

MASTER'S THESIS

---

# Network Security

---

*Candidate:*  
Jason BISSICT

*Supervisor:*  
Professor Andrew  
HUTCHISON

Augmenting Security Event Information with  
Contextual Data to Improve the Detection Capabilities  
of a SIEM

Department of Computer Science  
University of Cape Town

2016

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

## **Abstract**

The increasing number of cyber security breaches have revealed a need for proper cyber security measures. The emergence of the internet and the increase in overall connectivity means that data is more easily accessible and available. Using the available data in a security context may provide a system with an external contextual insight such as environmental awareness or current affair awareness. A security information and event management (SIEM) system is a security system that correlates security event information from surrounding systems and decides whether the surrounding environment (possibly an enterprise's network) is vulnerable or even under attack by a malicious person whether they be internal (authorised) or external (unauthorised). In this thesis, the aim is to provide such a system with context by adding non-security related information from surrounding available sources known as context information feeds. Contextual information feeds are added to the SIEM and tested using randomised events. There are various context information types used in this thesis, namely: social media, meteorological, calendar information and terror threat level. The SIEM is tested with each contextual data feed active and the results are recorded. The testing shows that the addition of contextual data feeds actively affects the sensitivity of SIEM and hence results in higher alarms raised during elevated context triggered states. The system showed a greater and deeper visibility of its surrounding environment and hence an improved detection capability.

### **Acknowledgements**

Although this section is not required, I feel it is still necessary. I would like to thank my parents for their support and unwavering faith throughout this endeavour. And I would like to thank Prof. Hutchison for sticking with me this whole time, always ready with good advice and stern encouragement.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Hypothesis . . . . .	7
<b>2</b>	<b>Background Research and Prerequisite Concepts</b>	<b>8</b>
2.1	Introduction . . . . .	8
2.2	The Importance of Cyber Security . . . . .	8
2.3	Overview of a SIEM . . . . .	11
2.4	AlienVault and its Open Source SIEM . . . . .	14
2.4.1	Introduction . . . . .	14
2.4.2	Summary Evaluation of SIEM Products . . . . .	16
2.5	Security Event Information . . . . .	20
2.5.1	Authentication and Authorization Logs . . . . .	21
2.5.2	Change Logs . . . . .	23
2.5.3	Network Activity Logs . . . . .	24
2.5.4	Resource Access Logs . . . . .	25
2.5.5	Syslog . . . . .	27
2.5.6	Examples of Real Logs . . . . .	27
2.6	Contextual Data . . . . .	29
2.7	Threat Intelligence . . . . .	30
2.7.1	Threat Intelligence Standards . . . . .	32
2.7.2	Conclusion . . . . .	33
2.8	Optimising a SIEM . . . . .	33
2.8.1	Optimising through Inspection . . . . .	34
2.8.2	System and Platform Integration . . . . .	35
2.8.3	Attack Modelling . . . . .	36
<b>3</b>	<b>Security Information and Event Management System</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Log Collection . . . . .	38
3.3	Parsing and Normalisation of Logs . . . . .	39
3.4	Correlation . . . . .	40
3.5	Event Log Retention . . . . .	41
3.5.1	Historical Analysis vs Real-time Analysis . . . . .	41
3.6	Alerting . . . . .	42

<b>4</b>	<b>The Nature of Contextual Data</b>	<b>44</b>
4.1	Introduction . . . . .	44
4.2	Challenges of Context . . . . .	45
4.3	Building Contextual Data Applications . . . . .	46
4.3.1	Context Models . . . . .	46
4.3.2	Context Type Categorisation . . . . .	47
4.3.3	Categorisation of Context Aware Systems . . . . .	48
4.4	Summary . . . . .	49
<b>5</b>	<b>Design of Testing Environment and Data Streams</b>	<b>50</b>
5.1	General Design . . . . .	50
5.1.1	The System: OSSIM . . . . .	51
5.1.2	Rsyslog . . . . .	52
5.1.3	The Network . . . . .	52
5.1.4	Application Development Environment . . . . .	52
5.1.5	Data Formats . . . . .	53
5.1.6	General Design Summary . . . . .	53
5.2	Context Data Retrieval Design . . . . .	54
5.2.1	Context: Social Media Information . . . . .	55
5.2.2	Context: Meteorological Information . . . . .	59
5.2.3	Context: Calendar Events . . . . .	64
5.2.4	Context: Terror Threat Level . . . . .	67
5.3	Log File Design . . . . .	68
5.3.1	Social Media Context . . . . .	69
5.3.2	Meteorological Context . . . . .	69
5.3.3	Calendar Events Context . . . . .	70
5.3.4	Terror Threat Context . . . . .	70
5.4	Rule Set Design . . . . .	71
5.4.1	Social Media Context . . . . .	75
5.4.2	Meteorological Context . . . . .	75
5.4.3	Calendar Events Context . . . . .	76
5.4.4	Terror Threat Context . . . . .	76
5.5	Summary . . . . .	76
<b>6</b>	<b>Implementation of Test Environment and Contextual Data Feeds</b>	<b>78</b>
6.1	Our Testing Environment Setup . . . . .	78
6.1.1	OSSIM System . . . . .	80
6.1.2	Linux Event System . . . . .	81
6.1.3	Rsyslog Setup . . . . .	82
6.1.4	Control Test Directive Ruleset Implementation . . . . .	85
6.2	Contextual Data Feed Programs . . . . .	86
6.2.1	Social Media Context . . . . .	88
6.2.2	Meteorological Context . . . . .	91
6.2.3	Calendar Event Context . . . . .	93
6.2.4	Terror Level Context . . . . .	95

6.3	Summary . . . . .	98
<b>7</b>	<b>Testing and Results</b>	<b>99</b>
7.1	Control Test Implementation Testing and Results . . . . .	99
7.2	Social Media Implementation Testing and Results . . . . .	100
7.2.1	Social Media Implementation Testing . . . . .	100
7.2.2	Limitations of the Social Media Implementation Testing . . . . .	104
7.2.3	Social Media Implementation Results . . . . .	104
7.3	Meteorological Information Implementation Testing and Results . . . . .	105
7.3.1	Meteorological Information Implementation Testing . . . . .	105
7.3.2	Limitations of the Meteorological Information Implementation Testing	108
7.3.3	Meteorological Information Implementation Results . . . . .	108
7.4	Calendar Event Information Implementation Testing and Results . . . . .	109
7.4.1	Calendar Event Information Implementation Testing . . . . .	109
7.4.2	Limitations of the Calendar Event Information Implementation Testing	111
7.4.3	Calendar Event Information Results . . . . .	111
7.5	Terror Threat Level Information Implementation Testing and Results . . . . .	112
7.5.1	Terror Threat Level Information Implementation Testing . . . . .	112
7.5.2	Limitations of the Terror Threat Level Information Implementation Testing . . . . .	114
7.5.3	Terror Threat Level Information Results . . . . .	114
<b>8</b>	<b>Analysis of Results</b>	<b>117</b>
8.1	Common Proof of Concept Results in each Contextual Data Feed Implemen- tation . . . . .	117
8.2	Social Media Results Analysis . . . . .	118
8.3	Meteorological Results Analysis . . . . .	119
8.4	Calendar Events Results Analysis . . . . .	119
8.5	Terror Threat Level Results Analysis . . . . .	120
<b>9</b>	<b>Conclusion</b>	<b>122</b>
9.1	Summary Overview . . . . .	122
9.2	Discussion of Proof of Concept . . . . .	125
9.3	Conclusion of Results . . . . .	126
9.4	Limitations to Findings . . . . .	128

# List of Tables

2.1	Table showing different attack methods [12]	11
2.2	Table showing the Strengths and Weaknesses of AlienVault	17
2.3	Table showing the Strengths and Weaknesses of IBM's QRadar	18
2.4	Table showing the Strengths and Weaknesses of HP's ArcSight Solution Line	19
2.5	Table showing OSSIM vs USM as AlienVault Products	20
2.6	Table showing authentication and authorisation logs	22
2.7	Table showing the change logs	24
2.8	Table showing network activity logs	25
2.9	Table showing resource access logs	26
2.10	Table showing threat information vs threat intelligence	31
5.1	Table showing the application development environment	53
5.2	Checklist of procedures for each contextual data type	55
5.3	Table of Social Mention Information	56
5.4	Table of Documenting Social Mention's Rest request parameters	57
5.5	Table of Documenting Social Mention's Rest response	58
5.6	OpenWeatherMap Information	60
5.7	OpenWeatherMap's Rest Response Parameters	62
5.8	OpenWeatherMap Library methods	63
5.9	Table of methods available for Google Calendar API client library	66
6.1	OSSIM's Social Media SQL information	91
6.2	OSSIM's Meteorological SQL Information	93
6.3	OSSIM's Calendar Event SQL Information	95
6.4	OSSIM's Terror Level SQL Information	97
7.1	Table of event number with number of alarms raised during each test	100
7.2	Table of event number with number of alarms raised during each test with Social Media Context Level Low	104
7.3	Table of event number with number of alarms raised during each test with Social Media Context Level Medium	105
7.4	Table of event number with number of alarms raised during each test with Social Media Context Level High	105
7.5	Table of event number with number of alarms raised during each test with an Extreme Meteorological Context Level	109

7.6	Table of event number with number of alarms raised during each test with a current day as a public holiday . . . . .	111
7.7	Table of event number with number of alarms raised during each test with a low terror threat level . . . . .	114
7.8	Table of event number with number of alarms raised during each test with a moderate terror threat level . . . . .	115
7.9	Table of event number with number of alarms raised during each test with a substantial terror threat level . . . . .	115
7.10	Table of event number with number of alarms raised during each test with a severe terror threat level . . . . .	115
7.11	Table of event number with number of alarms raised during each test with a critical terror threat level . . . . .	116
9.1	Context types used for the Contextual Data Applications . . . . .	123
9.2	Context type APIs used for the Contextual Data Applications . . . . .	124
9.3	Context test use cases used for the Contextual Data Applications . . . . .	125
9.4	Results showing the average number of alarms from each section of contextual data . . . . .	127

# Chapter 1

## Introduction

Network security has become a necessity to ensure the safeguard of sensitive information. A majority of devices, whether they are hand held, desktops or servers require security considerations because of all the threats that come hand in hand with having these devices connected to a network and, in most cases, access to the Internet. Network and information security has been in the spotlight following the exploits such as the Heartbleed bug [1], the release of sensitive information regarding the NSA and its surveillance of private information such as phone calls, emails etc. or even the attack of the StuxNet worm on a Uranium enrichment plant in Iran [2]. These news reports only reinforce the fact that more effective and efficient network security measures are needed.

The implementation of a SIEM system is one such security measure that has proven to be effective. A SIEM or, Security Information and Event Management system, combines the two functionalities of a security event management system (SEM) and a security information management system (SIM)[3]. This means that a SIEM is responsible for both real-time monitoring, correlation of events and notifications as well as long-term storage, analysis and reporting of log data. The SIEM will alert an analyst once a potential threat is detected on the network, however the SIEM can produce false-positives or miss threats entirely [4].

To improve a SIEM's accuracy in identifying threats, contextual data can be added to the security event information that the SIEM analyses. By deploying multiple collection agents throughout the network, the SIEM gathers different types of security information relating to different types of devices on the network such as end-user devices, network equipment and servers. [5] Since these collection agents are collecting large amounts of different, non-normalised data relating to a variety of devices, the SIEM is required to normalise the data in order to be effective. Normalised data means that the SIEM's statistical correlation engine can attempt to find relationships within the security event information. The SIEM also has a rule-based system that analyses the security event information to find known virus signatures. In both cases, the more information the SIEM has to work with, the better or more accurate its processing of the data will be.

This is where the idea of contextual data with the SIEM becomes important and is achieved by adding another dimension to the collection of data that relates to the different

security events, resulting in the SIEM having more information concerning the context of the event. In a paper by [6], it states that “Context is usually vital to allowing you to separate false positives from true detection. It is the difference between detecting an actual attack, rather than chasing after a merely misconfigured system”. The context of a security event could help the SIEM make more accurate decisions when conflicted with an event that is borderline malicious. This would lead to less false-positive alerts being generated by the SIEM. The coupling of security events with contextual information would also be helpful to the network analyst as more information about the event will be available for review.

This investigation of the role and impact of contextual data will compare the same SIEM - one SIEM using contextual data coupled with security event information and one using the standard security event information. Within the test case of the SIEM that is using contextual data, an evaluation for the most appropriate types of contextual data can also be run to find out which types of contextual data actually enrich the security event information being fed into the SIEM.

## 1.1 Hypothesis

It is proposed that by augmenting existing security event information feeds with contextual data, this will improve the performance of a SIEM. The term ‘performance’ needs to be clearly defined as different performance metrics relate exclusively to what is required of a system. In this case, performance will be directly related to the SIEM’s threat detection capabilities. When looking at the detection capabilities metric, the amount of alarms the SIEM detects during each trail will need to be considered. Before contextual data can be added to improve the detection capability, a control test of the detection capability will need to be done.

The augmentation of security event information with contextual data can then be tested with the same security event set. The contextual data will be another variable that needs careful consideration because different types of context will contribute differently. Moreover, the SIEM’s correlation ruleset for different information inputs, namely the contextual data feed, will need to be amended with new rules taking into consideration the new stream of information. These rules will be constructed to make sure that their implementation does not degrade the SIEM’s security analysis performance. With these considerations in mind, the hypothesis is:

*The augmenting of security event data with contextual data will improve a Security Information and Event Management System’s threat detection capabilities.*

The above hypothesis will attempt to investigate if the inclusion of non-security, contextual information into a security system can improve the system’s overall security analysis capability. This hypothesis aims to bring more information together in a way that is useful from a cyber security stand point and increase overall visibility of the monitored network.

# Chapter 2

## Background Research and Prerequisite Concepts

### 2.1 Introduction

In this chapter, past and current related work is assessed and considered. It is important to have a solid understanding of the topic at hand. In this chapter, the inner workings of a SIEM are discussed as well as why a SIEM is even necessary. After our knowledge of SIEM is proficient, different types of SIEMs are considered along with the different types of security event information that can be sourced into the SIEM.

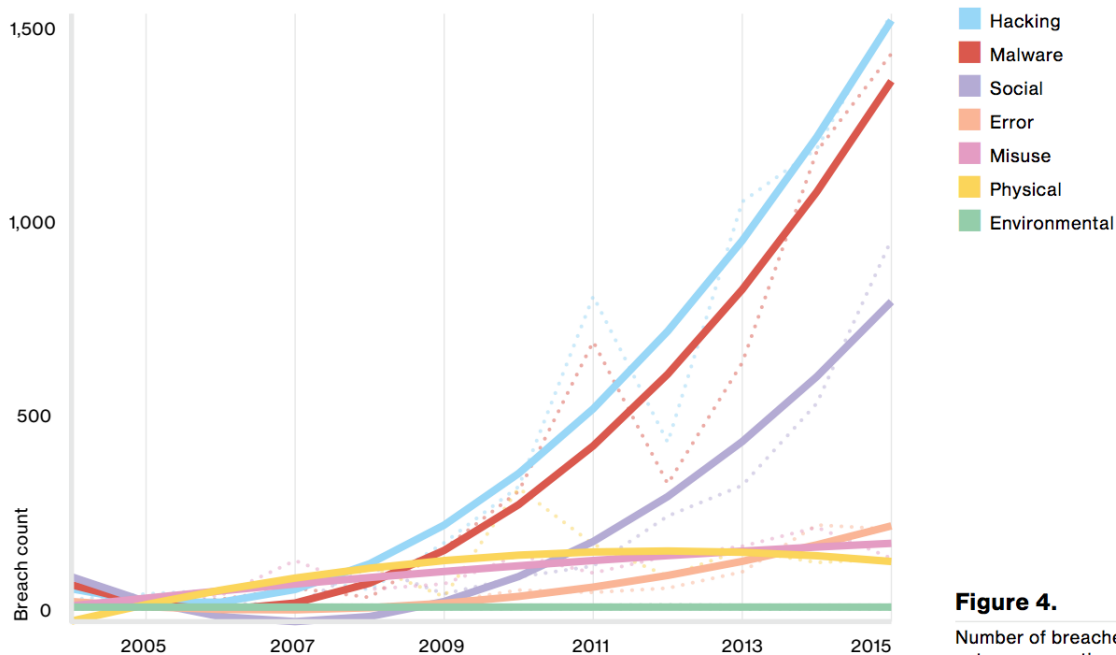
The matter of information or data is very important because not all information is useful, and as we know, computers tend to generate an incredible amount of information. The larger the network of devices generating security event information, the more information the SIEM will have to analyze and correlate. Hence, the more meaningful information fed into the SIEM, the more efficient the SIEM will be.

### 2.2 The Importance of Cyber Security

The importance of cyber security has grown with headline-making internal breaches such as the Wikileaks 'Cablegate' incident or Stuxnet, which was an example of state-sponsored cyber warfare in an attempt to tamper with Iran's nuclear programme. These incidents are on the increase and can be seen as IT security failure. In the case of the 'Cablegate' incident, an authorised person stole the data which means this incident could also be seen as a failure of policy or protocols. However, the result is heightened concern for sensitive data and the effectiveness of the security protecting it [7].

Verizon has taken to compiling data breach reports which outline the focus of data breaches from around the world and from thousands of different companies. The report clearly shows an increase in data breaches and hence a continual lack of network and cyber security [8]. The report also shows the types of data being stolen and the method in which this data is stolen - with the majority of breaches coming from hacking, malware and social

phishing. Malware and hacking are methods that this thesis's results could help deter. Figure 2.1 is a snapshot of data breaches over the years.



**Figure 4.**

Number of breaches per threat action category over time, (n=9,009)

**Figure 2.1:** [8]

Network and internetwork security is an ever expanding field. As we develop more devices that can connect to each other, hence joining a network, we must understand that each device will inherently come with its own security challenges. The internet is connecting people from around the world. This means that inadequate security could lead to your personal network and its devices being compromised by someone half way around the world. Unfortunately with network defence, the attacker need only be right once while the defender needs to be correct every time [9]. With the existence of digital intellectual property and company networks (Intranet) containing sensitive commercial data, the need for network and internetwork security has become a critical infrastructure.

Network security is a complex process because of all the different security considerations. The first emphasis would be that for a network to be considered secure, the entire network must be secure. This means that not only must each device on the network be secure, but so must the communication channels between these end-points. The following should be considered when trying to secure a network: [10]

- Access Control - only authorised users should be able to communicate throughout the network
- Confidentiality - all data in the network should be private to the network.

- Authentication - all users that are on the network should be checked that they are who they say they are.
- Integrity - that data being transferred between users has not been modified whilst in transit.
- Non-repudiation - users can not deny actions that are linked to their account because their account belongs only to them.

There are tools available that guard against certain security vulnerabilities. One of these tools is a security information and event management system which monitors your desired network so that suspicious activity on the network is recognised and a call to action is sounded. This tool is the base of this thesis and it is discussed in more detail in the next chapter.

In order to fully secure a network, an understanding of the types of attacks that threaten the network should also be understood. This way intrusion detection systems can be developed to defend against these attacks. When considering network intrusion, it can be helpful to understand why an attacker might be trying to cause problems. Usually packets are introduced to the network for the following reasons [11]:

- Consume resources that are available on the network, uselessly
- Interfere with different system resource's intended function
- Gain knowledge that can be used to exploit the network later in an attack

Now that we know different attacker intents and security considerations, developing a system that can handle and anticipate attacks is nearly within our grasp. The last network security topic to examine is different types of attack methods. Although there are many different variations of the following attack methods, they are categorised in Table 2.1:

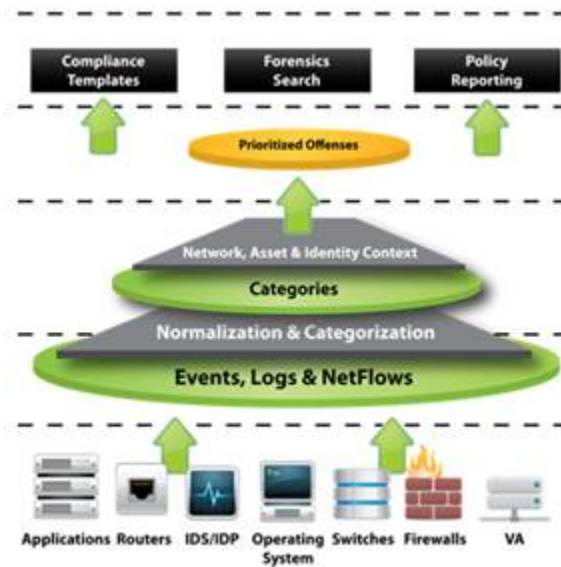
**Table 2.1:** Table showing different attack methods [12]

<u>Attack Method Intent</u>	<u>Attack Method Name</u>	<u>Description</u>
System Knowledge	Eavesdropping	An unauthorised party attempts to intercept communications and listen in on this channel to gain sensitive information
	Phishing	An unauthorised party attempts to trick someone into entering sensitive details that the unauthorised party can then record and use.
	Brute Force	An unauthorised party attempts to gain access to the network by repeatedly attempting to login using guessed account details
Interfere with System's Functions	Virus	A self replicating program that uses files to infect and propagate.
	Worm	Similar to a virus, but a worm propagates either through email (mass mailing worm) or through a network (network-aware worm).
	Trojan	A program that appears to be harmless or helpful but actually has some malicious purpose.
Waste System Resources	Denial Of Service	A system starts receiving too many requests, hence it can not return communication with the requestors. This leaves the system consuming resources waiting for the handshake to complete and hence the system can not respond to more requests.

There are many more risks involved in securing and maintaining a secure network, however these are the basic attack methods that should be anticipated. In conclusion, the need for fast and effective network security is growing steadily as the size of our networks grow. The larger the network, the harder the network is to secure because of all the possible entry points presented to our attacker.

## 2.3 Overview of a SIEM

Gartner describes a security information and event management system as a technology that “supports threat detection and security incident response through the real-time collection and historical analysis of security events from a wide variety of event and contextual data sources. It also supports compliance reporting and incident investigation through analysis of historical data from these sources.” With ‘big data’ being the interesting new topic in technology, the idea of gathering, analysing and large amounts of data has been used by SIEM systems for a long time as a commercial security solution [13].

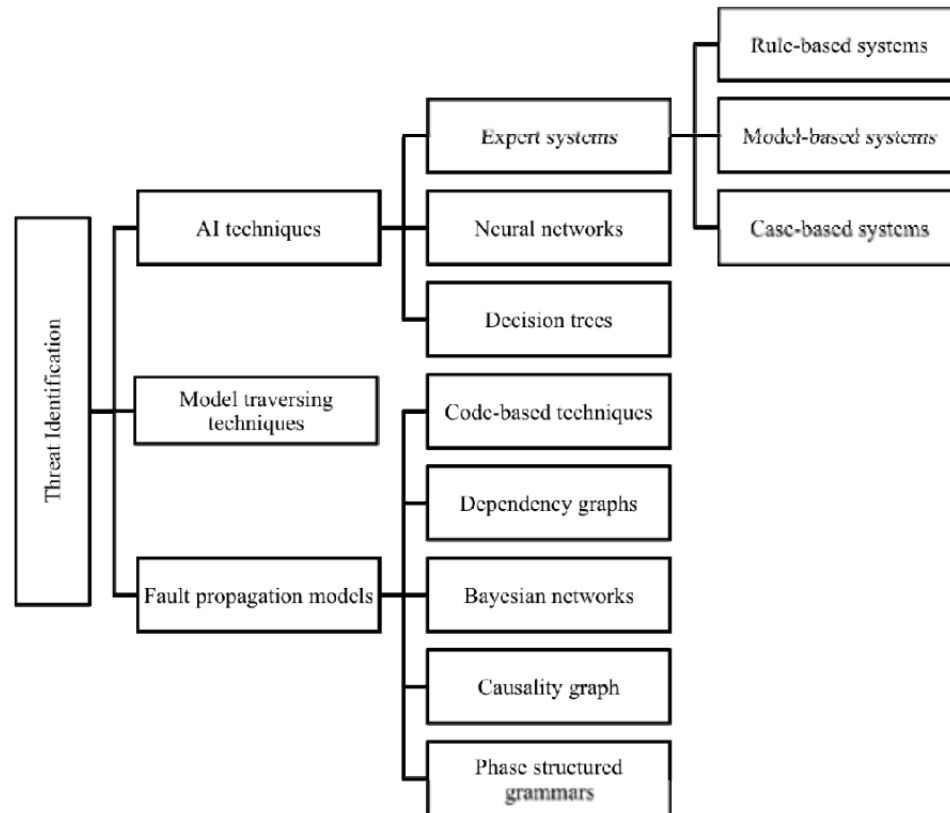


**Figure 2.2:** Diagram showing the general work flow of a SIEM [14]

“Security information and event management systems are an emerging technology that can significantly enhance critical infrastructure protection” [3] and it is considered to be the core of developing a personal, and often, commercial security toolkit that would be used to defend your network against potential risks [15]. Figure 2.2 displays each of the general processes that a SIEM undergoes. As you can see the SIEM will collect data from important network devices throughout the network in the form of security events and logs. It will normalise these events into a useful form. Once these logs are collected and normalised, different types of SIEM use these logs with a range of analytical approaches to determine whether there is anything suspicious happening on the network.

