Our approach was to run pre-defined attack detection plans over data logs that have been produced by various systems and devices in the network over different periods of time. This way an analyst does not have to keep repeating the same steps and tasks every time he/she wants to perform similar analysis on different datasets. Nevertheless an in-depth knowledge of the format and security context of each data log is critical to create effective detection plans in order to achieve the full benefits of such automation. We thus believe that our future work on context-aware activity classification can well improve the efficiency and effectiveness of the proposed detection approach.

ACKNOWLEDGMENT

This work was partially supported by the H2020 EU funded project C3ISP [GA #700294]. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of their employers, the C3ISP project, or the Commission of the European Union.

REFERENCES

Bhatt, P., Yano, E. T., Gustavsson, P. M. 2014. Towards a Framework to Detect Multi-Stage Advanced Persistent Threats Attacks. In *Proceedings of the IEEE 8th International Symposium on Service Oriented System Engineering* (Oxford, UK, Apr 2014). SOSE 2014.

Hutchins, E., Cloppert, M., Amin, R. 2011. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. In *Proceedings of the 6th International Conference on Information Warfare and Security* (Washington, DC, Mar 2011).

Barnum, S. 2007. An Introduction to Attack Patterns as a Software Assurance Knowledge Resource. In *OMG Software Assurance Workshop* (Fairfax, VA, Mar 2007).

Scarabeo, N., Fung, B. C. M., Khokhar, R. H. 2015. Mining known attack patterns from security-related events. *PeerJ Computer Science* 1:e25.

Herwono, I., El-Moussa, F. 2017. A Collaborative Tool for Modelling Multi-Stage Attacks. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy* (Porto, Portugal, February 2017).

MACCDC. 2012. Capture files from Mid-Atlantic CCDC (Collegiate Cyber Defense Competition). URL: https://www.netresec.com/?page=MACCDC.