A statement from a paper by [7], “As compliance moves from best practice to mandatory” tells us that certain security compliances needs to be met. In the same paper, it states that all compliance frameworks are based on policy and hence the value of a SIEM is exposed. A SIEM helps understand current processes and workflows, to allow the monitoring of policies. To meet compliance, an organisation needs to know everything going on inside their network, across applications and devices.

Another value of a SIEM is found in its ability to collect and store security related event logs from different network systems as well as make accurate use of this data. Figure 2.3 displays the many different techniques used in threat identification. The techniques relevant to our research will be discussed further in chapter 3.



**Figure 2.3:** Figure showing different Threat Identification Techniques [16]

Since not all work environments are the same, the rules that govern the SIEM are easily customizable to suit specific security needs. Through these rules, the SIEM will check for certain types of behaviour on the network and alert the network administrator if suspicious behaviour has been flagged. Since networks, especially commercially, tend to get quite large, a SIEM would have to run through millions of event logs. Hence, it is common for there to be many false positive alerts generated from the SIEM. A paper by [17], states that a 'principal objective' for optimising a SIEM would be reducing these types of alerts. It would then be the job of the security operations' (SecOps) personnel to determine whether the flags are real risks or just false positives.

As mentioned earlier, the customizability of the rules that govern the SIEM can be important when considering different work environments. In the same vein, the types of event logs that SIEMs collect from the network are also highly customizable because a simple collector would need to be developed to fetch these event logs and parse them into a useful format that the SIEM could use, along with a set of rules pertaining to that type of event log. This allows the SIEM to collect event logs from nearly any type of device on the network and hence gives the SecOps a profound visibility of the network itself.

The main objective of a SIEM is to achieve full network visibility in order to detect threats

and vulnerabilities. This means that, although networks are often very large and there are many different points of vulnerability for an attacker to exploit, we still want our SecOps to be able view as many actions on the network as possible - thus making them more effective. It is also necessary to continually analyse available vulnerabilities and evaluate security levels [18]. By correlating all the event logs from different devices throughout the network, the SecOps has a better idea of what is actually happening in the network.

In a paper by [13], he states that “organisations who have taken their implementations seriously with the appropriate levels of budget and resources find SIEM invaluable in protecting themselves from cyber-threats and criminals”. The biggest problem is that companies do not budget enough for the constant care and special attention that a working SIEM needs. Because SIEM solutions are so expensive, it can be difficult for smaller companies to implement and maintain. The following section covers a few SIEM solutions and the chosen solution too.

## 2.4 AlienVault and its Open Source SIEM

### 2.4.1 Introduction

AlienVault provides a multitude of commercial security software solutions. AlienVault have built a unified framework which is available as an open source product or a commercially supported product. Their main security event and information management product is their Unified Security Management (USM) platform while their open source solution is OSSIM, or Open Source Security Information and Event Management. OSSIM provides all the usual functionalities of a commercial SIEM such as event collection, normalization and correlation the tools necessary for basic network security visibility. The OSSIM platform is the product of many tried and tested open source security controls all built into one. OSSIM provides a web UI and console access [19].

The web UI provides informative graphs and flagged events allowing for easy analysis of the incoming security event information from the network. The console allows for a more profound customization of the platform which encourages security beginners and professionals alike to make OSSIM into a platform that will work optimally in their unique environment. AlienVault also host a service called the Open Threat Exchange which is a crowd sourced threat intelligence exchange. This open threat information sharing and analysis network allows for anyone to join the community and participate.

The choice of using AlienVaults OSSIM was not a straightforward decision. In a paper by [17], the decision to use OSSIM is the fact that it is the de facto standard Open Source SIEM. It was important that we understood what OSSIM offered in comparison to its competitors. According to Gartners 2015 Magic Quadrant for Security Information and Event Management by [20], the leaders are:

- IBM QRadar Security [21]

- HP ArcSight [22]

It should be noted that these security solutions are all large enterprise solutions which means that although their capabilities are extremely effective and vast, their price tag is equally extreme. Security has become such a global issue in recent years that the price for quality security of your network and enterprise software has become very expensive.



**Figure 2.4:** Figure showing the Magic Quadrant for Security Information and Event Management

In figure 2.4 [20], we are presented with scale of 'ability to execute' vs 'completeness of vision'. Although these are hardly metrics, it is Gartner's Magic Quadrant format which rates different software and places them appropriately. This figure has been included to introduce AlienVault's competitors as well as give the reader a visual summary of competing SIEM

software. The evaluation criteria for each of the axes are based on a number of categories. The criteria for 'ability to execute' hinged on the core services offered, the viability of the product, the customer experience with the product, operations and marketing. The criteria for 'completeness of vision' hinged on market understanding, innovation and strategies for the market, sales, offering, industry and geographic types.

### 2.4.2 Summary Evaluation of SIEM Products

The following information is taken from the Gartner's 2015 Magic Quadrant for Security Information and Event Management [20]. It is a summary of some of the leaders in the SIEM industry along with an analysis of AlienVault's Unified System Management product. Although OSSIM is not AlienVault's USM, it is still relevant to look at AlienVault's USM in comparison to the leaders in the industry in order to ascertain whether AlienVault as a security software solution company is effective and trustworthy.

#### Analysis of AlienVault

The AlienVault security event and information management solution is based on OSSIM (Open Source SIEM). Although this solution provides a SIEM, vulnerability assessment, NetFlow, network and host intrusion detection, and file integrity checking monitoring, AlienVault also offers a commercial product called the Unified Security Management (USM) which extends OSSIM but offers scaling enhancements, log management, reporting and multi-tenanting for managed security service providers. The USM product allows for an all-in-one solution which takes the sensor, logger and server components and integrates them into one server however, for larger enterprise environments these components can be deployed in several tiers to match the size of the environment.

As mentioned earlier, AlienVault has an Open Threat Exchange community which enables sharing of Internet Protocol (IP) and URL reputation information throughout all its clients. AlienVault Labs provides an integrated threat intelligence feed to its commercial products that includes updates to signature, vulnerability, correlation, reporting and incident response content. To aid with easier deployment of the AlienVault platform, several wizards have been added to support first time users with asset discovery and configurations. The platform also offers a dashboard feature to support maintenance of the network and all host-based sensors and controls. AlienVault is aimed at organisations that need a broad set of integrated security capabilities at a relatively low cost.

<u>Strengths</u>	<u>Weaknesses</u>
AlienVault USM provides integrated capabilities for SIEM, file integrity monitoring, vulnerability assessment, NetFlow and both host-based and network-based intrusion detection systems.	Although AlienVault has recently expanded the number of predefined correlation rules for third-party commercial products, some existing customers identify this as an area that needs further improvement.
Customer references indicate that the software and appliance offerings are much less expensive than corresponding product sets from most competitors in the SIEM space.	Identity and access management (IAM) integration is limited to Active Directory and LDAP monitoring, and application integration is primarily with open-source applications.
	AlienVault's workflow capabilities do not include integrations with external directories for workflow assignments.

**Table 2.2:** Table showing the Strengths and Weaknesses of AlienVault

Now that the analysis of AlienVault is done, we can look at its competitor's to get a good understanding of the ranking between SIEM solutions.

### Analysis of IBM Security

IBM Security offers a SIEM technology called QRadar which provides the usual SIEM capabilities of log management, event management and reporting. It also offers behavioural analysis for networks and applications. QRadar allows for an all-in-one setup which is aimed at smaller enterprise environments and for horizontal scaling which is used for larger environments. In these larger environments, QRadar uses specialised event collection, processing and appliances. QRadar's distinguishing factor is its collection and processing of NetFlow data, deep packet inspection, full packet capture, and behavioural analysis for all supported event sources.

QRadar has included several new features such as Incident Forensics which extends event flow analysis, and integrated vulnerability scanning via its QRadar Vulnerability manager. There have also been some API enhancements and an improved search performance. Along with QRadar, IBM offer a service option which combines on-premises QRadar deployment with remote monitoring from IBM's managed security services operations centers.

<u>Strengths</u>	<u>Weaknesses</u>
QRadar provides an integrated view of the threat environment using NetFlow DPI and full packet capture in combination with log data, configuration data and vulnerability data from monitored sources.	QRadar provides less-granular role definitions for workflow assignment compared with competitors' products.
Customer feedback indicates that the technology is relatively straightforward to deploy and maintain in both modest and large environments.	QRadar's multitenant support requires a master console in combination with distributed QRadars instances. The number of third-party service providers that offer QRadars-based monitoring services is limited when compared with vendors that lead in this area.
QRadar provides behavior analysis capabilities for NetFlow and log events.	

**Table 2.3:** Table showing the Strengths and Weaknesses of IBM's QRadars

### Analysis of HP

HP's Enterprise Security Products (ESP) is the department of HP that is behind HP's ArcSight line of SIEM solutions. ESP is also behind such products as HP TippingPoint and HP Fortify - other security solutions that have been successful. The ArcSight line includes ArcSight Enterprise Security Manager (ESM), ArcSight Express, ArcSight Logger, ArcSight Risk Insight and ArcSight Application View. Each of these products are aimed at providing separate functionality but integrate seamlessly. ArcSight ESM is oriented toward large enterprise deployments, while the ArcSight Express is an appliance designed for the midmarket with preconfigured monitoring and reporting. The ArcSight Logger provides log collection and management functions. ArcSight Risk Insight provides risk ratings and a management dashboard for security event data. ArcSight Application View enables application activity monitoring not dependent on log data.

Recently HP has made changes to its SIEM technologies to combat its previous inhibitors - namely cost and complexity. HP has introduced an implementation called simplified events per second (EPS)-based pricing model to make the solution more affordable, while their new ESM version saw the replacement of its Oracle Database with the Correlation Optimised Retention and Retrieval Engine (CORR-Engine). The CORR-Engine has been validated to have increased event-handling capacity. ArcSight Express should be considered for midsize SIEM deployments. ESM is appropriate for larger deployments, as long as sufficient in-house support resources are available.

<u>Strengths</u>	<u>Weaknesses</u>
ESM provides a complete set of SEM capabilities that can be used to support a security operations center.	ArcSight provides real-time statistical correlation, but profiling and anomaly detection operate against historical data only.
ArcSight Express provides a simplified option for midsize SIEM deployments.	While the CORR-Engine has eliminated a major source of deployment and support complexity, customers will still find ESM to be more complex than other leading solutions.
ArcSight Logger can provide an inexpensive log management capability for two-tier deployment architectures that require long-term event archiving.	
Optional modules provide advanced support for user activity monitoring, IAM integration and fraud management.	

**Table 2.4:** Table showing the Strengths and Weaknesses of HP's ArcSight Solution Line

### Summary of AlienVault and Competitor's Analysis

As seen by the analysis done above, AlienVaults product, although not the top of the range, still has considerable capabilities. AlienVaults weaknesses are not necessarily too much of a weakness for this thesis. AlienVault focuses its application integration on open-source applications. This allows the product to integrate seamlessly with other open-source applications which encourages the products use by smaller enterprises, students (such as myself) and confident security professionals. This category is added because open-source means that any security professional who is confident enough to code extras for their system, can do so easily with AlienVaults OSSIM. Naturally, due to the focus of AlienVault on the open-source community, their commercial product is also less expensive compared to the other SIEM solutions offered by their competitors.

The analysis above does, however, show points of interest for AlienVault to improve upon. Identity and access management is a problem area for AlienVault whereas this is strength for HP. This capability is important when dealing with large enterprises that have many employees constantly accessing the network from multiple devices. Splunk offers built-in support for a large number of external threat intelligence feeds. This is an important strength because threat intelligence would allow protection against even the latest attack patterns and malicious software. However, AlienVault does have community-supported threat intelligence which is perfect for the sake of this thesis.

The following table lists the offerings of AlienVault's OSSIM as compared to its commercial counterpart AlienVault's USM:

<u>AlienVault's OSSIM</u>	<u>AlienVault's USM</u>
Limited log collection and log retention only for SIEM Events	Robust log management, log search and long-term log retention
3 high-level reporting templates	150+ customizable reports
Flat and single server deployments	Multi-Tier architecture with multiple servers
Single User	Multi-user with role-based access control
Individual component management	Centralised administration and configuration
Community support	Professional Support

**Table 2.5:** Table showing OSSIM vs USM as AlienVault Products

As you can see in table 2.5, OSSIM has a subset of the capabilities of USM. This is to be expected since it is their commercial product and is designed for business enterprises. However, we can see that OSSIM has sufficient capabilities for our test environment which will test the hypothesis. By using OSSIM, and proving this hypothesis, we will know that adding contextual data to commercial SIEM solutions will probably yield even more successful results.

## 2.5 Security Event Information

Since there are many different types of devices that can connect to a network, there are many different types of security event information. It is important to know which type of security event information should be collected and fed into the SIEM for collection and evaluation. The array of devices and applications on a network will provide log data or machine data which is a mess and hence making sense of this data is a massive challenge [13].

Event information can be defined as an observable occurrence in an information system that actually happened at some point in time [23]. Hence, to extend this to security event information, the event information would need to have some security relevance. This security relevance is exactly why the type of event information is so important because millions of irrelevant logs would simply slow down the SIEM unnecessarily. A SIEM is only as useful

as the information feed into the system [6].

Possibly one of the most important sources of security event information comes from log files. Log files contain invaluable information because these files contain a record of each and every process that has happened on the device. Through log files, background processes become a lot more transparent because it is now easy to see what resources are being used on the device. SIEM is about collecting logs, and mapping information about your infrastructure and business processes to those logs [6]. Attempts to connect to random server addresses, suspicious login attempts and strange memory requests would cause the SIEM to flag these events among other things.

There is a whole review process that is adhered to when considering what events are going to be allowed to be fed into the SIEM system. Since devices can often provide an array of different kinds of logs, the network administrator is tasked with reviewing the different types of logs coming out of each device and deciding which provide the most value to the SIEM system.

Apart from which logs are important, there would also need to be a check to determine which devices would be providing the most valuable security event information. These systems or devices typically fall into three different categories [24]:

- Security systems. This category contains systems and devices that provide a security function on your network. An example would be authentication systems, firewalls, network intrusion detection and prevention systems etc.
- Business critical systems. This category contains systems that are vital the running of the business smoothly throughout the entire organization. An example would be a server that hosts banking details and records.
- Critical infrastructure systems. This category contains systems that are important for your network to keep running smoothly. It would be wise to identify which infrastructure systems are critical to your network and what the impact would be if these systems went offline. An example of such systems would be mail servers, DNS servers, web servers etc.

As you can see from the above, the choice of information to feed the SIEM is definitely not limited. Since there are so many different devices and different types of logs available, it is a case of too much readily available information rather than too little information. Once the devices that are going to be monitored have been decided upon, it is recommended to decide carefully upon the types of logs. These logs are categorized into 4 important categories according to [25]. Please keep in mind these log examples are simplified for explanation purposes. Section 2.5.6 show cases real log file examples:

### 2.5.1 Authentication and Authorization Logs

Authentication logs typically identify successful and failed access attempts to certain systems and services on the network, while authorization logs identify specific users activities and

their use of higher privileged capabilities on the network. Authentication is very important in today's systems because it is the main defence against anyone accessing your network or services. Authorization is important because even if you gain access to the network it will restrict services unless you have the appropriate privileges.

These logs can be discussed in more detail and are identified through the following events:

- Login failures and successes by users and different systems: the log report for these events would track both login successes and login failures across different systems, users and access methods.
- Login attempts (both successes and failures) to disabled/ non-existent/ suspended/ default/ service accounts: the log report for these events would show attempted access to accounts and services that should never be accessed.
- All logins after hours: the log report for these events is valuable because it is registered as suspicious activity and warrants checking by the security administrator of what is being accessed.
- VPN authentication and other remote login attempts (successes and failures): the log report for these events are similar to the above categories but these are of an escalated interest to the security administrator because most attackers will use remote logins to protect themselves from being traced.
- Privileged Account accesses: this category is always monitored very carefully because root logins, su use and equivalent for other systems mean that this privileged user can do way more damage than a normal user can. This type of access is always very limited to a select amount of individuals.
- Multiple failed login attempts followed by a successful login attempt: the log report for these events are of high interest because it is the pattern of a brute force attack.

Below is a typical example of the above types of logs:

<u>System</u>	<u>Account Name</u>	<u>Source IP</u>	<u>Status</u>	<u>Method</u>	<u>Count</u>
Venus	administrator	10.1.1.2	Failure	Local	1
Jupiter	anton	10.11.12.13	Success	Local	1
Mercury	root	10.1.2.3	Failure	SSH	893765

**Table 2.6:** Table showing authentication and authorisation logs

Points of interest in Table 2.6 would be the account name, such as root and administrator, because they would have escalated privileges. Another point of interest would be the method of access “SSH” because we know this is a remote login. Lastly, the very high count number of failed login attempts into the root account from the same IP address would be alarming since this is a clear sign of a brute force attempt to guess the root password.

### 2.5.2 Change Logs

These logs would identify critical security changes to various systems in general or on the network. These would typically be changes to configuration files, accounts, sensitive data and systems (or applications). These logs are important because changes to any systems or such critical files would need to be authorized and monitored, so as not to jeopardize the security and integrity of all systems involved. Unauthorized changes would be a huge security concern and almost definitely considered a security incident.

The key logs in this category are:

- Additions/changes/deletions to users or user groups: a common attack would be to add an account with privileges, and then delete it after, so that the attacker can access the systems and get sensitive information at will. Monitoring these logs can prevent this kind of attack.
- Password changes and resets by users and by administrators to users: use these logs to check that appropriate passwords are being used and to ensure that dictionary attacks are not easily successful.
- Additions/changes/deletions to network services: these are as important as monitoring new or modified accounts because new network services can open up your network to a new range of attacks. In fact, any new service or new process should be tracked because new services or new processes can cause a lot of damage internally even if not network centric. Tracking these logs can be invaluable when determining who is responsible for “leaving the door open” per se.
- Changes to system files such as binaries and configuration files: these must be tracked because again, changes to these kind of files can cause huge damage to the integrity of the system.
- Application updates or installs (successes and failures): random attempts at updating an application or installing an application must be tracked because often an attack will attempt to patch malicious software onto an existing application or install malicious software that will help the attacker at a later stage.

Below is an example of some logs showing the addition of groups:

<u>Date</u>	<u>System</u>	<u>Account Name</u>	<u>Operation</u>	<u>Object</u>	<u>Status</u>
1/10/11 11:11AM PST	Venus	root	Account Added	anton	Success
1/11/11 11:11AM PST	Jupiter	anton	Group Added	sudoers	Success
1/10/11 11:11AM PST	Venus	root	Account Added	root1	Failure

**Table 2.7:** Table showing the change logs

Points of interest in Table 2.7 would be the “Success” status of “adding an account” and “adding a group” since this “account” or “group” may have inappropriate privileges. The attempt of a “root1” account addition is also a concern since somebody is obviously trying to create a second root account to have root privileges.

### 2.5.3 Network Activity Logs

Network activity logs would identify suspicious activity on the network being monitored. Suspicious activity is found by searching through network events or system events on the network. These are also tracked for regulatory and compliance purposes -discussed in more depth in 3.2. These logs are important because it is the main access point for attackers i.e. via the network that connects all the enterprise systems together.

The following are key log types in this category:

- All outbound connections from internal and DMZ systems: the log report of these events is useful because it would show the security administrator whether there is malicious software or users within the network trying to contact an external site. This is very useful in detecting intrusions and network abuse.
- All outbound connections from internal and DMZ systems made during “off hours”: often malicious software will wait for a safe time before attempting to contact an external site and hence tracking of this information is important.
- Largest file transfers (inbound or outbound) or largest sessions by bytes transferred: the log report associated with this type of event could easily lead the security administrator to incidents of data theft or bandwidth abuse.
- Web file uploads to external sites: by tracking these logs, a security administrator can ensure that outgoing files are going to intended web sites and attached email data is safe.

- All file downloads by content type and protocol: it is important to monitor what data is entering the network environment and this is done easily by tracking both protocol as well as actual content.
- Internal systems using many different protocols and ports: it is important to monitor these simply because it can be suspicious if an internal system suddenly starts trying to use many different ports and protocols to gain outside access.

The following example shows VPN access and activities:

<u>Date</u>	<u>VPN</u>	<u>User Name</u>	<u>System</u>	<u>Action</u>	<u>Status</u>	<u>Count</u>
1/11/11	VPN1	anton	antonlaptop	Login	Success	2
1/12/11	VPN1	anton	antonlaptop	Login	Failure	1
1/13/11	VPN2	root	Lapt19847	Login	Failure	77

**Table 2.8:** Table showing network activity logs

Points of interest in Table 2.8 would be the multiple root login attempts made on system “Lapt19847” through “VPN2”. Root access attempts are always an escalated security event because if successful, the supposed root user would have high privileges.

#### 2.5.4 Resource Access Logs

Resource access logs would identify the different resource access patterns of the different systems, applications and databases on the network. Learning about the usual resource access patterns helps spot irregularities and would lead us to incident detection. This is important because it allows the security administrator to keep track of what resources different users and systems are using. This can be used to detect insider abuse and fraud. Tracking resource access logs can also lead us to determining what resources have been corrupted and modified.

The key logs in this category are:

- Access to resources on critical systems at any point: any access to critical resources should be monitored because of the importance critical resources have in a system.
- File, network share or resource access(success, failure): the log report of these events can be useful resource audits and checking what resources are being made available to any user on the network.

- Top database users: this is useful for security usage tracking because most sensitive data is kept in databases and should really be accessed only by select users. Most production databases should not allow direct access to users or developers.
- Summary of query types: this kind of event log report would serve as a detection tool since only certain queries should be called.
- All privileged database user access: this is a common log event to track, since privileged access to any database should only be allowed to a very select group of users.
- All users executing INSERT, DELETE database commands: these are considered potentially damaging commands and hence should always be tracked.
- Summary of database backups: backups need to be monitored constantly because it presents an easy and clean method of gaining lots of data without being detected. This is a big issue for data theft.
- Top internal email addresses sending attachments to outside: tracking of attachments from internal systems is an effective way to detect insider abuse and data theft.
- Log access summary: reviewing access to all these logs is a good practice because, as we have discussed, logs contain some very important information.

Below is an example of all file access in a network:

<u>Date</u>	<u>Server</u>	<u>User Name</u>	<u>File Name</u>	<u>Access Type</u>	<u>Status</u>	<u>Count</u>
1/11/11	Win1	anton	Expenses.xlsx	Read	Success	1
1/12/11	Win2	anton	Roadmap.ppt	Read	Success	1
1/13/11	NFS	anton	Blank.docx	Write	Failure	37

**Table 2.9:** Table showing resource access logs

Points of interest in Table 2.9 are the “Write” access type. Trying to write should always be monitored because writing to a server can be a huge security issue. Also the amount of attempts to write to that server would also be a clear sign of malicious activity [25].

Once the types of security event logs and the devices from which they are taken, are all finalized, comes the decision on how to get these log events. Most systems and devices have taken the standard of syslog.

### 2.5.5 Syslog

Syslog is a computer message logging standard and a primary source of security event information. It allows the separation of the software that generates the syslog messages, the system that stores them and the software that reports and analyses them. These messages are generated by different programs running on a device and are logged together. These logs are then used for a number of helpful things including security auditing, general debugging and program analysis. One of the reasons these files are so important is that they provide visibility into each device's inner processes. It also works with two levels: a severity level and a facility level.

The severity level is a number associated with each message contained in the log, determining the severity of the event that has just been logged. The facility level is a different number associated with each message in the log but it is linked to keywords pertaining to the event. For example, a process logged with the facility level of 4 means that the event is linked to an authorization process or security process. [26]. This is very helpful because the SIEM will go through millions of logs each day, and facility and severity levels help filter out which event information will be processed.

### 2.5.6 Examples of Real Logs

The following log files come from two different systems: Microsoft Windows Server 2003 and McAfee Epolicy Orchestrator. The first figure in each section, figure 2.5 and figure 2.7, shows the description of each ';' separated value. As you will see there are many values which are populated into a single log file line and so each line has large amounts of information valuable to any security analyst. The problem however is making sense of all this information and drawing up conclusions based on this information. Keep in mind that thousands of these lines are generated every day.

The figures following the description figures in each subsection are real log lines from their respective systems. Notice how there is not necessarily data between each ';' because each log line is not required to be full. The system logs as much information as it can at that time and ensures that there is a constant stream of log files being generated. As seen in figures 2.6 and 2.8, the information, although human readable, is not easily interpreted as intelligent and useful information. This is why the parsing of log file information is so important. It creates organised, useful information from a relative information overload.

A common notation technique for context-free grammar is the Backus-Naur Form (BNF). It is often used to describe the syntax of a language [27]. BNF contains the following notation:

- `<>` surround categories or groups.
- `::=` is the definition of non-terminals.
- Symbols which never appear on the left hand side are terminals.
- `|` indicates a choice between two non-terminals.

We can now define the log file syntax using this BNF notation.

```

<digit> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<number> ::= <digit> | <digit><number>
<letter> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | r | s | t
           | u | v | w | x | y | z
<word> ::= <letter> | <letter><word>
<category> ::= <word> | <number> | <word><number> | <number><word>
<log-line> ::= <category>";" | <category><log-line>

```

**Figure 2.5:** Backus-Naur Form for the following log lines

Although BNF greatly simplifies what the actual log might look like, it is easier to see what type of values are used and in what combination. The following log files could have a more complicated BNF syntax if we chose to define each category as its own rather than a generic category, but this would be excessive since each category, fundamentally, uses the already defined combinations.

### Microsoft Windows Server

This is the server version of the popular Microsoft Windows operating system. Server's typically require a lot of attention with regards to maintenance and security, hence the log files generated are complex with lots of useful information.

```

InternalSequenceNumber;TsomID;TrapRequestID;InternalTimestamp;TsomTimestamp;
SensorTimestamp;SensorName;SensorNameDescription;SensorType;EventType;
EventValidity;EventClass;SecurityDomain;UserName;Domain;SourceIP;SourceIPPrivate;
SourceIPGeoCC;SourceIPGeoASN;SourceIPGeoLat;SourceIPGeoLong;SourcePort;
DestinationIP;DestinationIPPrivate;DestinationIPGeoCC;DestinationIPGeoASN;
DestinationIPGeoLat;DestinationIPGeoLong;DestinationPort;SourceThreat;
DestinationThreat;InternalSubScenario;InternalUseCase;EventID;TypeOfAction;
Workstation;OSType;SourceIP2;SourceIPPrivate2;SourceIPGeoCC2;SourceIPGeoASN2;
SourceIPGeoLat2;SourceIPGeoLong2;SourcePort2;UserName2;SecurityID;Domain2;AccountType;
LogonType;LogonID;LogonGUID;LogonProcess;AuthenticationMethod;ServiceName;ServiceType;
UserPrivileges;CallerDomain;CallerUserName;CallerLogonID;FailureReason;StatusCode;
SubStatusCode;ResultCode;CodeDescription;

```

**Figure 2.6:** Log file headers from Microsoft Windows Server 2003

The Figure 2.7 is a log line from the Windows Server 2003.

```

TBS;;1043152677;1325184445;1301616866051;1301604946000;fb2f0416b09c5e0611ee5319;;
UNKNOWN;SE_AUDITID_UNKNOWN_USER_OR_PWD;99;;bfc4b43aa7982ce69bbd279f;
675aa07dbab86489548c99c8;1063a7aafe5fc05734e935f4;70.87.126.251;1;1918;;;4958;;
;;;0;65;99;S-6.1;U-6.1.3;529;Logon Failure;6716b98571370bf62ce681a3;
WIN-XP/WIN-2003;70.87.126.251;1;1918;;;4958;675aa07dbab86489548c99c8;;
1063a7aafe5fc05734e935f4;ComputerAccount;3;;NtLmSsp;NTLM;;;
a6a6131d5ad16328f7a4cb06;a6a6131d5ad16328f7a4cb06;-;
Unknown user name or bad password;;;

```

**Figure 2.7:** Actual log file line from Microsoft Windows Server 2003

## McAfee Epolicy Orchestrator

This is an advanced, extensible, and scalable centralized security management software that unifies security management through an open platform. Since this product unifies so many different security intensive products, the logs it generates are complex with network activity and other interesting data.

```
AutoID;AutoGUID;ServerID;ReceivedUTC;DetectedUTC;AgentGUID;Analyzer;AnalyzerName;
AnalyzerVersion;AnalyzerHostName;AnalyzerIPV4;AnalyzerIPV4Private;AnalyzerIPV4GeoCC;
AnalyzerIPV4GeoASN;AnalyzerIPV4GeoLat;AnalyzerIPV4GeoLong;AnalyzerMAC;
AnalyzerDATVersion;AnalyzerEngineVersion;AnalyzerDetectionMethod;SourceHostName;
SourceHostIP;SourceHostIPPrivate;SourceHostIPGeoCC;SourceHostIPGeoASN;
SourceHostIPGeoLat;SourceHostIPGeoLong;SourceIPV4;SourceIPV4Private;SourceIPV4GeoCC;
SourceIPV4GeoASN;SourceIPV4GeoLat;SourceIPV4GeoLong;SourceMAC;SourceUserName;
SourceProcessName;SourceURL;TargetHostName;TargetIPV4;TargetIPV4Private;
TargetIPV4GeoCC;TargetIPV4GeoASN;TargetIPV4GeoLat;TargetIPV4GeoLong;TargetMAC;
TargetUserName;TargetUserIP;TargetUserIPPrivate;TargetUserIPGeoCC;TargetUserIPGeoASN;
TargetUserIPGeoLat;TargetUserIPGeoLong;TargetPort;TargetProtocol;TargetProcessName;
TargetFileName;ThreatCategory;ThreatEventID;ThreatSeverity;ThreatName;ThreatType;
ThreatActionTaken;ThreatHandled;TheTimestamp;
```

**Figure 2.8:** Log file headers from McAfee Epolicy Orchestrator

The Figure 2.9 is a log line from the McAfee Epolicy Orchestrator.

```
1;{B9E91CFB-E25C-4A15-9154-048279BAF020};460bfe9ebd6978c9bb94bd44;
2011-09-20 11:47:01.190000000;2011-09-20 11:34:37;{20DFB4E3-4F94-4655-9488-5C772907A079};
VIRUSCAN8700;VirusScan Enterprise;8.7;e5570dcc1584fadbb53cd5f8;70.86.127.246;1;1918;;;;;
6473.0000;5400.1158;AutoUpdate;;;;;;70.86.127.246;1;1918;;;;;;
e5570dcc1584fadbb53cd5f8;70.86.127.246;1;1918;;;;;e0559ee6821fb98336ece5a5;;;;;;
ops.update.end;1119;4;none;none;none;True;00000000000003112;
```

**Figure 2.9:** Actual log file line from McAfee Epolicy Orchestrator

## 2.6 Contextual Data

The idea of contextual data comes from the word context. Context can be described as the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood. In very simple terms, context allows a deeper understanding of an event or statement, both of which are extremely pertinent in network security. In network security, we want to understand the circumstances of a particular event because with a deeper understanding of the event, we can make a more educated decision as how to handle the event. “Context is usually vital to allowing you to separate false positives from true detection ” [6].

SIEMs collect millions of logs of network security data in order to attempt to recognize attack patterns and other various suspicious network activities. The more security network data that is fed into the SIEM, the better visibility the SIEM has of the network. Contextual

data would be an extra type of useful data that could be added to the SIEM, hopefully giving the SIEM a better understanding of the circumstances surrounding the security data.

The problem here becomes what types of contextual data are useful for the SIEM. Feeding the SIEM with extra data comes at a performance cost because there is more data to process. Also this contextual data would need to be abstracted into a format that is useful to the SIEM and its correlation engine. Additionally, the SIEM would need an extra set of rules against which to judge the contextual data feed against.

Taking into consideration the above drawbacks, adding extra contextual data feeds to the SIEMs usual security event data feeds isn't a trivial task because of the added complexity of rule integration - this is discussed in depth later. Especially when the goal of adding this feed is to improve the SIEMs operational efficiency. The first challenge will be deciding on what types of contextual data are necessary and ultimately helpful. Followed by an efficient way to parse and normalize this data since contextual data is very seldom structured. In the following chapter focusing on different types of contextual data, we discuss different types of contextual, what each different type's value and drawbacks are and how these could be implemented into a SIEM.

## 2.7 Threat Intelligence

Gartner defines threat intelligence as Evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subjects response to that menace or hazard. Basically, threat intelligence would enable a more proactive security because the threat information contains intelligent data about the threat or attack. Building a more proactive SIEMs, using more intelligent threat information to actively detect and respond to security threats is the new approach rather than just using historical data to fix vulnerabilities [28].

Since the natural evolution of data mining has presented users with easy access to information, it is safe to say that there is a large amount of very valuable information waiting to be utilized in the form of Cyber Threat Intelligence. The definition of threat intelligence is very important because there is a wealth of useless or unintelligent information available too. The next issue would be how an organization integrates the threat intelligence into their existing workflows, and how they can fuse it with new and existing defensive technologies, whilst making sure that the information they use is both credible and useful.

An issue that has cropped up is that of the difference between threat intelligence and threat information. Organisations are already overloaded with information that they need to interpret, assess and decide on whether or not it is useful, and so the addition of more information in the form of raw data regarding a threat is not helpful; this is threat information. Threat intelligence is actionable data. This means the information is easily useful to the security analysts that receive it. Table 2.10 is comparing the difference between threat information

and threat intelligence: [29]

<b>Threat Information</b>	<b>Threat Intelligence</b>
Raw, unfiltered feed of information	Processed, sorted information
Unevaluated when delivered	Evaluated and interpreted by trained intelligence analysts
Aggregated from every source	Aggregated from reliable sources and cross correlated for accuracy
Maybe true, false, misleading, incomplete, relevant or irrelevant	Accurate, timely, complete and assessed for relevancy
Not actionable	Actionable

**Table 2.10:** Table showing threat information vs threat intelligence

A security system that has a useful threat intelligence feed will always be up to date with the latest threats. An example of current systems that use threat intelligence is that of banned IP lists that are always being updated by various information security companies. These types of companies will deploy an array of threat detection methods in order to ascertain whether an IP is malicious. One such method is that of a honey pot. A honey pot will pose itself as a vulnerable system so that attackers think they can exploit the vulnerabilities. The attackers IP address is flagged as malicious and added to a banned IP list. These lists of banned IPs or malicious IPs are passed into these systems.

The system now has an updated list of IPs that need to be prevented from connecting to the network. Another interesting and more advanced method of threat intelligence is the use of IOC (Indicators of Compromise). These are forensic artifacts of an intrusion that reside in OS and network devices[30]. IOC's are used to categorized threats that might be utilised in any number of ways in cyber-security. These IOC's, indicators of compromise, are considered easily actionable types of threat intelligence because IOC's are tied to observables whilst observables are tied to measurable events or stateful properties. This information is used to search for, or potentially identify, compromised systems. Once a security incident is confirmed, and the type or pattern of compromise is known, it is formulated into an attribute [31].

The advantages of using threat intelligence along with IOC's can be easily summarized:

- Helps gain faster access to actionable security information
- Helps organisations share security threat intelligence with each other, also building trust.
- Is very easily created.
- Supports an intelligence-driven security model.

### 2.7.1 Threat Intelligence Standards

**OpenIOC** : This is an open source framework for sharing threat intelligence. It is an extensible XML schema for the description of technical characteristics. These characteristics would help identify a known threat, an attacker's pattern or methodology, or some other evidence of a compromise in the network or in a system. It was designed to enable the sharing of threat intelligence both internally and externally via a machine-digestible format.

The base OpenIOC schema is a framework that allows users to easily extend the base to suit all environments. It categorizes different forensic artifacts of an intrusion in XML. OpenIOC comes standard with over 500+ indicators already but since each environment is different, additional indicators from various sources can easily be added. OpenIOC can also be parsed or converted into different formats that could be useful to feed into other systems that would benefit from this information [32].

**Cybox**: stands for Cyber Observable eXpression, which is a structured language for cyber observables. As stated on their home page, "CybOX is a standardized schema for the specification, capture, characterization, and communication of events or stateful properties that are observable in the operational domain". It goes on to say that CybOX "provides a common mechanism (structure and content) for addressing cyber observables across and among this full range of use cases improving consistency, efficiency, interoperability, and overall situational awareness".

In an operational cyber domain, events that occur such as the value of a registry key, deletion of a file, or the receipt of an http, are considered cyber observables. It should be stated that cyber observables are different to cyber indicators. While cyber observables are a statement of fact, cyber indicators are cyber observables patterns with relevant contextual information that provide meaning and guidance around the observable patterns.

CybOX supports a range of cyber security domains including threat assessment and characterization, cyber situational awareness, incident response and indicator sharing. [33]

**STIX**: stands for Structured Threat Information eXpression, and as the website states, "is a collaborative community-driven effort to define and develop a standardized language to represent structured cyber threat information. The STIX Language intends to convey the full range of potential cyber threat information and strives to be fully expressive, flexible, extensible, automatable, and as human-readable as possible."

STIX evolved out of a need for a consistent way to automate and share indicators of malicious events or activity. Although the initial focus was on indicators, it was soon decided that STIX should be broadened to include related threat and mitigation information [34].

**Mitre**, a not-for-profit organization, operates multiple federally funded research efforts in many different areas. The one area, threat intelligence and cyber security, has lead them to develop both CybOX, STIX and TAXII. TAXII (Trusted Automated Exchange of Indicator Information) allows the sharing of actionable cyber-threat information across organisations. Although all three of these threat intelligence standards, are created by the same company and actually work very well together too, they do not need to be used in conjunction with each other. This allows companies or organisations the free use of these standards, adding their own details and enhancements at will [35].

### 2.7.2 Conclusion

In conclusion, threat intelligence is an effective way to share relevant data amongst trusted groups of organisations. In this way, chance of catching an intruder, is greatly enhanced, since there are more security teams working against them. The sharing of critical attack patterns and security event observables in a format that can be intelligently processed in real time allows organisations to communicate and mitigate attacks on their network. IOCs enhance operational capabilities of incident management because of this common format amongst incident reporters.

## 2.8 Optimising a SIEM

The process of optimizing anything would be to closely evaluate the current standing or state of the system, and then deciding which elements that make up this system could be improved upon. There is the rare case where the system, as it stands, is running optimally and so the addition of extra systems or modules would be the only solution to optimizing the system. In a paper by [3], it states that “existing SIEM systems lack several important features” which they remedy by proposing the integration of a decision system. This is an example of optimising a SIEM. In a paper by [18], the suggested approach to optimising a SIEM is considering different attack modelling approaches and security evaluation. The following are suggested ways to optimise a SIEM.

### 2.8.1 Optimising through Inspection

ReliaQuest, an information technology security consulting firm, has compiled steps that should be taken when attempting to optimize a SIEM solution. The following is a summary of the steps: [36]

#### Step 1: Conduct a SIEM Health Check

Develop a detailed assessment of the current state and performance of the SIEM. Analyse this assessment to identify inefficiencies. There are some components to check:

- Queries, Rules and system components such as OS updates, patches etc.
- SIEM back-up and recovery process.
- Implementation mapped to current business needs

### **Step 2: Integration Services**

Taking into consideration the detailed assessment:

- Ensure that all devices are pointing to the SIEM.
- Proper analysis of the cross-correlation capabilities; to ensure that they are being used correctly. This will enhance intelligence development.
- Ensure relevant data types are being sent to the SIEM.
- Analyse the incoming data. Checking that it is parsed correctly for the SIEM.

### **Step 3: Content Development**

Develop a platform or means to bridge the gap between the current SIEM state and the desired state. Possible ways to bridge the gap:

- Build custom connectors
- Create queries, reports and alerts
- Modify filter/rules to eliminate the false positives.

### **Step 4: Education and Training**

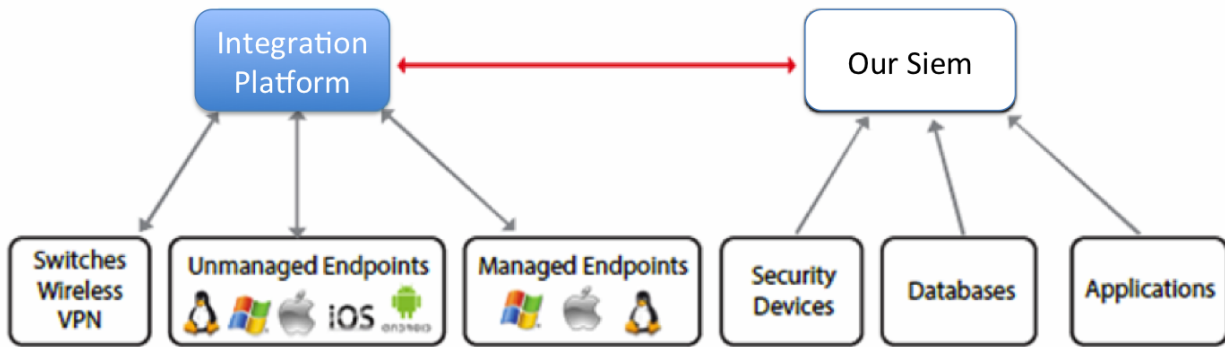
An important step is developing an education system that will ensure that the security team and other implicated employees are adequately educated on the proper procedures and processes. This can build organization wide security awareness and hence build towards the overall security goal.

## **2.8.2 System and Platform Integration**

One way to optimise an already existing SIEM solution would be to add other security based platforms that would ease the processing load of the implemented SIEM. These additional security systems can be added into any point of the entire SIEM's process. One such system is called a traditional policy enforcement product. This is an agentless solution that automatically discovers, classifies and applies policies for users, devices, systems and applications on the network. This kind of solution would help minimize security risks because it would identify security gaps that would be missed by a SIEM that uses only agent-based security measures.

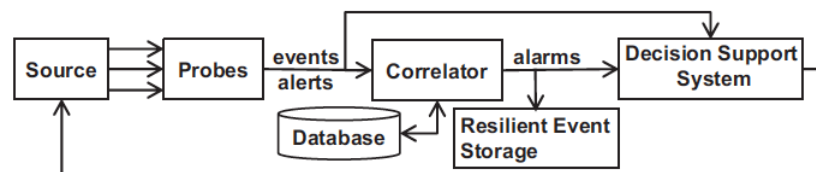
ForeScout's CounterACT is an example of platform integration that offers SIEM interoperability. This system offers end-point data visibility which means it makes sure that it

discovers all end-points on the network allowing for an accurate security snapshot of the network. This kind of integration would optimise a SIEM solution by gaining the SIEM a wider variety of information from all the systems on the network, thus a more complete protection. This is an example of a system that can be added to a SIEM to help with the processing load and even add more security functionality.



**Figure 2.10:** Figure showing integration platform handling external data sources.

Di Sarno. et al proposes a decision support system (DSS) integration[3]. This DSS would consist of two components: DSS-solver and DSS-analyser. The DSS-solver implements a customised policy conflict resolver. The strategy is to address and resolve conflicts between different policies but also allow conflicting actions. The DSS-analyser discovers unauthorised network access and allows redefinition of network configurations. This is a great example of integrating modules that would optimise a SIEM. This DSS would assist the SIEM's ability to detect and resolve conflicts between security policies.



**Figure 2.11:** Figure showing the proposed enhanced SIEM from [3].

### 2.8.3 Attack Modelling

Attack modelling is a method of building attack graphs or trees. An attack graph then represents all possible ways that an attacker could achieve their goal - the goal being a data breach or some unwanted consequence to the owner of the network[37]. An attack tree would be described as the root node, being the attacker's goal, while the leaf nodes are possible

ways to achieve this goal.[38]. Attacks can be represented in structured and re-usable tree-based form as seen in [39] and [40].

Different attack model approaches use different classifications to try to optimise the attack graph. By optimize the attack graph, it is meant that, as with any type of model, there are cases that the model does not predict, and hence to optimise the attack graph is to attempt to predict all attack paths. In [38] different classification approaches are described. The first is the classification of attacks into three dimensions, which branch into new nodes. The nodes then branch into new nodes until it can no longer be classified into any of the three dimensions: incidents, response, and consequences. The second is the classification of people into three dimensions: system role, reason of misuse, and system consequences. The final classification is a detection based approach and not prediction based. This classification takes misuse cases that happen at different layers of a system. These are classified into network-level misuses, system-level misuses, application and data-level misuses.

The difficulty with attack modelling is the computational complexity needed to develop a holistic model. [37] describes that to build a complete attack graph for one attacker is a computationally complex problem that usually takes a long time. When the graph must be built for a network with hundreds or thousands of hosts, devices and services the result should be obtained in a limited time (or even in real time). The algorithms used for these graphs require a very large amount of computational resources. Moreover, as time passes the composition of hosts and links between them can be changed, and the attack graphs will require reconstruction.

# Chapter 3

## Security Information and Event Management System

### 3.1 Introduction

At the base of this study is the use of a security information and event management system, or a SIEM. This suite of software consists of many security driven processes, working together to detect security anomalies on a chosen network. This network is monitored by the SIEM and uses many different tools - different SIEM products will have different tools depending on what the focus of the SIEM should be. However, there is a standard tool set found in most SIEM products.

Holistically, the term SIEM is a blanket term for the suite of underlying software. The following list quickly iterates over the underlying systems:

- Log Management System (LMS) which collects and stores log files from different devices on a network into a single location. This allows centralised access to these logs.
- Security Event Management System (SEM) which is an LMS geared towards security analysts instead of system administrators. SEM highlights security logs as more significant.
- Security Information System (SIM) which is an asset management system that incorporates security information such as anti-virus reports, intrusion detection and vulnerability reports.
- Security Event Correlation (SEC) detects a sequence of events that may be worthy of investigation.

[6]

The first standard process or tool that a SIEM will use is collection of an array of different security log files from as many different systems on the network. This is known as the log collection. These security event logs are then parsed and normalised into a format that the SIEM can use to extract important information. Once the information is available to the

SIEM, its correlation engine will run the security event information against a pre-determined rule set to look for attack patterns.

Next, the SIEM will store the security event logs for reasons involving compliance and analysis. And finally the SIEM will alert the necessary personnel if an anomaly has been found. The more advanced SIEM solutions have more functions such as vulnerability tests that are on the network and do not require security event log files, or web interfaces that allow the security administrator to have full visibility of the network at a glance.

## 3.2 Log Collection

The first important aspect to consider is how these security events are to be collected throughout the network. Collection is done in two ways: the agent-based approach or the agent-less approach. The agent-based approach provides more in-depth assessment, since it has to be installed on the endpoint device (server, desktop, network device). Using this approach, the security/network administrator has access to granular data on each of the hosts, and since the software is installed on the host, the agent can scan and perform checks whilst network availability is down. Hence, a constant connection to the network is not required making the agent-based approach an efficient bandwidth usage approach.

The agent-less approach means that software is not installed on the end-point device. Instead, a series of vulnerability tests are conducted and the results used. The agent-less approach also has its advantages. The obvious advantage is that the network administrator does not need access to all the end-point devices on the network. This greatly reduces cost of resources since not all end-point devices are the same. Hence, a variety of different software and expertise may be needed to manage different end-point devices.

Agent-less approaches also typically target devices where agent-based software is actually not possible or allowed (such as routers etc). It uses an asset discovery mechanism which utilizes UDP, TCP and other mechanisms. The network is scanned, then each asset is identified and classified. This feature is important because it reduces false positive alerts since each asset is known and so no confusion can take place. These scans also identify vulnerabilities and flaws on the network that could be exploited by an attacker.

There are of course disadvantages to both approaches and hence it is often encouraged, if not required, for companies to have both approaches. Agent-based is a more expensive solution since a higher and wider expertise is required for each device on the network. Also, not all devices on the network allow for agent-based software to be installed. Agent-based requires more host CPU processing power as well as resources.

The disadvantage of agent-less is the lack of depth in its assessment. To add to this, agent-less also requires that certain ports be open for the scanning process. Naturally, this could be bad for network security because it is opening extra ports. Certain scans will also be required to use certain ports on the router which requires a bit of expertise. The next

issue is the need for constant network connectivity since the scans do their job over the network. Any lack of network availability can lead to a vulnerable defenseless device.

The recommended approach would be to use both since they each have advantages. This is stated in the McAfee Total Protection for Compliance.

### 3.3 Parsing and Normalisation of Logs

The parsing and normalization of the log events is the next important aspect of the SIEM system's functionalities. Although many commercial SIEM solutions provide the customer with parsers for the different types of collectors, it is still relevant to understand this step since these parsers are not always available. Since the SIEM system uses a set of rules to compare events against, the SIEM needs the log event data in a certain form in order to extract useful information.

The data must be parsed as a first step. This is the process of taking data from a certain device in the network and making the event information that is specific to that device, understandable or useful to the SIEM system. This is usually done by developing a small program which accepts the device log event data as an input, and outputs that same data in such a way that the SIEM system can use this data with many different devices' event log data.

The next step is to normalise this data. The normalisation process is that of "breaking apart messages and organising their individual components" from the parsed data so that the SIEM system can store the data in a database. This is commonly recognised as a very labour intensive and technical process in the implementation of a SIEM system. [13] Although the storing of raw log files might also need to be present to comply with some of the IT security industry's policies, it is much faster and less process-intensive to search through a database than through many different log files. The normalization process also allows for a more space-efficient storage of millions of instances of security event log data.

Normalisation also comes with a few disadvantages. The process of normalization can become exponentially more complex with the increasing amount of different types of data-sources that are being normalized, since each data source will have its own data fields that need to be considered. Additionally, normalization of data into a specific database schema, such as a relational database management system, requires that only certain data be normalized and this means that some important data could be easily lost or ignored [41].

The other problem with this kind of data management is that extending an existing database schema is an incredibly complex task, so it is recommended to account for future schema changes. Naturally, this is a difficult task in itself. The solution to this problem would be implementing a NoSQL database, but this, again, is a complex implementation (and certainly not offered cheaply) [42].

The reason normalization is important, however, is the ability it gives the SIEM to do important analysis and reporting. It allows the security network administrator to query data across different platforms, such as requesting what a certain user has been doing lately. The user could have logged in to multiple devices across the network, but the common data would be their credentials, not their device. This is very useful because it allows the security network administrator to analyse security event log data without having prior knowledge of the systems that the security data may have originated from [43].

## 3.4 Correlation

In the field of network security, “Correlation is the act of linking multiple events together to detect strange behavior. It is the association of different, but related events to provide broader context than a single event can provide” [44]. The source goes on to say that this definition is quite broad, because the definition of the “event” used here is relatively broad since, as the breadth of analysis increases, the data may expand beyond traditional events.

Once the above discussed processes of event-parsing and normalization are done, the SIEM system can start using its correlation engine to detect known attack patterns. “A correlation engine has the job of analysing specific attributes of the log data, such as the users name, and can compare it with other contextual factors such as location, time, whitelists, blacklists, known vulnerabilities and so on. It needs to be able to do this in near real-time” [13]. A correlation engine allows the SIEM to be loaded with use cases which are used when building the correlation ruleset.

In very simple terms, events are run linearly against these rules such as “if X=Y then do something”. The difference is that these rules are often not that simple and are strung together to build composite rules. The more specific the rules, the greater chance that the attack is actually happening and the SIEM system is not reporting a false positive. The more generic the rules in the SIEM system, the greater the chance that it will find new attacks. This will, however, lead to more false positives.

The other challenge with rules, is that linking of different rules together for different types of devices' data. It is important for the network administrator to know which devices will have related data. Too many rules, and linking of rules, will require too much processing resources while too few rules and too few links will allow attackers an easy path through the network. This balancing act is constantly being fine-tuned to ensure that the SIEM system is running optimally.

A problem has already been presented in the form of checking for known attack patterns. This, of course, leaves all unknown attack patterns as a point of vulnerability. The next issue to consider with correlation, is that of time. Since events are not always happening simultaneously, a time factor needs to be considered, since events at different times can still be related [44].

## 3.5 Event Log Retention

The next important aspect involved in a SIEM is the storing of event logs or log retention. This is an important aspect because it allows companies to have historical data of what has been going on in their network. This means that a company can go back a few days and check for specific events, for specific devices. In most cases, attackers will leave a trail. This is a powerful tool because it can give an administrator insight into how an attack happened, and hence allow for the administrator to put in a safe guard against such future attacks.

Another important reason for log retention is meeting audit and compliance demands for large companies. “Compliance is one of the greatest challenges faced by organisations. Observing regulatory compliance audit policies is a requisite for every organization.” (IT Compliance and Regulatory Challenges). Since these large companies are trying to protect very sensitive information and also protect themselves from liability, there are industry requirements and compliance audits that require the retention of logs along with other security controls. The following are such standard compliances:

- PCI Compliance (Payment Card Industry) is comprised of credit card companies such as Visa, MasterCard, and Discovery who decided together to create industry requirements with the goal of reducing theft and fraud of payment card information.
- HIPAA Compliance (Health Insurance Portability & Accountability Act) demands much attention, resources, and money from the covered organizations to remedy their existing and planned systems and processes where protected health information (PHI) is involved.
- GLBA Compliance (Gramm-Leach Bliley Act) gives the authority to eight federal agencies to administer and enforce the Financial Privacy Rule and the Safeguards Rule.
- FERPA Compliance (Family Educational Rights and Privacy Act) was enacted in August of 1974 to protect student education records and pertains to any school, either K-12 or higher education, public, or private, that receives funds under any program from the U.S. Department of Education.
- SOX Compliance (Sarbanes-Oxley Act) is a United States federal law passed in response to a number of major corporate and accounting scandals which resulted in a decline of public trust in accounting and reporting practices.

[45]

### 3.5.1 Historical Analysis vs Real-time Analysis

The advancement of SIEM system technology has lead to the requirement of both real time and historical analysis. Both are important because they are searching for different types of patterns and sequences of events, as previously stated, the time period factor is important to include, since not all related events happen in close proximity. As the need for both historical

and real time analysis grows, so does the actual SIEM system, because it's capabilities are being tested more and more. There are often separate needs to ensure effective real time analysis and historical analysis.

Real time analysis has always been an essential functionality of a SIEM. Using the previously spoken about rule-based correlation, SIEM systems are expected to excel at analysing streams of data being fed into the system through out the day. This means that the storage of data is not necessarily that important, since it is trying to analyse current event data streaming in. Since so much data is streaming in, it is essential that the data is normalized to allow for fast correlation. The common performance bottle neck here would then be the CPU processing, since storage is not an essential.

As the suspicious activity on the network is discovered, it is required that the SIEM immediately alerts the appropriate personnel. The real time alert allows them to handle the problem as soon as possible. Real time analysis and reporting ensures that known attacks, and attack patterns, are being found and stopped before too much important information can be stolen or leveraged.

Historical analysis has its own set of needs entirely, and that is why deciding which analysis is the focus is important : unless a very expensive alternative is possible. Historical analysis is on the way up in terms of necessity, because historical analysis focuses on discovering attacks as opposed to finding known ones. This is referred to as hidden/ persistent/ advanced threat discovery. A recent research showed a 36% increase in APTs (advanced persistent threats) against organisations. This means this kind of attack is on the increase, and systems need to be protected from it [46].

Historical analysis generally requires a lot of storage space because it aims to store lots of unstructured logs. These logs are carefully analyzed using modeling methods and interactive exploration. Using historical analysis, an administrator can learn about network data patterns, how the patterns looked during normal usage, which would help detect unusual activity from certain network devices. This would then allow the network administrator to keep an eye on a certain part of the network, watching for any suspicious activity. The common performance bottleneck for historical analysis is storage capacity and input/output of that storage [47].

## 3.6 Alerting

The final important process that a SIEM contributes to the security of a monitored environment or network, is its ability to alert the necessary personnel about its findings. A SIEM's ability to alert is a core purpose because it is constantly checking and correlating thousands of sources, looking for anything suspicious. Alerting comes in two types: active and passive [48].

Passive alerts are seen as normal reporting where the SIEM presents a dashboard to give the SecOps a holistic view of all the devices and information coming into the SIEM. The

SecOps can then check the dashboard for points of interest and hence spot suspicious behaviour. Passive alerts are often a dashboard full of automated analysis of correlated events. These dashboards help SecOps spot activity that is non standard in pattern [49].

Active alerts is the SIEM's way of notifying SecOps that a high impact correlation has been made and that immediate attention is required. This would usually involve the SIEM in sending a message in the form of an email to the SecOps with detailed information about the suspicious behaviour. Active alerts make near real time intrusion detection possible because the SecOps is notified straight away that someone is doing something they should not. The alerts also give the SecOps information that would help them stop the malicious intruder if the alert is deemed to be correct.

# Chapter 4

## The Nature of Contextual Data

### 4.1 Introduction

The exponentially increasing amount of contextual data collected, mined and shared means that there is a large volume of useful information available to any person or system regarding its environment. [50]. Context is very important when considering just how much information it can provide about a place, person or situation. It is any information used to characterise the situation of an entity [51].

Situational awareness, or context, is used in everyday conversation between humans. Fortunately, this is due to factors including, but not limited to, the knowledge of a shared language, the common understanding of how the world works and the nature of everyday situations. This ability to gain context in conversation simply by being part of it, is not shared with computers. Computers can only gain knowledge or information about a situation if it is given the information explicitly. [52].

Contextual data is being used by numerous sectors of the business world. For example, technology and marketing companies use contextual data to more accurately accommodate their customers, based on their customer's historical and current contextual data. In a paper by [53] information retrieval functions and browsing tasks are common uses for contextual information. In information retrieval, contextual information is important because each information retrieval process happens in a particular environment linked with its own information and hence is tied to the specificity of that environment. In relation to browsing tasks, the same paper states that the availability of contextual information made browsing tasks more productive.

As technology grows and networks become ever more ubiquitous, information about the environment becomes more important. The growth of mobile devices and increasing speeds of mobile networks means that contextual information can be used by more devices and produced by more devices. To take advantage of the potential information and use cases accompanied by it, "context awareness, broadly defined as the ability to provide services with full awareness of the current execution environment, is widely recognized as one of the

cornerstones to building modern mobile and ubiquitous systems” [54].

Hence there are solutions aimed at managing contextual information and delivering it effectively to devices that require it. These solutions, known as context data distribution, which is the ability to collect and deliver relevant context data are important in context-aware research. It has been found to be important because context data can often mean that a service must adapt to its execution environment. As you can see context information is not a new idea and it is being used in many applications already.

## 4.2 Challenges of Context

There are many challenges when developing a system that can use context effectively. As the system is given more information, the system needs to know how to use the information in a manner that is useful. The goal is to create systems that understand both simple and complex situational information. One such challenge is building a greater understanding between the complex relation of context, human activity and human behaviour [55]. There are many theories relating to this relation, but they are beyond the scope of this thesis.

In general, it could be thought that adding a deeper context awareness requires understanding a theory about the complex relationship between context, human activity and human behaviour, but in actual fact the challenge is greater than that. In practice, developers would take complex problems and create simplified models that relate to each other in such a way that they represent the complex original. However, this assumes that we live in a stable world unproblematically recognised by all. This is found by [56]. The problem stems from two conflicting philosophical theories: positivism and phenomenology.

Positivism is a scientific and empirical tradition from which Computer Science derives while phenomenology is the background behind explaining context in complex human activity and behaviour. The incompatibility of these two standpoints explains why the limitation in developing a deeper context aware system arises [55]. Phenomenology explains context as an interaction problem and in this approach, context can only be understood as it arises. This alleviates the need to develop underlying models that would describe the objective reality behind context. Hence we would need machines to have human-like cognitive skills. There would be a need to extract the mathematical model of the brain and imitate it in a computer [57].

Since this thesis is not aiming to argue the feasibility of what is required according to the above theories, what is important to understand is that context data in relation to machines is of a radically different nature to that of humans. Positivism looks at context as a representational problem. This means we can take the concept of determining why a situation happens and use this to determine the action the computer would take [52]. This makes the concept of context awareness operational, since we can use actors and information sources, but nevertheless, since this stems from a positivist view, we are still limited by the 'what' that context developers can foresee, and their ability to precode the appropriate action or

response [58].

With these concerns in mind, we look at the steps context-rich applications take in being designed and developed.

## 4.3 Building Contextual Data Applications

When deciding to build or develop a program that will use contextual data, there are a few factors that must be taken into account. First, a quick overview and discussion of different context models. These are important to consider when deciding on the type of context-data you want to retrieve. The next section will look at context-type categorisation and finally context-aware system categorisation. By looking at these section, we will develop a deeper knowledge of context data and how to use it.

### 4.3.1 Context Models

“Context model identifies a concrete subset of the context that is realistically attainable from sensors, applications and users, and able to be exploited in the execution of the task.”[59] In our case, the exploitation of the context in the execution of a task will be the use of various online APIs in the execution of OSSIM’s correlation engine’s detection capabilities.

This model is explicitly specified by the developer to be used in an application that makes use of the context. However, the model often evolves over time, as the need for more context or different context presents itself. According to [60], there are three different approaches to context models:

- **No application-level context model** - this is when the application does all work involved such as preprocessing, context acquisition, storing and reasoning.
- **Implicit context model** - this is the use of frameworks and libraries to do context data acquisition, preprocessing, storing and reasoning tasks. This approach provides a standard design to follow which allows quick integration into applications but it also means that the context data is hard bound to the application.
- **Explicit context model** -this is the use of a context management infrastructure or middleware solution. This means the usual tasks such as context acquisition, preprocessing, storing and reasoning are independent of the application. It allows the application and the context to be clearly separated and hence independently developed further.

The implicit context model will be used for this thesis solution. The reason for this choice is the fact that the SIEM system, OSSIM, is an independently developed and available SIEM which allows for external data feed integration. We will develop applications that will handle context acquisition, preprocessing, storing (if necessary) while OSSIM will handle the reasoning tasks. These reasoning tasks will be what to do with the context data that is now

available to OSSIM. Each context application will have a standard log file design relevant to the data it is retrieving.

### 4.3.2 Context Type Categorisation

In order to look at the different categorisation schemes researched, context types must first be discussed. According to [61], context can be divided into primary and secondary context types.

- **Primary Context** - this is context that is retrieved directly from the source and not inferred by using existing available context. Generally, primary context is either location, identity, time or an activity.
- **Secondary Context** - this is context that is computed using the primary context. An example would be predicting a user's activity based on the user's calendar. One context is determined using another context's information.

The problem with this classification of context type is that if you do not know how the context was found, you have no way to identify whether the context type is primary or secondary. For example, getting the location of a person via their device's location services would be a primary context, but getting the location of a person by taking the context that they are with someone, and that this someone is at a certain location, would be secondary.

Categories of Context (Operational Perspective)		
	Primary	Secondary
Categories of Context (Conceptual Perspective)	Location	<div>Location data from GPS sensor (e.g. longitude and latitude)</div> <div>Distance of two sensors computed using GPS values</div> <div>Image of a map retrieved from map service provider</div>
	Identity	<div>Identify user based on RFID tag</div> <div>Retrieve friend list from users Facebook profile</div> <div>Identify a face of a person using facial recognition system</div>
	Time	<div>Read time from a clock</div> <div>Calculate the season based on the weather information</div> <div>Predict the time based on the current activity and calendar</div>
	Activity	<div>Identify opening door activity from a door sensor</div> <div>Predict the user activity based on the user calendar</div> <div>Find the user activity based on mobile phone sensors such as GPS, gyroscope, accelerometer</div>

**Figure 4.1:** Context type categorisation using two different perspectives [62]

In Figure 4.1, we can see the classifications based on different perspectives - operational and conceptual. The conceptual perspective has already been discussed, because it is the conceptual perspective to say that certain context-types are primary, and that secondary context-types come from these primary context-types. The operational perspective of categorising context is based on how the data was acquired.

The operational perspective promotes understanding in the data acquisition techniques -both the challenges and advantages. However the conceptual perspective promotes the understanding of the relationship between contexts. We must then use perspective to model context more accurately. With this in mind, quality, validity, accuracy, cost and effort of context-data acquisition can vary based on the acquisition technique, so using different perspectives to decide the best route forward can be helpful.

### 4.3.3 Categorisation of Context Aware Systems

The categorisation of systems that use contextual data is the final look at previous research of context-aware systems. This section will bring to light the interaction of such systems with the user, while also looking at the features that these systems might employ. Whilst doing this, we must keep in the mind the previously discussed limitations and categorisations involved in developing and using contextual data.

When looking at the level of interaction between a context-aware system and the user, we must consider the user needs and the level of autonomy we want the system to have. Originally, context-aware systems were intended to use context-data to make decisions without user interference, [55] but as we discussed earlier, humans make better judgements based on given context. Hence we must find a balance between user interaction and context-data driven autonomous decision making. In a paper by [63], the aim is to reduce user intervention, ease system's use and decrease user distraction.

The interaction between a user and the context aware system is classified in [64], as one of these three:

- Personalisation - this is when the user will manually set the preferences, likes and expectations to the system.
- Passive Context Awareness - this is when the system will monitor contextual data feeds and give the user options relating to actions the system should take based on the contextual data.
- Active Context Awareness - this is when the system constantly monitors the contextual data feed and takes action on its own i.e. autonomously.

In our case, the interaction will be that of active context awareness because the contextual data applications and OSSIM will constantly be monitoring the environment and evaluating the context along with the network activity data to make a decision. OSSIM will then decide whether to alert the administrator and raise a ticket or not.

## 4.4 Summary

In summary, context is a complicated and broad topic. It can be perceived differently when using it for different reasons. We have seen that by simply specifying the use of the context data in an application, the technique in which we retrieve it, use it, process it and reason with it changes. Context data is difficult to define as a single entity and hence it makes it hard to create a concrete model that encompasses all context.

In an attempt to model context data, different challenges are raised depending on the different approaches and hence it is important to clearly define the type of contextual data and what the use case for that data is. With a clear definition, a model can carefully be constructed. Depending on the use case of the contextual data, the categorisation of how the system will use the contextual data changes too. This, in turn, also affects the type of contextual data that is required. The final point is to understand the interaction of the user, the system and the contextual data. This will help decide what the system does with the contextual data and how it will affect the interaction of the system and the user.

# Chapter 5

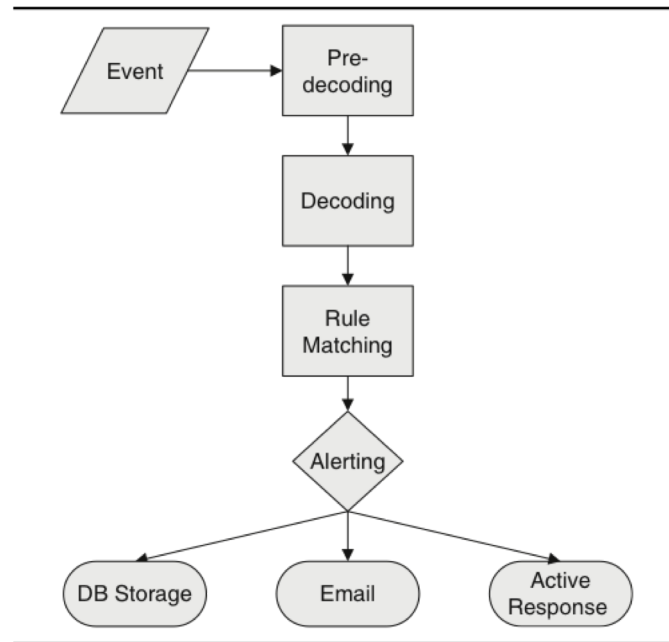
## Design of Testing Environment and Data Streams

This chapter documents the creation of an acceptable environment with appropriate testing conditions in which the hypothesis can be evaluated. In Chapter 4, we defined four different types of contextual data that would be useful to SIEM and we motivated why these would be useful. In this chapter, these four types of contextual data will be assessed further for implementation feasibility, so that we can test that these types of data are in fact useful.

### 5.1 General Design

The general design section refers to parts of the design that will remain consistent throughout the use of different contextual data. This will ensure that the difference in observations found throughout the evaluation section of this thesis are dependent on the type of contextual data, rather than the actual system, therefore making the type of contextual data the independent variable.

Figure 5.1 shows the process of getting from the event data, through the SIEM system, to finally alerting the appropriate personnel. This process is followed by all the event data - regular security event data as well as our contextual feed event data. The following sections will meticulously go through the processes.



**Figure 5.1:** Flow chart showing the general event data life cycle

### 5.1.1 The System: OSSIM

As already discussed in Chapter 2, the security information and event management system that will be used is AlienVault's open source SIEM solution known as OSSIM. Although OSSIM comes standard with many decoders and rules, each of these will be customized specifically for each type of contextual data. All contextual data will, however, be fed into the OSSIM system via the Rsyslog facility. This is done because it allows easy monitoring of a log file over a network and ensures secure communication between OSSIM and the system generating the log data. Using the Rsyslog facility also allows completely new detector plugins to be created for each different kind of log data coming through.

Another approach would be to use an agent such as OSSEC. An OSSEC agent can be installed on systems across the network and enabled to monitor sensitive system files. The positive point of using an agent such as OSSEC would be that the agent does some preprocessing of log data. It then sends through the preprocessed log data to ensure the OSSIM system can get the most information and use from the log data. The reason this approach has not been used is because it adds unnecessary complexity with regards to creating custom log data. Since the events are coming through the OSSEC agent, new rules would need to be added to the existing OSSEC agent detector plugin for each new type of log data that the OSSEC agent might receive.

These log files will be forwarded, as they change, to the OSSIM system. Keeping this

design of the implementation uniform is imperative because it will allow us to observe important points regarding the contextual data such as:

- In what log form is contextual data most useful?
- Should the contextual data be optimized more before it is sent to the SIEM? I.e pre-processing.
- What information should be included with the contextual data?

Once the logs reach the OSSIM system, the system will need to decide how to handle these logs. By means of a detector plugin enabled via the OSSIM agent, the OSSIM system will recognize the program from which the logs are originating. For each different type of contextual data application, a custom detector plugin will need to be created and tested to ensure that the OSSIM system is reading in the logs as the appropriate log data. This part of the process is pivotal because if the OSSIM system can not decide which logs come from which programs, the rules that the OSSIM system runs against the logs will be incorrect. This would make the custom generated logs useless.

The log files will be made unique to the application that generated them. Each log line generated will have the name of the contextual data application appended to the beginning of it. Then using regex, and some plugin rules, the OSSIM system can quickly detect which contextual data application generated the log line.

### 5.1.2 Rsyslog

### 5.1.3 The Network

The OSSIM system will be run on the same network for all the different types of contextual data. This is done to ensure, again, that the only independent variable being tested is that of the contextual data. Since we will be using the same system with rsyslog as well as the same OSSIM system, the network settings will be identical for each case. It is also important that the network has an internet connection because the context data applications will be gathering data from the internet.

### 5.1.4 Application Development Environment

All the contextual data will need to be fetched from the data sources discussed in Chapter 4. Small programs will be developed to ensure that the contextual data is being fed into log files that are being monitored by the rsyslog facility. These small applications will be programmed in the integrated development environment (IDE) called Eclipse using the programming language, Java. Table 5.1 displays some of the details:

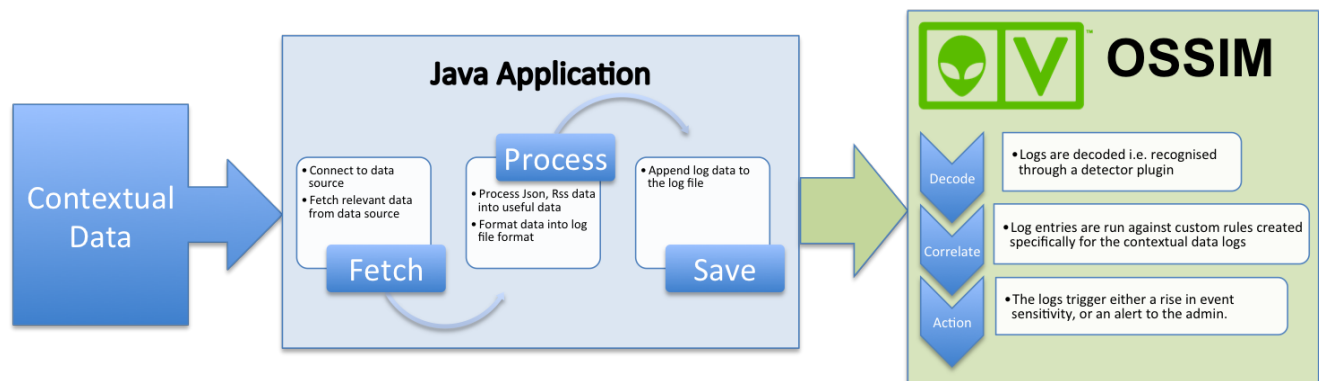
<b>IDE Name:</b>	Eclipse
Version	Kepler Service Release 2
Website	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
<b>Programming Language:</b>	Java
Version	Java SE 7
Website	<a href="http://www.oracle.com/technetwork/java/index.html">www.oracle.com/technetwork/java/index.html</a>

**Table 5.1:** Table showing the application development environment

### 5.1.5 Data Formats

### 5.1.6 General Design Summary

The Figure 5.2 describes the different general processes that each contextual data application will have to go through along with the outline of the main processes required to make the contextual data useful to the OSSIM system.



**Figure 5.2:** Flow chart outlining the different general process of each contextual data application

The flow chart starts with the contextual data. At this stage, the contextual data is available but not yet useful in a security context. The contextual data is fetched, processed and saved through the processes that each contextual data Java application program will complete. Since not all the contextual data is available in the same format, each contextual data will have its own Java application processing the data.

Once the Java application has written the formatted data to a log file, OSSIM takes the log file and decodes the data into objects that are useful for the correlation engine to use. The correlation engine will determine the course of action by running the logs against custom defined rules. Each contextual data will have its own rules specific to its use case.

## 5.2 Context Data Retrieval Design

The context-specific design section refers to each type of contextual data individually, taking into account the considerations mentioned in Chapter 4. Since each type of contextual data is different, the design for each type of contextual data will be different. This is necessary because we want to design how each type of contextual data is used, to maximize its potential, rather than trying to keep the implementation of each type of contextual data uniform.

There are four different types of contextual data that we will be implementing, each in its own way. For each type we will ask the same questions when deciding on the implementation strategy:

- How do we acquire this contextual information?
- How do I log this information in a useful form?
- How can I adapt existing rules to incorporate this information?

Once we have implemented the contextual data and ensured that the OSSIM system is receiving the different types of contextual data, the way in which the OSSIM system handles the data needs to be configured. A design of rules for each type of contextual data will be the next step. This step is the most important, because the rules are linked to each type of contextual data and will tell the OSSIM system how to handle the log data being fed in. The process of optimizing these rules will definitely be iterative to try to ensure that the addition of this contextual data to the OSSIM security event data feed is useful.

Each of the following contextual data sections will need to go systematically through the same general steps, to ensure that everything has been covered regarding that type of contextual data. Table 5.2 is a checklist of procedures necessary, for each contextual data type to function from start to finish:

Step	Description
1	Get required data from source (online, rss feed, json, RESTful)
2	Process into logs
3	Feed logs successfully into OSSIM (through remote OSSEC Agent)
4	Write decoder for OSSIM to recognize logs
5	Write rules for each type of logs
6	Check that Alerts are functioning

**Table 5.2:** Checklist of procedures for each contextual data type

### 5.2.1 Context: Social Media Information

Social Media information is so widely available because of the many different social media companies such as Facebook, Twitter, Bebo, MySpace and Instagram. Each of these social media companies offer much of the same information but also diverge just enough for each have its own purpose. Understanding the purpose of each social media feed is important, because it allows us to filter the data so that it is useful to us. In this case, Instagram probably won't be too useful because the data logs that are going to be fed into the SIEM cannot contain images.

In this case, filtered results trending based on the hashtag naming convention, this would give us more valuable information. Hence, each social media company, depending on how results are filtered, is useful to us. Table 5.3 is a summary of the API that will be used to gain and aggregate social media information.

<b><u>API Name:</u></b>	<b>Social Mention API</b>
Website:	<a href="https://code.google.com/p/socialmention-api/">https://code.google.com/p/socialmention-api/</a>
Description:	Social Mention is a social media search and analysis platform. The website allows for the aggregation of user generated content across different social media platforms into a single stream of information. It allows a user to easily track and possibly measure what social media information is being posted about a specific person or company. This service is provided in real-time to developers through the Social Mention API which allows developers to interact with the Social Mention website programmatically. The API is freely available for users that will use it personally and non-commercially. It allows for a maximum of 100 queries per a day.
Terms and Conditions	<a href="https://code.google.com/p/socialmentionapi/wiki/APITermsOfUse">https://code.google.com/p/socialmentionapi/wiki/APITermsOfUse</a>

**Table 5.3:** Table of Social Mention Information

The next step in using the social media context information is developing an application that uses the above API. This Java-based application will need to connect to the Internet to make calls to the API. In order for this to happen, the following steps need to be taken to setup the connection:

- First an `URLConnection` is created and a request to the Social Mention is made. This url will have all the necessary REST parameters amended to the Social Mention base url.
- The request method is set to “GET” since the Social Mention API allows developers to interact via simple HTTP requests.
- The response code is received through the connection that was made. This response code is checked. If the “HTTP\_OK” response is received, then the request to the Social Mention API was successful and we can continue with dealing with the response data.

Table 5.4 is a break down of the Social Mention API call syntax regarding the calls made to its RESTful system.

<b>Base Url Prefix:</b>	<b>http://socialmention.com/search</b>
<b>REST parameters</b>	<b>Description</b>
q	The url encoded query term. This is the main part of the call because it specifies the term that the API should search for on social media.
f	The response type. This parameter is used to specify the type of data your application will receive in response to this query. Available options are json, php, xml, rss and csv.
t[]	The search type. This also allows for multiple search types at once. Available options are blogs, microblogs, bookmarks, comments, events, images, news, videos, audio, questions, networks and all.
src[]	Specific social media sources. This is an optional parameter that allows the user to explicitly state which social media platform to search through.
l	Specify location. This optional parameter allows the user to filter results by location.
<b>Example</b>	http://socialmention.com/search?q=cyber+security&f=json&t[]=blogs&t[]=comments&src[]=twitter &lang=en

**Table 5.4:** Table of Documenting Social Mention's Rest request parameters

The response we expect from the calls to this API will be in a JSON format because we will have specified it to be that response type. The response will contain a number of fields of data relating to the query that we made to the Social Mention API. To better understand what kind of response data will be returned, Table 5.5 explains each field that can be expected in a typical response. These are known as the response properties.

<u>Field Name:</u>	<u>Description</u>
title	Title of the search
timestamp	time of the search
count	Number of items returned from the search
...	<b>The start of the results as an array. Each array item will have the following properties</b>
id	Unique hash id based on URL
title	Title of item
description	Description of item
timestamp	Time of item being posted/published
language	Language code of item
user	Authors username
source	Source's name
type	Mention type i.e. blog

**Table 5.5:** Table of Documenting Social Mention's Rest response

Once we have received the JSON response, the application will parse the data into a usable form. The process of parsing the JSON response data will go as follows.

- The JSON data will be run through the Java provided JSON Parser.
- The JSON Parser creates a JSON Element object out of the parsed data. This step is done because a JSON Element allows for the data to be manipulated into a JSON Array.
- The JSON Element object is changed into a JSON Array because this JSON array stores each event matching the query as an object in the JSON Array.
- Next we use a Google class called Gson. Gson is Google's Java library that provides functionality to covert JSON strings into Java objects and visa versa. Gson also works with arbitrary Java objects. We create a Gson object and parse the JSON array into our SocialMentionItem array.

- Now we have all the information from the Social Mention search in a readable form with each element of the JSON response correctly parsed into its own objects for easy usability.

The next step will be adding this information into a log file that OSSIM can interpret. These steps will be discussed in 5.3 - Log File Design.

### Use Case of Social Media

Since the Social Mention API has the potential to return thousands of results, the use case of this type of contextual data needs to be very specific and limited. To limit the results, the query term will need to be very specific and link to the action that we will take with regards to OSSIM. The query will search for a term such as 'Brute Force Attack' or 'Denial of Service'. We can check the response data for the number of currently trending topics involving these terms. If the number of returned results is extremely high, then we heighten OSSIM's detection sensitivity to that kind of attack. This will mean that if denial of service attacks are in the news or social media, our system will be more aware that it should be more sensitive towards characteristics common to denial of service attacks.

#### 5.2.2 Context: Meteorological Information

Meteorological information is also readily available, but it is important to determine what information is useful. Requesting meteorological information in general can yield many results, most of which would not be useful. As discussed in Chapter 4, the location of the weather we are requesting is a great way to filter some of results. Unfortunately not all weather or meteorological information is necessary, or useful i.e. does the SIEM need to know that it is sunny outside. Hence, extreme meteorological will be the only information written to the log files since it is the only useful information.

Although there are many meteorological information offerings on the Internet, the OpenWeatherMap API was chosen because it is free and it offers an easy to use API with community support. Table 5.6 contains the basic information about OpenWeatherMap.

<b>API Name:</b>	OpenWeatherMap
Website	<a href="http://openweathermap.org/api">http://openweathermap.org/api</a>
Description	OpenWeatherMap is a service that provides weather data for around 200,000 cities and any geo location that is available on the website. This service provides a wide variety of weather data such as current weather, weather forecasts, precipitations, wind data, cloud cover data, maps and analytics. The weather data is gathered from over 40,000 weather stations located around the world. All this data is available freely to developers through an API. There is a registration requirement, which attaches a developers unique API key to the developer to restrict API calls to 1200 API calls per min.
Terms and Conditions	<a href="http://openweathermap.org/terms">http://openweathermap.org/terms</a>

**Table 5.6:** OpenWeatherMap Information

The next step is the development of a Java-based application that will implement the above OpenWeatherMap API. Luckily there is an open source library developed especially for this API, which enables developers to make API calls relatively easily. The process of connecting to the OpenWeatherMap API service is as follows:

- Register an account on the OpenWeatherMap.org website in order to receive a unique API key.
- Download the open source java library for OpenWeatherMap.org Weather APIs from <http://code.aksingh.net/owm-japis/downloads>. (This library is known as OWM JAPIs - OpenWeatherMap Java APIs)
- Create an OpenWeatherMap object. This object's constructor requires the desired temperature measurement and unique API key as its parameters. In our case, we will specify the metric measurement of Celsius and add the API key linked to my account.

A noteworthy comment is that the application must be connected to the Internet to make any sort of connection to the OpenWeatherMap API service. Once we are connected to the API we can start making some requests for meteorological information.

Figure 5.3 give an overview of the OpenWeatherMap API calls available to developers. Table 5.6 shows that API allows developers to search for current weather data for one location using a variety of different methods. Please note that the data response type can be chosen too, but the default response type is JSON.

Call Current Weather Data for One Location	
Method:	Example:
City Name	api.openweathermap.org/data/2.5/weather?q=London
City Name and Country Code	api.openweathermap.org/data/2.5/weather?q=London,uk
City ID	api.openweathermap.org/data/2.5/weather?id=2172797
Geographic Coordinates	api.openweathermap.org/data/2.5/weather?lat=35&lon=139
Zip Code	api.openweathermap.org/data/2.5/weather?zip=94040,us

**Figure 5.3:** Current weather api call examples

These are examples of the calls that we make to the OpenWeatherMap API. Using the method of creating a `HttpURLConnection` and following the same procedures as the social media context data application, the above calls would be sufficient to get a response from the OpenWeatherMap API service. Luckily, the library we are using, OWN JAPIs, has preset methods that build the query to the OpenWeatherMap API service automatically. All that is required in order to obtain the weather data of any city or location, are the co-ordinates of the location, the name of the city, city ID or ZIP code.

The response data is in a JSON format. Table 5.7 displays most of the response data returned by a single location weather data request.

<u>Parameter:</u>	<u>Description:</u>
id	City identification
dt	Data receiving time, unix time, GMT
name	City name
<b>main</b>	Name of dataset to follow which contains main weather conditions
temp	Temperature in Kelvin (OWM JAPIs converts it to Celcius)
humidity	Humidity in percentage
<b>weather</b>	Name of dataset to follow which contains general weather info
id	Weather condition Id
main	Group of weather parameters (Snow, Extreme, Rain, etc)
description	Weather condition within the group
icon	Weather icon Id

**Table 5.7:** OpenWeatherMap's Rest Response Parameters

Once again, by following similar procedures followed for the social media context data application, the JSON format could be parsed into a more useful java object, but the OWM JAPIs automatically takes the response and stores it in a CurrentWeather object. This CurrentWeather object offers a magnitude of 'getter' methods that return different weather data stored in the CurrentWeather object. Table 5.7 is useful to note because we can see exactly what data is available from a single call to the OpenWeatherMap API.

Table 5.8 briefly shows the methods implemented to return the response data that we would need for our log files.

<b>Parameter Returned:</b>	<b>OWM JAPIs Method:</b>
<b>coord</b>	getCoordInstance()
latitude	getCoordInstance().getLatitude()
longitude	getCoordInstance().getLongitude()
<b>main</b>	getMainInstance()
temp	getMainInstance().getTemperature()
humidity	getMainInstance().getHumidity()
<b>wind</b>	getWindInstance()
speed	getWindInstance().getWindSpeed();
<b>weather</b>	getWeatherInstance(index i)
id	getWeatherInstance(index i).getWeatherCode()
description	getWeatherInstance(index i).getWeatherDescription()
icon	getWeatherInstance(index i).getWeatherIconName()

**Table 5.8:** OpenWeatherMap Library methods

Table 5.8 shows that most of the response data is easily acquired using the OpenWeatherMap Java APIs. Keeping to the steps in table 5.2a, the next step is to process this data into logs. This step is covered in the next section 5.3 - Log File Design.

### Use Case of Meteorological Information

Once the meteorological data is available, it needs to be implemented in such a way that it is useful to the OSSIM system. In our case, the use of the OpenWeatherMap API allows us to leverage location-specific weather information. Hence, we will extract the weather code from the response data and match it with the corresponding weather condition.

The weather codes that relate to extreme weather conditions will be linked to the OSSIM system as a condition to heighten the sensitivity of OSSIM's correlation engine with regards to login security events. The thinking behind this use case, is that during an extreme weather condition, minimal or zero employees should be trying to enter the monitored

system, because there is not likely to be a working day during extreme weather.

Another use case would be trying to help the OSSIM decide whether the monitored system is under a denial of service attack. If the weather condition is extreme, then it is logical to think that more users than normal may be trying to check the weather on a system on the network. A sudden increase in requests for a network resource can trigger a false positive linked with a denial of service attack. By leveraging the OpenWeatherMap API, we can lower OSSIM's detection sensitivity regarding service request security events because it will receive a log confirming extreme conditions and possible increased service requests.

### 5.2.3 Context: Calendar Events

Calendar events can be a vast and useful contextual data source because it gives us insight into what someone, or some company, or some country even, should be doing on this day. Public holidays, although different for different countries, allows us to see why the population at work might be low, or why the internet usage might be low, or why remote logins have increased.

Since there are many countries, for this kind of data to be useful, a location would need to be specified. This actually works out well for us because the way in which we get public holiday information requires that we specify a country. Since there are not many public holidays, filtering this information is not going to be an issue (although specifying a country works as a kind of filter anyway).

There are many online services that have public holiday event data available in one way or another. For this public holiday contextual data application, the Google Calendar API will be employed because there is online support and it can be easily implemented. Figure 5.4 is a brief overview of the API:

<b><u>API Name:</u></b>	Google Calendar API
<b>Website:</b>	<a href="https://developers.google.com/google-apps/calendar/">https://developers.google.com/google-apps/calendar/</a>
<b>Description:</b>	<p>Google Calendar API allows developers to create applications that can create, modify, delete events and search of events. The API allows developers to use a RESTful calling style and some client libraries for multiple programming languages. Via the Google Calendar web interface, a program can execute many operations including viewing public calendar events.</p> <p>This API does require that the developer acquire an API key by creating a project. This service is free up to making a certain amount of call a day.</p>
<b>Terms and Conditions:</b>	<a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a>

**Figure 5.4:** Google Calendar API information

The next step is developing an actual Java-based application that has this Google Calendar API implemented. Unfortunately, implementing this API is not as straightforward as our previous one. To start we need to set up an authorized Calendar API client service. The process of setting this up is as follows:

- Sign up for a Google Account and create a project on <https://console.developers.google.com>.
- Enable the Google Calendar API.
- Create a new Client ID and download client secret.
- Load client secret and authorize user (both programmatically).

Now that our application is connected to the Google Calendar API, we can work on making some useful calls. Again it must be said that the application must be connected to the Internet to make these RESTful calls to the Google Calendar API service. The next consideration is the response data. Once the application is connected to the API service, there are a few calls we can make to the API. Table 5.9 displays what kind of methods we can call on, for the live data to filter our response.

<u>Method</u>	<u>Description</u>
<code>list(String s)</code>	This is the most important method because it tells the Google Calendar service exactly what events you want to get from the service. This method determines whether the user's primary calendar is returned, or a country's public holiday listing. There are many more calendar instances that can be returned.
<code>setMaxAttendees(int i)</code>	Max number of attendees to include in the response. If there are more than the specified number of attendees, only the participant is returned. (Optional)
<code>setMaxResults(int i)</code>	Max number of results returned on one result page. The default is 250 events and maximum is 2500 events.
<code>setOrderBy(String s)</code>	Order of the events returned in our result. (Optional)
<code>setQ(String q)</code>	Free text search to find events that match these terms in any field. (Optional).
<code>setSingleEvents(boolean b)</code>	Decides whether recurring events should be expanded into instances, and only return single one-off events and the instances of the recurring events. (Optional).
<code>setTimeMin(DateTime d)</code>	Inclusive lower bound for an event's end time to filter by. Default is not to filter by end time. (Optional).
<code>setTimeMax(DateTime d)</code>	Exclusive upper bound for an event's start time to filter by. Default is not to filter by start time. (Optional).
<code>setTimeZone(String s)</code>	Time zone used in the response. Default is the time zone of the calendar used.

**Table 5.9:** Table of methods available for Google Calendar API client library

Once the response data has been returned, it is stored in an Event object. The Event object, also part of the Google Calendar API client library, allows us to get individual events into a Java list of single Event objects. This Event list can now be passed through to our log file writing method. This next process will be discussed in 5.3 - Log File Design.

### Use Case of Calendar Information

Once the calendar information is available, the way in which we use it will be important. The use-case behind the calendar information that we have available to us, is to list all the public holiday events for the location of the OSSIM system. This public holiday events will be checked against the current date. If the current date and a public holiday date match, then we can heighten OSSIM's correlation engine sensitivity with regards to login security events. The reasoning here is similar to that of the meteorological use case - it is less likely that employees would be trying to log into the network on days that they are not required to log in.

#### 5.2.4 Context: Terror Threat Level

The terror threat level of a country can be an important contextual data to take notice of. Cyber terrorism has become a massive problem everywhere and hence it is important to be ready for anything. By monitoring the threat level of a country, we can brace our systems for a cyber attack. If the attack is imminent, we can heighten security sensitivity. The reasons for monitoring this contextual data have already been covered in full in Chapter 4 The nature of Contextual Data.

For the purpose of our testing, we can use the British Government's MI5 website to retrieve the state of the threat level. The British Government provides the threat level as a RSS feed which makes it openly available. Figure 5.5 is a brief overview of our threat level source.

<b><u>Data Source Name:</u></b>	Threat Level RSS
Website:	<a href="https://www.mi5.gov.uk/home/about-us/what-we-do/threats/terrorism/threat-levels/threat-level-rss.html">https://www.mi5.gov.uk/home/about-us/what-we-do/threats/terrorism/threat-levels/threat-level-rss.html</a>
Description:	<p>The Security Service of the British Government provides the current UK Threat level in a Really Simple Syndication (RSS) feed. This enables anyone to get the current UK Threat level without having to access the website everytime. This also means that any updates to the UK Threat level will be automatically relayed through the RSS feed.</p> <p>This RSS feed is openly available for anyone wishing to subscribe to it.</p>
Terms and Conditions:	<a href="https://www.mi5.gov.uk/home/bottom-nav-items-holding/terms-and-conditions.html">https://www.mi5.gov.uk/home/bottom-nav-items-holding/terms-and-conditions.html</a>

**Figure 5.5:** Threat Level Information

Since we have our source of context data, we can proceed to the implementation of the Java application. This Java application will have to feature a simple RSS feed reader and parser so that we can convert the information received in the RSS feed to some useful Java objects. The Java library provides developers with a couple of useful classes when attempting to parse an RSS feed. In order to parse the RSS feed XML, the following steps will be taken:

- Setup a connection to the RSS feed URL,  
*https : //www.mi5.gov.uk/UKThreatLevel/UKThreatLevel.xml*  
using an input stream. By opening an input stream with a URL, you are requesting to open a connection to the URL. This works well for RSS feed reading.
- Once the RSS feed XML is streamed, the XML data will be fed into the an XML event reader, which is a top level interface, allowing developers to peek at the next event whilst returning configuration information too.
- Now the parsing of the XML begins. Each XML element is provisioned for, and as our application detects, through the XML event reader that there is a start-tag then the data following the start-tag is mapped to that tag's name until its subsequent end-tag is detected.
- This is repeated throughout the XML - mapping all the data to their appropriate tags.

We now have the response data from the Threat Level RSS feed in a more useful format to create our log file. Our RSS feed data can now be processed appropriately for it to be written to its log file. This is covered in the next section 5.3 - Log File Design.

### Use Case of Threat Level Information

Since threat level information is not very dynamic, the rule set for this type of contextual data will be less complicated. The RSS feed data will return the current threat level of the British government and we will send this threat level through to OSSIM. Each threat level will be linked to an amount that OSSIM's sensitivity will be raised. This will allow OSSIM to always be aware of the current threat level of the network it is monitoring, hence ensuring that the appropriate security measures are taken.

## 5.3 Log File Design

Once the response is received and parsed successfully, a log file needs to be updated or created. This is an important time to think about what data included in the log file would be most useful to the SIEM. However, this answer can only be answered after some implementation and testing and hence it will be discussed later in Chapter 6 Implementation.

### 5.3.1 Social Media Context

The log file will be generated automatically once all the response data has been processed into *SocialMentionItems*. Each item will be written to the log file on its own line after the total amount of response data is written first. The logs will follow this format:

```
SocialMentionContextCount : SocialMentionItem.getCount
SocialMentionContext: SocialMentionItem[0].getTitle() @ SocialMentionItem[0].getDomain()
@ SocialMentionItem[0].getSource() @ SocialMentionItem[0].getType()

SocialMentionContext: SocialMentionItem[1].getTitle() @ SocialMentionItem[1].getDomain()
@ SocialMentionItem[1].getSource() @ SocialMentionItem[1].getType()

...
SocialMentionContext: SocialMentionItem[n].getTitle() @ SocialMentionItem[n].getDomain()
@ SocialMentionItem[n].getSource() @ SocialMentionItem[n].getType()
```

where *SocialMentionItem.Data* is the item's property attached to the variable **.Data** stored in the *SocialMentionItems* array. On each line the item property data will come after the : character because this allows for the SIEM to easily, by means of regular expression, separate each item's properties, and assign it to its own internal variables. This is a good reason for adding the 'SocialMentionContext' tag to the beginning of each log line. It allows the OSSIM agent to quickly recognize which contextual-data application log is being fed into the SIEM.

Once the log files have been populated, the *rsyslog* will detect a change and forward the log file changes through to the OSSIM agent. This process is explained in section 5.1.2 above. At this point, the OSSIM system can start running the log file against the custom-defined rule set. The rule set design is discussed in the next section 5.4 - Rule Set Design.

The *SocialMentionContextCount* : *SocialMentionItem.getCount* value will determine the severity of the attack that is trending on social media. This value will mean the SIEM should determine whether to increase sensitivity towards that trending attack or not.

### 5.3.2 Meteorological Context

This log file will also wait for the response data to be parsed. Since weather log file will only have the weather for the current day, it will be much easier for the SIEM to process. Hence this application will append a couple of lines to the log file each day, notifying the SIEM the weather condition of the chosen location. The rule set will determine how the SIEM deals with the weather conditions that are being fed to it. The format of the meteorological information log file will be as follows:

```
WeatherStatusContext: CurrWeatherData.getCityName() @
CurrWeatherData.getWeatherInstance(0).getWeatherCode()
```

where *CurrWeatherData* is the current weather data object returned from the OpenWeatherMap API request. The other methods will return the location of the current weather state with a weather code. Figure 5.6 contains the codes of only the extreme weather.

Code:	Description:
900 , 781	tornado
901	tropical storm
902, 962	hurricane
903	cold
904	hot
905	windy
906	hail
602	heavy snow
511	freezing rain
504	extreme rain
959	severe gale
961	violent storm

**Figure 5.6:** Weather Codes

We need only notice the extreme weather codes because it is these weather conditions that our SIEM will need to take into careful consideration.

### 5.3.3 Calendar Events Context

The log file design for the Calendar Events contextual data feed will be in a similar format to the previous log file. Since all the public holiday events are being taken into consideration, all the public holiday events will be written their own log line. The format of the log file will be as follows:

*PublicHolidayContext: event.getSummary() @ event.getStart().getDate()*

where *event* is the variable returned with the public holiday data. From this *event* variable, we can extract the summary which is the name of the public holiday and we can extract the start date which is the date of the public holiday. Each public holiday date will be run through a script to determine whether the date in the log is the same as the current date.

### 5.3.4 Terror Threat Context

The log file for the terror threat context data will be generated every time the terror threat level is changed. This information is extracted through the RSS feed setup on the British Government's website. Since only one terror level can be active at once, the log file will only have one line. The format of this one line will be as follows:

*TerrorLevelContext: getTerrorLevel(feed.getMessages().get(0).getDescription())*

where *feed.getMessages().get(0).getDescription()* will return a string detailing the current state of the terror threat level. This string will be passed into a *getTerrorLevel()* method that will extract the threat level from the string and return the threat level only. This kind of log design will ensure that the OSSIM system can quickly identify the contextual data type and the data that comes along with it - in this case a string with the level of terror threat level. Figure 5.7 describes the different terror threat levels that the British Government can choose.

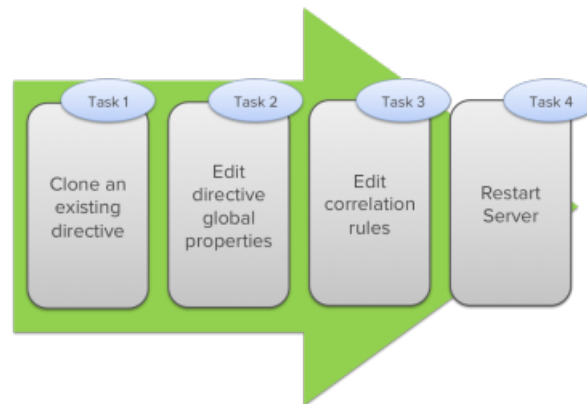


**Figure 5.7:** British Terror Threat Levels

[65]

## 5.4 Rule Set Design

This portion of the Design Chapter looks at how we will modify the current rule set, and create new rules that OSSIM will use when reading the contextual log file. In this section, careful consideration is given to not encumber OSSIM with rules that do not optimize its performance. The performance of a SIEM is vital to its success because of real time alerting. To ensure that the rules are not too intrusive, the rules will be copied from existing AlienVault rules and modified to use our custom contextual data feeds.



**Figure 5.8:** Flowchart of the steps taken to modify an existing directive [66]

Figure 5.8 shows the tasks necessary for defining your own directive. As mentioned before, you would clone an existing directive that contains a similar ruleset and hence has a similar goal. In our case that would be the AlienVault version of the bruteforce attack. The second task is to change the global properties to suit your needs. This involves changing the `name`, `id`, `priority` etc of the actual directive. The next task is to edit the rules appropriately by defining the necessary `plugin_id`, `plugin_sid`, `type` etc. These values would match up with our newly created contextual plugins. Lastly, the server is restarted in order to activate the new directive.

When considering how we would like to modify the rules to satisfy our means, we must consider the use-case meticulously. Each of the following sections will briefly overview the aim of the rule and an idea of how it may be implemented. Rules in OSSIM are structured in an XML format. Figure 5.9 is an example of a rule set for a brute force attack.

```

<?xml version='1.0' encoding='UTF-8' ?>

<directive id="50004" name="AV-FREE-FEED Bruteforce attack, login authentication attack against DST_IP"
priority="4">
  <rule type="detector" name="PAM UNIX authentication failure" reliability="1" occurrence="1" from="ANY"
to="ANY" port_from="ANY" port_to="ANY" plugin_id="4004" plugin_sid="2">
    <rules>
      <rule type="detector" name="PAM UNIX authentication failure" reliability="4" occurrence="20"
from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="600" port_to="ANY" plugin_id="4004"
plugin_sid="2">
        <rules>
          <rule type="detector" name="PAM UNIX authentication failure" reliability="4"
occurrence="50" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="1200"
port_to="ANY" plugin_id="4004" plugin_sid="2">
            <rules>
              <rule type="detector" name="PAM UNIX authentication failure" reliability="6"
occurrence="100" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="1200"
port_to="ANY" plugin_id="4004" plugin_sid="2">
                <rules>
                  <rule type="detector" name="PAM UNIX authentication failure"
reliability="10" occurrence="200" from="1:SRC_IP" to="1:DST_IP"
port_from="ANY" time_out="7200" port_to="ANY" plugin_id="4004"
plugin_sid="2">
                    <rules>
                      <rule type="detector" name="PAM UNIX authentication failure"
reliability="10" occurrence="1000" from="1:SRC_IP"
to="1:DST_IP" port_from="ANY" time_out="14400"
port_to="ANY" plugin_id="4004"
plugin_sid="2"/>
                    </rules>
                  </rule>
                </rules>
              <rule type="detector" name="PAM UNIX authentication successful detected"
reliability="10" occurrence="1" from="1:SRC_IP" to="1:DST_IP"
port_from="ANY" time_out="10" port_to="ANY" plugin_id="4004"
plugin_sid="1"/>
            </rules>
          </rule>
          <rule type="detector" name="PAM UNIX authentication successful detected"
reliability="7" occurrence="1" from="1:SRC_IP" to="1:DST_IP" port_from="ANY"
time_out="10" port_to="ANY" plugin_id="4004" plugin_sid="1"/>
        </rules>
      </rule>
      <rule type="detector" name="PAM UNIX authentication successful detected" reliability="2"
occurrence="1" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="10" port_to="ANY"
plugin_id="4004" plugin_sid="1"/>
    </rules>
  </rule>
  <rule type="detector" name="PAM UNIX authentication successful detected" reliability="0"
occurrence="1" from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="10" port_to="ANY"
plugin_id="4004" plugin_sid="1"/>
</rules>
</rule>
</directive>

```

**Figure 5.9:** Alienvault's bruteforce XML format

Figure 5.9 shows the directive is defined in the beginning with an id, a name and a priority. The directive is filled with nested rules - each with their required attributes. OSSIM offers a Web user interface (WebUI) that we can use to clone and modify directives and their underlying rules. The WebUI allows for easy reading and understanding of the rules.

NAME	RELIABILITY	TIMEOUT	OCCURRENCE	FROM	TO	DATA SOURCE	EVENT TYPE	ACTION
▼ PAM UNIX authentication failure	1	None	1	ANY	ANY	pam_unix (4004)	SIDs: 2	More +
▼ PAM UNIX authentication failure	4	600	20	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
▼ PAM UNIX authentication failure	4	1200	50	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
▼ PAM UNIX authentication failure	6	1200	100	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
▼ PAM UNIX authentication failure	10	7200	200	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
▼ PAM UNIX authentication failure	10	14400	1000	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
PAM UNIX authentication failure	10	43200	10000	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 2	More +
PAM UNIX authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +
PAM UNIX authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +
PAM UNIX authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +
PAM UNIX authentication successful detected	7	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +
PAM UNIX authentication successful detected	2	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +
PAM UNIX authentication successful detected	0	10	1	1:SRC_IP	1:DST_IP	pam_unix (4004)	SIDs: 1	More +

**Figure 5.10:** OSSIM's web UI showing the rule set for a Bruteforce Attack

As figure 5.10 shows, the rules are set up as nested steps. Once the first level is passed, then those events are compared to the next step of the rule set. In this case, if there is an authentication failure once (shown by the '1' in the 'Occurrence' column) from the 'pam\_unix' data source (plugin id of 4004) then OSSIM will compare these events against the next step of the rule set. OSSIM knows it was an authentication failure because of the event type 'SIDs' that are matched in the detector plugin's rules.

There are three different values that are considered when the overall risk of an attack directive is determined by OSSIM. They are defined below: [67]

- Priority - "The Priority is related to threats and it reflects the importance of a specific attack, having nothing to do with a specific host or environment. It only measures the relative importance of the attack itself." This value can range from 0 - 5 with the default being 1.
- Reliability - "Classical risk-assessment would refer to it as 'probability'. Since it is quite difficult to determine how probable it is for a network to be exposed to certain vulnerabilities, the IDS related 'reliability' approach was considered more appropriate." This value can range from 0 - 10 with the default value being 1.
- Asset Value - "It is assigned to both the source and the destination hosts and represents the importance of the asset to the enterprise"

With the above values defined, OSSIM computes the risk of an attack and decides when it should trigger an alert. The calculation that OSSIM uses to decide the severity of the

correlation directives is

$$Risk = (Priority * Reliability * AssetValue)/25 \quad (5.1)$$

where if  $Risk > 1$ , then OSSIM will send out an alert for attention to these events. When we define our rules for each contextual data feed, we will need to define the above values for each source and rule set.

### 5.4.1 Social Media Context

In our Social Media Context, we want the rule set to heighten OSSIM's sensitivity to the type of attack trending on social media. The detector plugin for this contextual data type will have two rules (i.e. two event types) because the log file design will present OSSIM with a count variable of the number of results returned, and then the actual results written on their own log line.. The count variable event type will be used in the correlation rules because the number of results returned will determine the severity of the attack trending on social media. Hence we will be correlating the number of results returned from a certain social media trending attack with the ruleset of that attack - in our case the bruteforce attack. This will enhance OSSIM by giving it the context that a certain type of attack is more likely and hence OSSIM should be more sensitive to signs of this type of attack.

The social media plugin will end up having four event subtypes. The first event subtype will be the social media descriptions which can be used for review, while the other three subtypes will be based on the count variable returned in the rule of the detector plugin. The event type id will be added to the base brute-force correlation rules as an extra step that will heighten OSSIM's overall risk value. It will ensure that the social media's context count value will affect the degree to which we increase the reliability of the rule. This will mean that the reliability of predicting an attack will grow as the number of results returned from the social media context value grows.

### 5.4.2 Meteorological Context

In our Meteorological Context, we want the rule set to heighten OSSIM's sensitivity to login failures, according to extreme weather conditions in the location of the monitored network. The rationale behind this is that networks in areas with extreme weather may be more vulnerable on days that an attacker knows work personnel will not be present. The detector plugin for this contextual data type will have one rule because it will only receive one type of log. The information in the log will lead to two event subtypes. The weather code will be translated into an extreme weather state or not extreme weather state which are the two different event subtypes. The weather state variable will be used in the correlation rules because it gives OSSIM context as to whether there should be any more or less suspicious behaviour on the monitored network due to the current weather state.

If a non-extreme weather state is returned the rules will go on unchanged because normal weather has been detected. If an extreme weather state is returned, OSSIM's risk value will

be increased according. This extreme weather event sub type will be added to the rule base and will add to the reliability of the attack directive.

### 5.4.3 Calendar Events Context

In our Calendar Events Context, we want the rule set to heighten OSSIM's sensitivity to login failures according to the type of current day in the location of the monitored network. The detector plugin for this contextual data type will have one rule, because it will only receive one type of log. The information extracted from the log will lead to two event subtypes. The current date will be checked against the date in the log. If the dates match, then that becomes a public holiday event subtype. If none of the dates match the current date, then that becomes an inconsequential event subtype.

The public holiday event subtype will cause OSSIM's risk value to increase, because public holidays could mean people are not logging in. Public holidays mean that there will be less activity on the monitored network and hence any activity on these days should be considered suspicious. The public holiday event subtype will add to the reliability of the attack directive.

### 5.4.4 Terror Threat Context

In our Terror Threat Context, we want the rule set to heighten OSSIM's sensitivity to any attacks or suspicious behaviour according to the current terror threat level. The detector plugin for this contextual data type will have one rule, because it will only receive one type of log. The information extracted from the log will lead to the five event subtypes. Depending on the terror threat level returned by our RSS feed application, our script will return the correct event subtype matching the terror threat level in terms of severity.

Each event subtype will cause OSSIM's risk value to increase proportionately with the terror threat level, meaning that the higher the terror threat level, the higher OSSIM's risk value will be increased. This will be done by adding these event subtypes to the rule base and making the reliability of an attack increase proportionately based on the severity of the terror threat level returned.

## 5.5 Summary

This design chapter laid out the plans to design an experiment that would test our hypothesis. In this chapter, all design related topics are covered to ensure that the experiment is properly planned and thought out before its implementation. By ensuring that we have a control test to compare with the modified SIEM that will use contextual data, we will be able to determine whether the addition of contextual data feeds is useful and effective.

Each contextual data application has been designed to ensure that the data is readily available to the SIEM, while the SIEM's internal processes are redesigned to be modified in

such a way as to be able to use the incoming contextual data. For this to occur smoothly, each process involved in this attempt to optimise a SIEM is designed carefully to work together to ensure that the SIEM's performance is not heavily affected.

## Chapter 6

# Implementation of Test Environment and Contextual Data Feeds

In this chapter we look at how the designs defined in the previous chapter are implemented in order for testing to be carried out. The implementation chapter will go over each aspect of our system's setup and configuration. We will first look at our OSSIM system's implementation along with the configurations needed to get the testing platform ready.

After we have introduced a few configurations required for our network device to send the events, we will look at the different contextual data feed programs. These sections will give an overview of the development of the application along with considerations that needed to be made. The section will continue with an in-depth look at applications OSSIM detector plugin configuration, and why it was designed in such a way, followed by an analysis of the rule set put forth for each contextual data type.

Since we do not want this chapter to be misconstrued as a manual, more details relating to the implementation, such as steps to installing the necessary frameworks and programs used, will be included in the appendix. This chapter aims solely at describing the important details of project-related implementation.

### 6.1 Our Testing Environment Setup

In this section of the Implementation chapter, we look at the setup of OSSIM and the setup of the Linux machine that will be sending the events to OSSIM. Figure 6.1 explains a little bit about Oracle's VirtualBox. It is necessary to know that VirtualBox creates a virtual environment that allows us to install any type of operating system type software and use this software as if it were its own independent machine.

<b><u>Application Name:</u></b>	Oracle VM VirtualBox
<b><u>Version:</u></b>	4.3.28
<b><u>Description:</u></b>	VirtualBox is a cross-platform virtualization application which means that it allows multiple installations of different operating systems. The VirtualBox program creates a virtual machine that it installs almost any operating system on, and allows the user to use this virtual machine as if it were your physical machine. This virtual machine mimics a normal installation allowing multiple different operating systems on one machine alongside all their relevant applications.
<b><u>Website:</u></b>	<a href="https://www.virtualbox.org/">https://www.virtualbox.org/</a>

**Figure 6.1:** Information about Oracle's VirtualBox

Once we have our VirtualBox setup complete, we can install the OSSIM system onto a VirtualBox instance. To do this, we use the disk image provided by AlienVault on their website. Figure 6.2 helps describe some information about this disk image.

<b><u>Disk Image:</u></b>	<a href="#"><u>AlienVault_OSSIM.iso</u></a>
<b><u>Version:</u></b>	64bits_5.0.2
<b><u>Description:</u></b>	An <a href="#"><u>.iso</u></a> file type is used to declare a file that is a disk image. This means that a user can 'mount' the disk image and use it as if it were a normal physical disk in a CD-ROM. The OSSIM disk image is a bootable disk image because it allows us to install the OSSIM product solely onto a machine that doesn't have a prior operating system. In short, OSSIM is built on its own operating system and hence requires a clean installation on a clean machine.

**Figure 6.2:** Information about our required OSSIM disk image

<b><u>Disk Image:</u></b>	<a href="#">ubuntu-14.04.4-desktop-amd64.iso</a>
<b><u>Version:</u></b>	14.04.4 LTS
<b><u>Description:</u></b>	Ubuntu is an open source Linux operating system based on <a href="#">Debian's</a> architecture and infrastructure. The disk image can be downloaded from <a href="#">www.ubuntu.com</a> . The version we have chosen to download is the <a href="#">64 bit</a> version because our <a href="#">VirtualBox</a> supports 64bit systems. This disk image will allow the clean installation of Ubuntu onto a virtual machine.

**Figure 6.3:** Information about our required Ubuntu Linux disk image

The process of setting up and configuring our VirtualBox instance to be suitable for the OSSIM installation will be covered in the appendix under 'VirtualBox's OSSIM Configuration'. Since we use VirtualBox for both the OSSIM implementation as well as our Linux machine, it is recommended to check the appendix for a more indepth understanding on how VirtualBox sets up the virtual environments for both our systems.

Once OSSIM and our Linux systems are both running on the same network, we can start configuring both systems to do what we need for our testing.

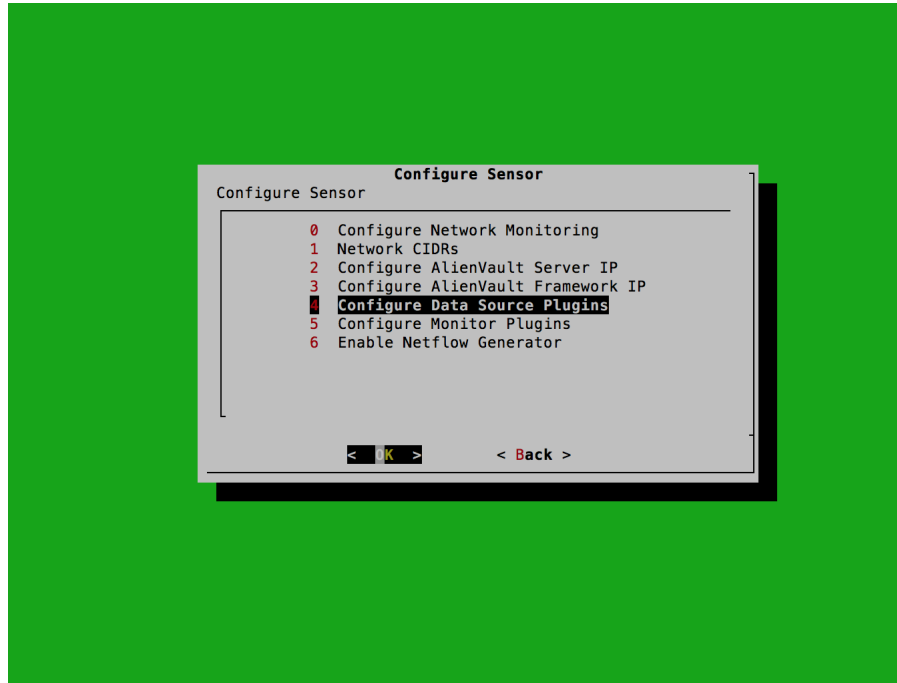
### 6.1.1 OSSIM System

Some important details that are going to be used repeatedly through the rest of this implementation chapter are noted below. The following list will quickly cover the necessary information known to us after the installation of OSSIM.

- OSSIM's web interface can be accessed through OSSIM's IP address entered into a web browser on the local network.
- OSSIM's backend can be accessed through SSH by 'ssh root@;OSSIM's IP address;'
- Log related activity is located in '/var/log/'
- OSSIM Agent related activity is located in '/etc/ossim/agent/'
- OSSIM custom scripts are located in '/usr/share/alienvault/ossim-agent/'

OSSIM comes standard with many rule directives and the associated plugins to collect that data too. This makes OSSIM an easy to implement open source SIEM. The plugins that are readily available are created by other open source security vendors that seek to integrate their product with OSSIM. An example of this is the popular and free network intrusion detection system, Snort. This means that OSSIM already has a plugin detector that will recognise logs generated by Snort, and OSSIM already has a set of directives governing OSSIM on how to handle the logs from Snort.

For our implementation, we need to disable all auxiliary and pre-defined plugins, as to not interfere with our custom feeds. However, we can clone the rule sets of the existing plugins and adapt them to work for us. We can manage (i.e. disable/enable) our active sensor plugins from the OSSIM Setup console.



**Figure 6.4:** OSSIM's setup console

The next range of configurations will all happen within the terminal that OSSIM provides us. This option means that we have full access to the backend of OSSIM which allows us a lot more freedom in configuring the system to suit our needs. We will need this functionality for setting up Rsyslog and calibrating many features for our custom contextual data feeds.

### 6.1.2 Linux Event System

Since our Linux event system is used purely for running the contextual data retrieval programs and sending the information through to OSSIM, there is not much extra configuration that needs to be done to the system in order for it to function as we want it to. Since our contextual data retrieval programs come in the form of a `.jar` file, we need to ensure that the system has Java installed.

The following is a brief list of commonly referred to details about this Linux event system.

- Linux event system's IP address is 192.168.88.101
- Each contextual data retrieval program's log file is found in `/home/jay/`

- `java -jar <program_name>.jar` is the terminal command used to run each contextual data retrieval program.

As can be seen, there is not much configuration needed on the Linux event system. The last part that needs to be configured are the Rsyslog functions, which are covered in the next section.

### 6.1.3 Rsyslog Setup

The Rsyslog configuration is very important to this project because we rely on Rsyslog to send the logs from our Linux event system to OSSIM.

#### The Server Setup: OSSIM

In order for this to be successful, we need to configure Rsyslog through the backend of OSSIM known as jailbreaking. Once we are in the terminal section of OSSIM, we first take a look at our Rsyslog configuration file. To do this we,

```
vi /etc/rsyslog.conf
```

which will open the global Rsyslog configuration file in a text editor known as Vim. It is important to notice in this configuration file that the line

```
IncludeConfig /etc/rsyslog.d/*.conf
```

shows the location Rsyslog should search for individual configuration files that should be loaded and the file extension that Rsyslog configuration files need to have. The screen will look like this:

```

### GLOBAL DIRECTIVES ###
#####

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLLOG_TraditionalDateFormat

#
# Set the default permissions for all log files.
#
$FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf

#####
### RULES ###
#####

#
# First some standard log files. Log by facility.
#
auth,authpriv.*          /var/log/auth.log
*.*;auth,authpriv.none   -/var/log/syslog
#cron.*                  /var/log/cron.log
daemon.*                 -/var/log/daemon.log
kern.*                   -/var/log/kern.log
lpr.*                    -/var/log/lpr.log
mail.*                   -/var/log/mail.log
user.*                   -/var/log/user.log

```

**Figure 6.5:** OSSIM's Rsyslog configuration file

The next step is setting up our Rsyslog configuration files for each contextual data feed. This process is the same for each contextual data feed program. By using the command,

```
cd /etc/rsyslog.d/
```

which will take us to the directory specified in our global Rsyslog configuration file. The configuration files for each contextual data feed will be in the format of `context_type.conf` such that `context_type` is the name of the contextual data feed program. The following code is each in its own Rsyslog configuration file.

```
if $fromhost-ip == '192.168.88.101' and $rawmsg contains 'custom-log-terror' then -/var/log/ossim/terrorlevel.log
```

**Figure 6.6:** The configuration file for our Terror Threat Level Rsyslog setup

```
if $fromhost-ip == '192.168.88.101' and $rawmsg contains 'custom-log-social' then -/var/log/ossim/socialmention.log
```

**Figure 6.7:** The configuration file for our Social Media Rsyslog setup

```
if $fromhost-ip == '192.168.88.101' and $rawmsg contains 'custom-log-weather' then -/var/log/ossim/weatherstatus.log
```

**Figure 6.8:** The configuration file for our Weather Status Rsyslog setup

```
if $fromhost-ip == '192.168.88.101' and $rawmsg contains 'custom-log-holiday' then -/var/log/ossim/publicholiday.log
```

**Figure 6.9:** The configuration file for our Public Holiday Rsyslog setup

As can be seen, the configuration file is similar in each contextual data feed type. The differences are, to which log file each contextual data feed Rsyslog log must be written to on OSSIM, and the custom identifier used to determine which contextual data feed each Rsyslog log line is coming from.

## Linux Event System

OSSIM is ready to receive the Rsyslog log lines from our linux event system. The last Rsyslog configuration needs to be done on the linux event system. The global Rsyslog configuration file on the linux event system is identical to the one on OSSIM. This tells us that the location of the configuration files that we want Rsyslog to load need to be placed in

/etc/rsyslog.d/

Figure 6.10 shows us the Rsyslog configuration file used to send logs from the linux event system to OSSIM. The details will be covered after the figure.

```
$ModLoad imfile

$InputFileName /home/jay/socialmention.log
$InputFileTag custom-log-social
$InputFileStateFile custom-log-social
$InputFileSeverity info
$InputFileFacility local7
$InputRunFileMonitor

$InputFileName /home/jay/weatherstatus.log
$InputFileTag custom-log-weather
$InputFileStateFile custom-log-weather
$InputFileSeverity info
$InputFileFacility local7
$InputRunFileMonitor

$InputFileName /home/jay/terrorlevel.log
$InputFileTag custom-log-terror
$InputFileStateFile custom-log-terror
$InputFileSeverity info
$InputFileFacility local7
$InputRunFileMonitor

$InputFileName /home/jay/publicholiday.log
$InputFileTag custom-log-holiday
$InputFileStateFile custom-log-holiday
$InputFileSeverity info
$InputFileFacility local7
$InputRunFileMonitor

$InputFileName /home/jay/newlogfile.log
$InputFileTag custom-log-entries
$InputFileStateFile custom-log-entries
$InputFileSeverity info
$InputFileFacility local7
$InputRunFileMonitor

$InputFilePollInterval 10

*. * @192.168.88.200
```

**Figure 6.10:** The Rsyslog configuration file for the linux event system

In Figure 6.10 the file is structured per a contextual data feed type (except for the last block which is used for the custom security events). The first line specifies to Rsyslog which module is needed for the rest of the file to be run. This module `imfile` allows Rsyslog to read input files and specify their location. This is seen in the first line of each declaration block starting with `InputFileName <location>/file.log`.

The next line of each block allows us to specify the tag that will accompany each log line sent through to its destination. The next two lines are used to give more information to each log line sent through. These features can be used as another filter. The last line in each declaration block, `InputRunFileMonitor`, enables the file specified to be monitored for changes.

The `InputFilePollInterval 10` line tells the Rsyslog system to check each of the files declared in this configuration file every ten seconds. This value could be optimised to check the files less or more often depending on the urgency of each log file's contents. The final line in this configuration file, `*.* @192.168.88.200` specifies where all these log files must be forwarded to. In this case, it is our OSSIM server's IP address.

### 6.1.4 Control Test Directive Ruleset Implementation

The control test ruleset is the set of correlation directives that are custom made for the events that we will be sending OSSIM. This directive ruleset is adapted from OSSIM's usual 'Bruteforce Attack' directive ruleset. AlienVault recommend that you clone a set of rules similar to what you are trying to achieve and then adapt the directive ruleset accordingly. Figure 6.11 is the adaptation.

**Custom Feed Bruteforce attack**  
Delivery & Attack, Bruteforce Authentication, Linux/Unix - Priority 3

**RULES**

NAME	RELIABILITY	TIMEOUT	OCCURRENCE	FROM	TO	DATA SOURCE	EVENT TYPE	ACTION
authentication failure	1	None	1	ANY	ANY	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	4	600	20	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	4	1200	50	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	6	1200	100	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	10	7200	200	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	10	14400	1000	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication failure	10	43200	10000	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 1	More
authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 2	More
authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 2	More
authentication successful detected	10	10	1	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 2	More
authentication successful detected	7	10	1	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 2	More
authentication successful detected	2	10	1	1:SRC_IP	1:DST_IP	rsyslog_auth_event (9002)	SIDS: 2	More

**DIRECTIVE INFO**

Figure 6.11: Custom Bruteforce attack directive for OSSIM

In Figure 6.11 the data source is the custom made `rsyslog_auth_event` which is the event source that was created specifically for this project. The authentication failure is linked to the `event_type` `SID: 1` while the authentication success is linked to the `event_type` `SID: 2`. As the occurrence of the failure event increases in the time designated by the `timeout` value, so does the `reliability` value. As you can recall, the `reliability` value is used to determine whether OSSIM should signal an alert.

As a successful authentication event is detected, the `reliability` value decreases. Since this is the control test ruleset, it is cloned and adapted for each contextual data source so that OSSIM can react according to the contextual data that is being fed from the contextual data programs.

## 6.2 Contextual Data Feed Programs

The contextual data feed programs are all Java developed applications that fetch contextual data from an online source. The way in which they fetch the data sometimes differs and hence different libraries were used depending on the contextual data type. The following sections look at each contextual data type implementation indepth.

Each contextual data type section will follow the same structure. The first subsection looks at the way in which the contextual data feed program is linked to OSSIM via a detector plugin. The next subsection shows the SQL script for OSSIM's database needed to add the necessary information to our OSSIM database.

The detector plugin implementation section of each contextual data feed will show the actual detector plugin configuration file. All of the configuration files follow the same style of declaring important information that the OSSIM agent uses to extract information from the incoming log files. The figure shows the detector plugin linked to one of the contextual data type's log files.

The plugin starts with some descriptions for anyone looking at the plugin source code (some commented out information about the plugin). The first important information shown is the `plugin_id`. The `plugin_id` is used when enabling the plugin and used in the OSSIM database as a primary key. Hence it is referred to as the ID.

The next section of the plugin is the `[config]`. This section tells us the following:

- `type`= This is the plugin type which could be detector or monitor.
- `enable`= This is whether the plugin is automatically enabled.
- `source`= This is the source file type for the plugin.
- `location`= This is the location of the log file.

- `create_file=` This is used to determine whether the plugin should create a new log file, if one does not already exist.

Examples of the type of log file that this plugin would expect to attempt to parse are provided in the comments. This is done so that it allows easier creation and testing of the regex used in the rules to follow. Each plugin has at least one rule which is used to extract the information from the log file. The rules are structured as follows:

- `event_type=event`. This tells the OSSIM agent that this is a security event.
- `regex=" regexpression "`. This tells the OSSIM agent exactly how to extract the data from the log and gives the extracted data a variable name. This variable name is used to define the data.
- `date={normalize_date(<DateVar>)}`. This line normalises the <DateVar> date format into an OSSIM date format. The `normalize` function is a custom python function stored on OSSIM. We will discuss custom python functions in more depth shortly.
- `plugin_sid=<IntVar>`. This line assigns the log line a plugin type ID. Each plugin has a main ID and multiple type IDs. The type ID is linked to different types of data that is expected from the same contextual data type.
- `userdata<1-9>=<TypeSpecificVar>`. There can be up to nine other types of data extracted and assigned within a plugin.

Most of the detector plugin details are similar across different contextual types, but the notable differences will be documented and discussed. Lots of the information that is extracted might not be used currently but it is still important to extract it because the variables can easily be used for searching the OSSIM database or creating new correlation rules.

The SQL script queries subsection shows how the OSSIM database is prepared for the incoming contextual data. The first two lines are `DELETE` functions which make sure that the OSSIM database doesn't have information stored under that plugin ID and plugin SID that we are going to use for that particular contextual data type. The following lines in the SQL script are `INSERT` statements. The first `INSERT` statement adds the following information into the `plugin` table:

- `id`. This is the ID of the plugin.
- `type`. This is the type of the plugin. Either detector or monitor.
- `name`. This is the name of the plugin.
- `description`. This is the description we choose to give this plugin.

The next few `INSERT` statements add the multiple types that are associated with that plugin. These types are defined in the plugin rules. Since each contextual data plugin has different rules, these `INSERT` statements will be covered in each context's subsection.

### 6.2.1 Social Media Context

In this section we look at the Social Mention API implementation which were used to determine the network attacks that were trending - namely brute force and denial of service attacks. Depending on the number of results returned, we adjust OSSIM's sensitivity to these attacks.

## Detector Plugin Implementation

```
;; SocialMentionContext Feed plugin
;; plugin_id: 9004
;; type: detector
;; description: Plugin for the social mention contextual feed

[DEFAULT]
plugin_id=9004

[config]
type=detector
enable=yes
source=log
location=/var/log/ossim/socialmention.log
create_file=true

#Feb 18 16:14:43 jay-VirtualBox custom-log-social SocialMentionContextCount : 8
#Feb 18 16:14:43 jay-VirtualBox custom-log-social SocialMentionContext: Massive Brute-Force
#Attack on Alibaba Affects Millions @ infosecurity-magazine.com @ google_news @ news
#Feb 18 16:14:43 jay-VirtualBox custom-log-social SocialMentionContext: Alibaba hit by
#massive brute-force password hack @ itproportal.com @ google_news @ news

[Social Mention Context Event - Count Rule]
event_type=event
regex="(P<date>\w{3}\s+\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2})\s(P<hostname>\S+)\s
(P<log_tag>\S+)\s(P<log_type>\S+)\s:\s(P<count>\d*)"
date={normalize_date($date)}
userdata1={$log_tag}
userdata2={$log_type}
userdata3={$count}
plugin_sid={checkSocialCount($count)}

[Social Mention Context Event - Event Descriptions Rule]
event_type=event
regex="(P<date>\w{3}\s+\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2})\s(P<hostname>\S+)\s
(P<log_tag>\S+)\s(P<log_type>\S+)\s(P<description>(\w+\s*\-*\'*)*)\s(P<domain>
(\w+\s*\-*\'*.*)*)\s(P<source>(\w+\s*\-*\'*.*)*)\s(P<type>(\w+\s*\-*\'*.*)*)"
date={normalize_date($date)}
userdata1={$log_tag}
userdata2={$log_type}
userdata3={$holiday}
userdata4={$description}
plugin_sid=4
```

**Figure 6.12:** OSSIM Agent plugin configuration file for the Social Media context

The noteworthy information regarding our Social Media contextual plugin before the plugin rules are:

- `plugin_id=9004`. This is the social media's plugin id and it is referenced throughout the Social Media Context section.

- `location=/var/log/ossim/socialmention.log`. This is the location of the log file that is populated by our Rsyslog implementation and sends the social mention program's log lines.

The following rule section of the plugin contains some context specific information. The Social Media contextual plugin contains two rules. The first one relates to the count variable that can be found in the social mention log. The second rule relates to parsing the returned stories found trending on social media through Social Mention.

The interesting line, `plugin_sid={checkSocialCount(\$count)}` of the first rule in the Social Media plugin accesses the `checkSocialCount` custom python function. This is a simple function that attempts an integer input variable and returns an integer. The custom function uses the input integer to decide the severity of the trending attack on social media. The custom function returns an ID in the form of an integer. This ID becomes the `plugin_sid` which is linked to different severities in the OSSIM database.

The second rule in the Social Media plugin is the a standard rule to store the information coming into the OSSIM agent. The `plugin_sid` is assigned to 4 which isn't linked to a severity but rather to a description type.

## SQL Script Queries

```
-- Social Mention Context
-- plugin_id: 9004
DELETE FROM plugin WHERE id = "9004";
DELETE FROM plugin_sid where plugin_id = "9004";

INSERT INTO plugin (id,type,name,description) VALUES (9004,1,'social_mention_context',
    'Social Mention Context Feed');
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,relability)
VALUES (9004,1,NULL,NULL,'social_mention_context:LOW',3,2);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,relability)
VALUES (9004,2,NULL,NULL,'social_mention_context:MEDIUM',3,5);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,relability)
VALUES (9004,3,NULL,NULL,'social_mention_context:HIGH',3,8);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,relability)
VALUES (9004,4,NULL,NULL,'social_mention_context:DESCRIPTION',3,1);
```

**Figure 6.13:** Social Media context SQL script for OSSIM

Figure 6.13 script shows the necessary `INSERT` statements for our Social Media context. The noteworthy information is as follows with the exception that the `plugin_id=9004`:

Plugin_sid	Description	Priority	Reliability
1	LOW	3	2
2	MEDIUM	3	5
3	HIGH	3	8
4	DESCRIPTION	3	1

**Table 6.1:** OSSIM's Social Media SQL information

In Table 6.1 the SQL `INSERT` statements from the SQL script insert the different plugin types which are in turn linked to a different severity. Although the priority value is the same, the reliability value changes based on the increasing severity. The 'DESCRIPTION' plugin\_sid, has a reliability of one because we do not want OSSIM to use these log lines to trigger alerts. As the reliability value is increased, so is the calculation that is done to decide whether OSSIM should trigger an alert.

## 6.2.2 Meteorological Context

In this section we look at the Open Weather Map API implementation which we used to determine the weather severity in an area. Depending on the severity of the weather returned, we adjust OSSIM's sensitivity to failed and successful login attempts.

Figure 6.14 shows the detector plugin for the Meteorological Context, there is only one rule governing how OSSIM is to parse the log lines received via our weather status contextual data program. The interesting line in this configuration file is the final line, `plugin_sid={...}`, which takes in the parsed data named 'code' and feeds it into another custom python function called `checkWeatherStatus`. This custom python function uses the weather code from the Open Weather Map API and returns whether the code is linked to extreme weather conditions.

## Detector Plugin Implementation

```
;; WeatherStatusContext Feed plugin
;; plugin_id: 9005
;; type: detector
;; description: Plugin for the weather status contextual feed

[DEFAULT]
plugin_id=9005

[config]
type=detector
enable=yes
source=log
location=/var/log/ossim/weatherstatus.log
create_file=true

#Feb 23 15:05:01 jay-VirtualBox custom-log-weather WeatherStatusContext: Cape Town @ 800
#Feb 23 15:06:52 jay-VirtualBox custom-log-weather WeatherStatusContext: Test @ 902

[Weather Status Context Event - Check Rule]
event_type=event
regexp="(P<date>\w{3}\s+\d{1,2}\s\d{1,2}\:\d{1,2}\:\d{1,2})\s(P<hostname>\S+)\s
(P<log_tag>\S+)\s(P<log_type>\S+)\s(P<city_name>(\S+\s+)*)\s(P<code>\d*)"
date={normalize_date($date)}
userdata1={$log_tag}
userdata2={$log_type}
userdata3={$code}
userdata4={$city_name}
plugin_sid={checkWeatherStatus($code)}
```

**Figure 6.14:** OSSIM Agent plugin configuration file for the Weather Status context

The noteworthy information regarding our Meteorological contextual plugin is:

- `plugin_id=9005`. This is the weather status' plugin id and it is referenced throughout the Meteorological Context section.
- `location=/var/log/ossim/weatherstatus.log`. This is the location of the log file that our Rsyslog implementation sends the weather status program's log lines.

## SQL Query

```
-- Weather Status Context
-- plugin_id: 9005
DELETE FROM plugin WHERE id = "9005";
DELETE FROM plugin_sid where plugin_id = "9005";

INSERT INTO plugin (id,type,name,description) VALUES (9005,1,'weather_status_context',
'Social Mention Context Feed');
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9005,1,NULL,NULL,'weather_status_context:NORMAL',3,1);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9005,2,NULL,NULL,'weather_status_context:EXTREME',3,5);
```

**Figure 6.15:** Weather Status context SQL script for OSSIM

Figure 6.15 shows the necessary **INSERT** statements for our Meteorological context. The noteworthy information is as follows with the exception that the `plugin_id=9005`:

Plugin.sid	Description	Priority	Reliability
1	NORMAL	3	1
2	EXTREME	3	5

**Table 6.2:** OSSIM's Meteorological SQL Information

In Table 6.2 our Meteorological Context data feed requires only two different `plugin_sid`. The first `plugin_sid=1` is linked to normal weather codes returned by our `checkWeatherStatus` function while the `plugin_sid=2` is linked to an 'extreme' weather condition. The extreme weather condition is given a high reliability value because that will cause OSSIM to be more sensitive to attacks after the necessary correlation rule has been satisfied.

### 6.2.3 Calendar Event Context

In this section we look at the Google Calendar API implementation which we used to determine public holidays or non working days. Depending on the result returned from the comparison of public holiday dates and the current date, we adjust OSSIM's sensitivity to failed and successful login attempts accordingly.

## Detector Plugin Implementation

```
;; PublicHolidayContext Feed plugin
;; plugin_id: 9003
;; type: detector
;; description: Plugin for the public holiday contextual feed

[DEFAULT]
plugin_id=9003

[config]
type=detector
enable=yes
source=log
location=/var/log/ossim/publicholiday.log
create_file=true

#Jan 26 15:17:01 jay-VirtualBox custom-log-holiday PublicHolidayContext:
# Holy Saturday @ 2016-03-26
#Jan 26 15:07:05 jay-VirtualBox custom-log-holiday PublicHolidayContext:
# Mother's Day @ 2017-05-14

[Public Holiday Context Event - Generic rule #1 ]
event_type=event
regex="(P<date>\w{3}\s+\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2})\s(P<hostname>\S+)\s(P<log_tag>\S+)\s(P<log_type>\S+)\s(P<holiday>(\w+\s*\')*\s{0,20})\s@\s+(P<custom_date>\d+\-\d+\-\d+)"
date={normalize_date($date)}
hostname={$hostname}
userdata1={$log_tag}
userdata2={$log_type}
userdata3={$holiday}
userdata4={checkPublicHoliday($custom_date)}
plugin_sid={checkPublicHoliday($custom_date)}
```

**Figure 6.16:** OSSIM Agent plugin configuration file for the Calendar Event context

The noteworthy information regarding our Calendar Event contextual plugin is:

- **plugin\_id=9003.** This is the public holiday's plugin id and it is referenced throughout the Calendar Event Context section.
- **location=/var/log/ossim/publicholiday.log.** This is the location of the log file that our Rsyslog implementation sends the public holiday program's log lines.

In Figure 6.16 there is only one rule for the public holiday program's log lines sent to the OSSIM agent. This rule parses the same general information and gets the date accompanied with each public holiday. This `<custom_date>` value is fed into the `checkPublicHoliday(<custom_date>)` python function which compares that `<custom_date>` value to the current date. The custom function then returns a value linked to the appropriate `plugin_sid`.

## SQL Query

```
-- Public Holiday Context
-- plugin_id: 9003
DELETE FROM plugin WHERE id = "9003";
DELETE FROM plugin_sid where plugin_id = "9003";

INSERT INTO plugin (id,type,name,description) VALUES (9003,1,'public_holiday_context',
'Public Holiday Context Feed');
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9003,1,NULL,NULL,'public_holiday_context:FAILED',3,1);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9003,2,NULL,NULL,'public_holiday_context:SUCCESS',3,5);
```

**Figure 6.17:** Calendar Event context SQL script for OSSIM

Figure 6.17 shows the necessary **INSERT** statements for our Calendar Event context. The noteworthy information is as follows with the exception that the `plugin_id=9003`:

Plugin_sid	Description	Priority	Reliability
1	FAILED	3	1
2	SUCCESS	3	5

**Table 6.3:** OSSIM's Calendar Event SQL Information

In Table 6.3, the Calendar Event context SQL scripts add only two different `plugin_sids`. The first `plugin_sid=1` is linked to the 'failed' response from our custom python function `checkPublicHoliday()`. This would be returned if the current date differs from the input value. This then means that the reliability value will be low because it is not a public holiday and hence OSSIM need not be more sensitive. The second `plugin_sid=2` is linked to the 'success' response which means that the public holiday date and the current date have matched. The reliability is increased because people should not be at work on a public holiday.

### 6.2.4 Terror Level Context

In this section we look at our RSS feed implementation of the British Government's MI5 terror level feed. We determine what the terror level is and adjust OSSIM's sensitivity according to the terror level.

## Detector Plugin Implementation

```
;; TerrorLevelContext Feed plugin
;; plugin_id: 9006
;; type: detector
;; description: Plugin for the terror level contextual feed

[DEFAULT]
plugin_id=9006

[config]
type=detector
enable=yes
source=log
location=/var/log/ossim/terrorlevel.log
create_file=true

#Feb 26 14:28:45 jay-VirtualBox custom-log-terror TerrorLevelContext: SEVERE

[Terror Level Context Event - Check Rule]
event_type=event
regexp="(P<date>\w{3}\s+\d{1,2}\s\d{1,2}\:\d{1,2}\:\d{1,2})\s(P<hostname>\S+)\s(P<log_tag>\S+)\s(P<log_type>\S+)\s(P<level>\S+)"
date={normalize_date($date)}
userdata1={$log_tag}
userdata2={$log_type}
userdata3={$level}
plugin_sid={checkTerrorLevel($level)}
```

**Figure 6.18:** OSSIM Agent plugin configuration file for the Terror Level context

The noteworthy information regarding our Terror Level contextual plugin is:

- `plugin_id=9006`. This is the social media's plugin id and it is referenced throughout the Terror Level Context section.
- `location=/var/log/ossim/terrorlevel.log`. This is the location of the log file that our Rsyslog implementation sends the terror threat level program's log lines.

In Figure 6.18, there is only one rule that is used to check the terror threat level and return an appropriate `plugin_sid`. The `checkTerrorLevel(<level>)` custom python function takes a value parsed from the log and returns a number linked to the appropriate `plugin_sid`.

## SQL Query

```

-- Terror Level Context
-- plugin_id: 9006
DELETE FROM plugin WHERE id = "9006";
DELETE FROM plugin_sid where plugin_id = "9006";

INSERT INTO plugin (id,type,name,description) VALUES (9006,1,'terror_level_context',
'Terror Level Context Feed');
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9006,1,NULL,NULL,'terror_level_context:LOW',3,1);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9006,2,NULL,NULL,'terror_level_context:MODERATE',3,2);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9006,3,NULL,NULL,'terror_level_context:SUBSTANTIAL',3,3);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9006,4,NULL,NULL,'terror_level_context:SEVERE',3,4);
INSERT INTO plugin_sid (plugin_id,sid,category_id,class_id,name,priority,reliability)
VALUES (9006,5,NULL,NULL,'terror_level_context:CRITICAL',3,5);

```

**Figure 6.19:** Terror Level context SQL script for OSSIM

Figure 6.19 shows the necessary **INSERT** statements for our Terror Level context. The noteworthy information is as follows with the exception that the `plugin_id=9006`:

Plugin_sid	Description	Priority	Reliability
1	LOW	3	1
2	MODERATE	3	2
3	SUBSTANTIAL	3	3
4	SEVERE	3	4
5	CRITICAL	3	5

**Table 6.4:** OSSIM's Terror Level SQL Information

In Table 6.4, the Terror Level context SQL script creates five different levels of severity linked to the terror threat level that may be returned by our `checkTerrorLevel()`. The increasing severity in terror threat level is associated with the increasing `plugin_sids`. As the severity increases so does the reliability of each `plugin_sid` which will cause OSSIM to be more sensitive as the severity of the terror threat level increases. The description of each `plugin_sid` is the usual terror threat level description found on the MI5's website.

## 6.3 Summary

This chapter used the designs defined in chapter 5 and followed the steps taken to implement the necessary systems for our testing chapter. The implementation of OSSIM and the contextual data application is important to document because it ensures that the testing environment is set up adequately for accurate tests.

The OSSIM system was implemented using a virtual machine which we linked to another virtual machine running Ubuntu. The Ubuntu machine was implemented to run our contextual data applications and send the resulting log files through Rsyslog to OSSIM. The next step in our contextual data flow is the manner in which OSSIM uses the log data it receives.

Each contextual data application has its own implementation in OSSIM. OSSIM requires each contextual data feed to have its own detector plugin to decode and normalise the data contained within the log file. OSSIM also requires that separate SQL queries be run for each contextual data feed to ensure that OSSIM has the correct database information linked to each plugin.

# Chapter 7

## Testing and Results

In this chapter the testing phase of the implementation is reported. This phase aims to prove that contextual data feeds can be augmented with an open source SIEM system such as OSSIM. This phase will determine the hypothesis outcome and whether it can be proven. The testing phase serves as a 'proof of concept' phase where the implementation aims to show the viability of the augmentation of contextual data. Additionally, this chapter presents the results of the tests. In chapter 8, the results are analysed in depth.

The following results are specific to the type of contextual data implementation. Each context data implementation uses a proof of concept section to add to the investigation of the hypothesis. This is followed by the results recorded from the tests of that context data implementation. Firstly, a look at the control test is considered, in order to get a control test result that we can use for comparison with each contextual data implementation test.

### 7.1 Control Test Implementation Testing and Results

The control test was the test in which OSSIM was fed security events only - no contextual data feed had been enabled. This also means that the rules that pertain to the contextual data feeds were not present in the system. This set of tests was necessary to determine OSSIM's stock standard detection capabilities.

This is a necessary step for comparison with the tests with the contextual data feeds enabled. There is one type of attack we simulate because our different contextual data feed use cases concern themselves with this type of attack in this study. The following list defines parameters of the test:

- Brute Force attack where many attempts to log into the enterprise network are attempted.
- Randomly generated events are defined as successful and failed login attempts on the network.
- Each event contains just enough data for OSSIM to recognise a successful or failed login attempt has occurred.

- The number of events, 10000, is used because OSSIM only allows a certain number of events before it discards events. This is part of the restrictions of a free SIEM.
- Alarms are defined as OSSIM's correlation engine triggering an alert from the events received.

Table 7.1 shows the control test results. The test had 10000 randomly created events run through the OSSIM system three times. The average is shown in the table below. This kind of test will happen for each type of contextual data and the average results will be shown in a similar table to make comparisons easier.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	37
2	100000	39
3	100000	35

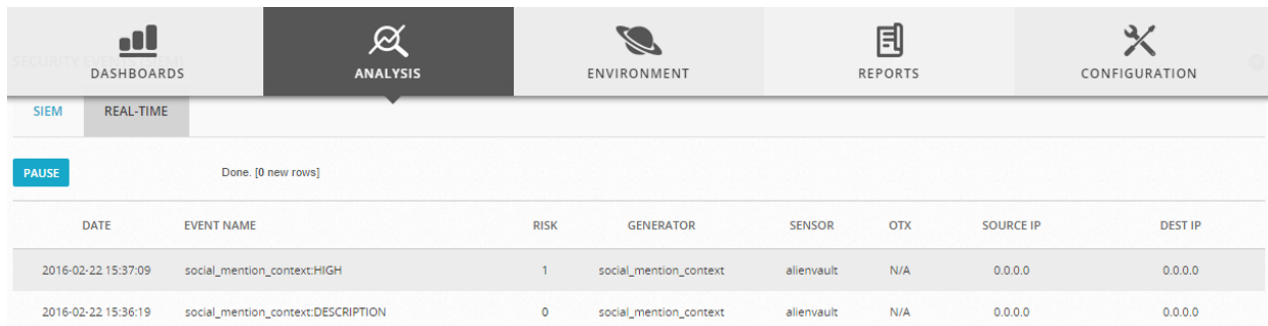
**Table 7.1:** Table of event number with number of alarms raised during each test

The results presented in table 7.1 are used in chapter 8 as the control test comparison. The test was run with the same number of events because each time the events are generated, they are generated randomly. The same set of results was not used throughout the test because a randomised test done multiple times would give us a more accurate average result for comparison. If one set of events is used, we run the risk of the event set being quite static.

## 7.2 Social Media Implementation Testing and Results

### 7.2.1 Social Media Implementation Testing

In figure 7.1, OSSIM recognises the detector plugin and has used that plugin to handle the log files being populated in the Social Mention log file. The figure is of OSSIM's web interface which gives us a user friendly display of OSSIM's functions and current processes. The Social Mention detector plugin configuration file can be seen in Chapter 6 - Implementation.



DATE	EVENT NAME	RISK	GENERATOR	SENSOR	OTX	SOURCE IP	DEST IP
2016-02-22 15:37:09	social_mention_context:HIGH	1	social_mention_context	alienvault	N/A	0.0.0.0	0.0.0.0
2016-02-22 15:36:19	social_mention_context:DESCRIPTION	0	social_mention_context	alienvault	N/A	0.0.0.0	0.0.0.0

**Figure 7.1:** Social Mention Contextual Feed in the OSSIM Web Interface

OSSIM has successfully used the regex expression found in the detector plugin configuration to extract social media related contextual information for OSSIM to run through its correlation engine. If we access this log file line through the web interface, we can see the information that OSSIM has extracted. The following figure is another screen shot of an example of the information that OSSIM may extract from the log file.

In our social media contextual data type, we see that OSSIM has extracted the number of results returned by the Social Mention API regarding the attack. This allows OSSIM to know whether the attack is trending and hence being spoken about at this moment. The term that will be searched for is 'brute force attacks'.

The screenshot displays the OSSIM web interface for social media mention data. The top navigation bar includes DASHBOARDS, ANALYSIS (selected), ENVIRONMENT, REPORTS, and CONFIGURATION. The main content area is titled 'social\_mention\_context:DESCRIPTION' and includes an ACTIONS button. Below this, there are two columns of metadata:

DATE	2016-02-22 15:36:19 GMT+2:00	CATEGORY	N/A
ALIENVault SENSOR	alienvault [192.168.88.200]	SUB-CATEGORY	N/A
DEVICE IP	192.168.88.200 [eth0]	DATA SOURCE NAME	social_mention_context
EVENT TYPE ID	4	DATA SOURCE ID	9004
UNIQUE EVENT ID#	d96911e5-aa52-0800-2704-214f4998d5a0	PRODUCT TYPE	Unknown type
PROTOCOL	TCP	ADDITIONAL INFO	N/A

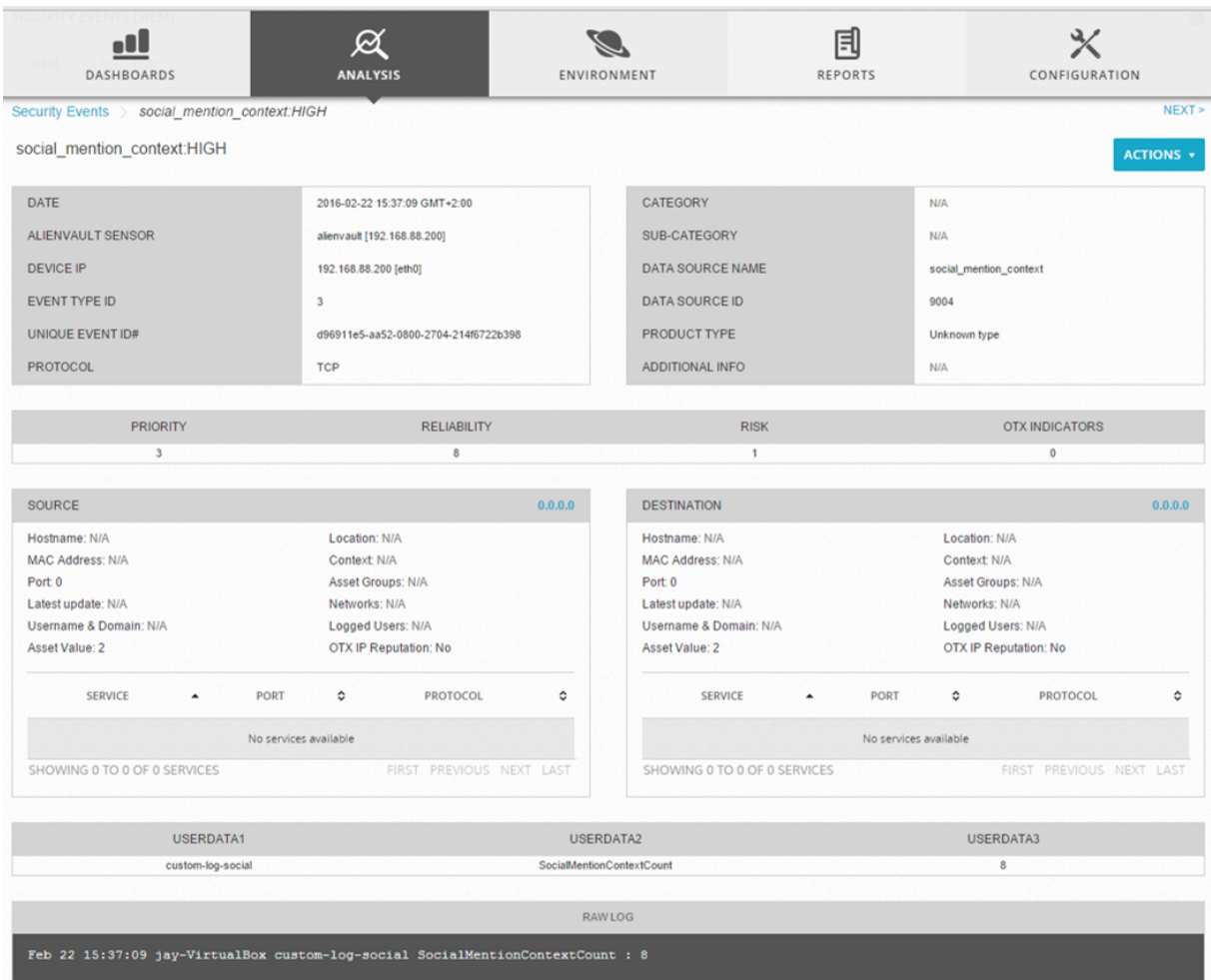
Below the metadata, there is a table of OTX INDICATORS:

PRIORITY	RELIABILITY	RISK	OTX INDICATORS
3	1	0	0

The interface also features two panels for SOURCE and DESTINATION, each with a 0.0.0.0 score. These panels display various attributes like Hostname, MAC Address, Port, Latest update, Username & Domain, and Asset Value. Below these panels are two tables for SERVICE, PORT, and PROTOCOL, both showing 'No services available'. At the bottom, there is a RAW LOG section with a message: 'Feb 22 15:36:19 jay-VirtualBox custom-log-social SocialMentionContext: NASA breach shows again that brute force password attacks work @ itworldcanada.com @ google\_news @ news'.

**Figure 7.2:** Social Mention data extracted from the feed

Figure 7.2 shows the OSSIM web interface of the actual description of the news trending on social media news portals. This description type log allows the system to store the relevant news heading found so that we can look back and see the actual article data. Figure 7.3 is the warning log. This log data also extracted from the Social Mention contextual feed, shows the number of results returned when social media was used to find trending phrases that may be helpful to OSSIM. The number of results represents how popular the searched word is at the moment on social media. Hence if the search word is an attack that we are looking out for, we now know that it is being used more and our SIEM should be more sensitive to its attack pattern. This is how the rules are modified. To ensure that OSSIM is more sensitive relative to the popularity of the attack on social media.



**Figure 7.3:** Social Mention data extracted as a warning in the OSSIM Web Interface

The number of results of the trending word is run through a custom script which returns a severity level that is used in the rules. Using our example above, the key word search on social media is 'brute force'. The system's rules have been successfully augmented to respond appropriately to the high severity result returned. This means that OSSIM's correlation engine will be more sensitive to brute force attacks on the network.

social media trending low	1	None	1	✚ ANY	✚ ANY	✚ <a href="#">social_mention_context (9004)</a>	✚ SIDs: 1
social media trending medium	+3	None	1	✚ ANY	✚ ANY	✚ <a href="#">social_mention_context (9004)</a>	✚ SIDs: 2
social media trending high	+5	None	1	✚ ANY	✚ ANY	✚ <a href="#">social_mention_context (9004)</a>	✚ SIDs: 3

**Figure 7.4:** The rules added to OSSIM for Social Media

Figure 7.4 shows the rules that are embedded in OSSIM's current rule set. These rules are considered only when OSSIM receives logs from the social media application. As figure

7.4 shows, the rule reliability is increased depending on the severity of the number of results returned from the Social Mention API.

### 7.2.2 Limitations of the Social Media Implementation Testing

The Social Mention API is very powerful and extends its social media coverage to many different social media platforms. This actually means that the amount of information returned can be too much, especially since we want our contextual data feed to be supplementary - too many results could cause strain on OSSIM's correlation engine. The API doesn't allow us to be specific enough to limit the results and hence we use the number of results as an indication of a trending topic.

### 7.2.3 Social Media Implementation Results

The following results are recorded from the test using 10000 randomly generated security events augmented with a social media context feed. This means that OSSIM's usual 'brute-force' attack rule set is in place with the addition of the embedded social media context feed rules. This test is repeated with social media information such that it would trigger the system to heighten its sensitivity. Due to the varying levels of rule sensitivity linked to the number of results returned from the social media application, the test is run for each sensitivity level to determine whether the new social media directives are effective.

The first table is the results when the social media application returns under five results linked to the attack that we are investigating - the brute force attack. This is classed as 'LOW' and hence the rule does not increase the sensitivity of OSSIM.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	36
2	100000	40
3	100000	37

**Table 7.2:** Table of event number with number of alarms raised during each test with Social Media Context Level Low

The next table is the results when the social media application returns more than five results but less than ten linked to the attack that we are investigating. This is classed as 'MEDIUM' and hence the rule increases the brute force directive sensitivity by adding three to the current reliability used to calculate the risk.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	42
2	100000	39
3	100000	43

**Table 7.3:** Table of event number with number of alarms raised during each test with Social Media Context Level Medium

The final table is the results when the social media application returns more than ten results linked to the attack that we are investigating. This is classed as 'HIGH' and hence the rule increases the bruteforce directive sensitivity by adding five to the current reliability used to calculate the risk.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	47
2	100000	51
3	100000	48

**Table 7.4:** Table of event number with number of alarms raised during each test with Social Media Context Level High

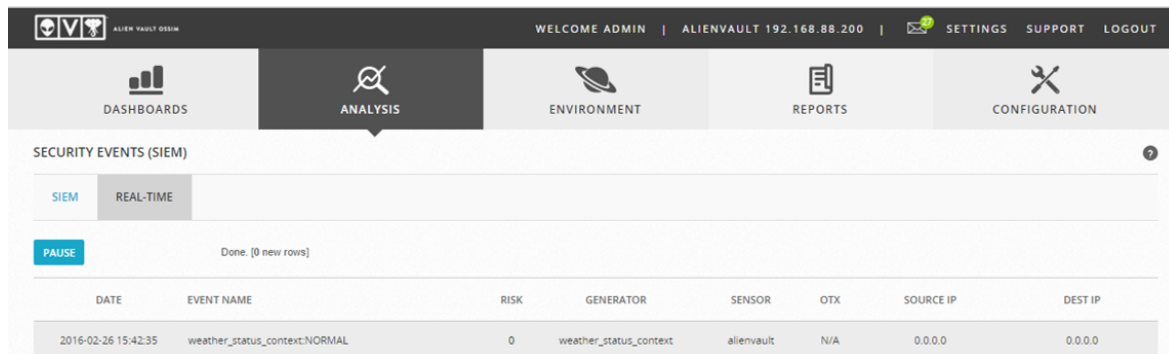
These results are discussed further in the next chapter, however in summary, each level of social media context clearly affects OSSIM's interpretation of the security information being passed into it. As the number of social media results increases, so does the social media context level, and consequently an increase in OSSIM's sensitivity.

## 7.3 Meteorological Information Implementation Testing and Results

### 7.3.1 Meteorological Information Implementation Testing

In figure 7.5 the successful implementation of a meteorological data feed into OSSIM is presented. The OSSIM web interface recognises that there is a detector plugin enabled which

relates to our meteorological information data feed. See the OpenWeatherMap detector plugin configuration file in Chapter 6 -Implementation.



The screenshot shows the AlienVault OSSIM web interface. The top navigation bar includes 'WELCOME ADMIN', the IP address 'ALIENVault 192.168.88.200', and links for 'SETTINGS', 'SUPPORT', and 'LOGOUT'. Below this is a main menu with 'DASHBOARDS', 'ANALYSIS' (selected), 'ENVIRONMENT', 'REPORTS', and 'CONFIGURATION'. The 'ANALYSIS' section is titled 'SECURITY EVENTS (SIEM)' and has tabs for 'SIEM' and 'REAL-TIME'. A 'PAUSE' button and a status 'Done. [0 new rows]' are visible. A table displays a single log entry with the following data:

DATE	EVENT NAME	RISK	GENERATOR	SENSOR	OTX	SOURCE IP	DEST IP
2016-02-26 15:42:35	weather_status_context:NORMAL	0	weather_status_context	alienvault	N/A	0.0.0.0	0.0.0.0

**Figure 7.5:** Weather Status log coming into OSSIM's web ui

Figure 7.6 shows the log information that OSSIM is extracting through the regex expression found in the OpenWeatherMap detector plugin configuration file. The important piece of information to notice in this figure is that the weather code is extracted from the log file and run through a custom script to decide what the weather code actually means. These weather codes can be found on the OpenWeatherMap website or in the appendices. The custom script returns a severity level that OSSIM uses in the rules.

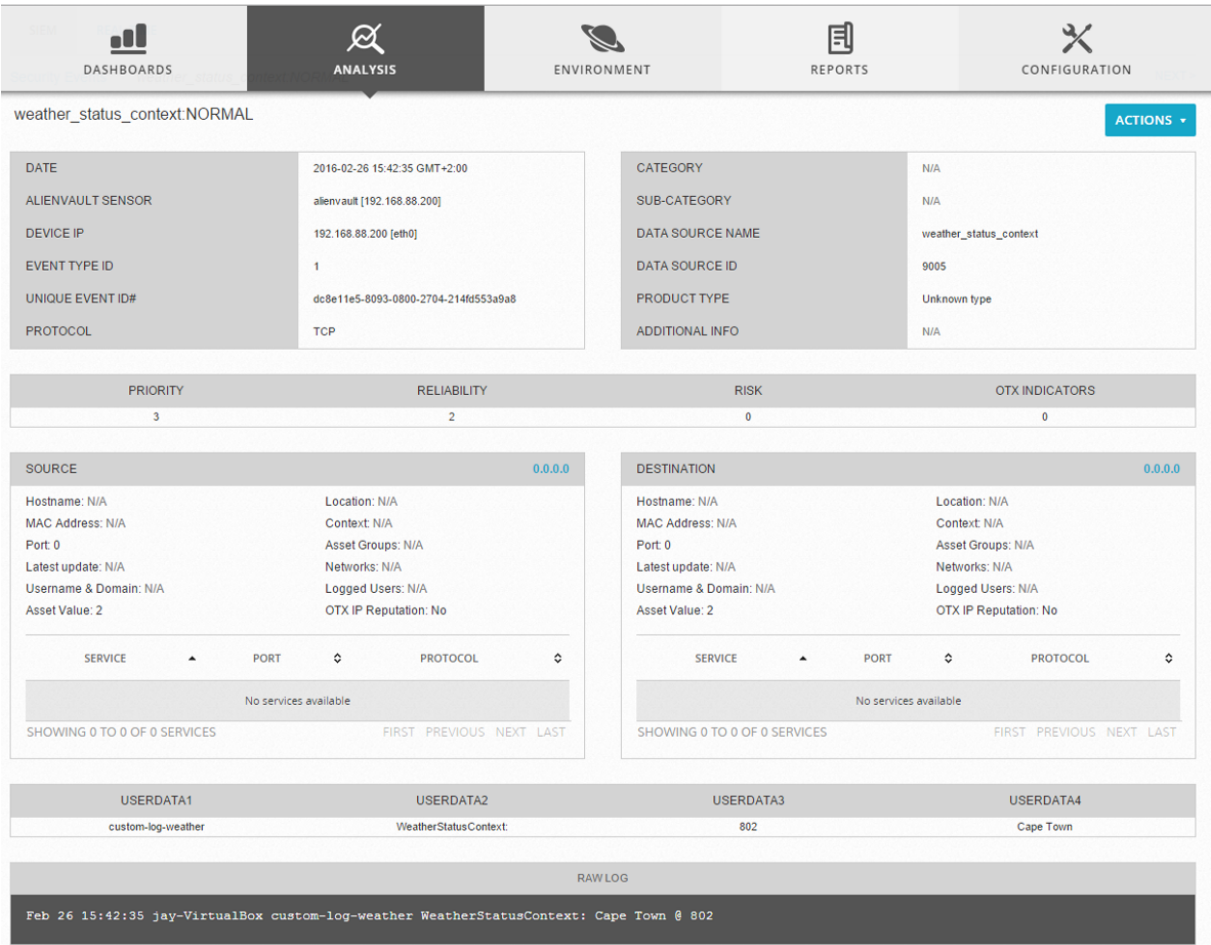


Figure 7.6: Event data extracted from a weather status log event

In Figure 7.6, the weather condition of 'normal' is returned by the custom script. If 'extreme' was returned by the custom script, this would prompt OSSIM’s correlation engine to raise its sensitivity toward rules involving bad login attempts, because fewer work personnel would be using the network on a day that they can not be at work. This type of day would be perfect for attackers, because they know that less people would be using the network. Hence we have satisfied one of the use cases for meteorological contextual data. To show that we can implement this contextual data type to satisfy the second use case, a different test was run using different types of security events.

weather status normal	1	None	1	ANY	ANY	weather_status_context (9005)	SIDs: 1
weather status extreme	+5	None	1	ANY	ANY	weather_status_context (9005)	SIDs: 2

Figure 7.7: Rules added to OSSIM for Meteorological Context

Figure 7.7 shows the rules that OSSIM considers when it receives events relating to the current weather status. One rule is present for the 'normal' weather situation and one rule present for the 'extreme' weather status. These rules each affect OSSIM's reliability value of the correlation engine when this event is triggered. These rules are embedded in the brute force attack rules for testing.

### 7.3.2 Limitations of the Meteorological Information Implementation Testing

The use of meteorological information is limited to location. By using the OpenWeatherMap API, we must specify the location from which we want the weather information. Hence our limitation is that OSSIM would need to specify the location of the network before retrieving the appropriate weather information. On that same note, if OSSIM is monitoring networks that span multiple locations, then specifying one location will become a limitation. The following are possible solutions to this limitation.

- Have a custom script retrieve the weather information needed using the OpenWeatherMap API except using python and taking the location of the network stored in a variable on the OSSIM system. This would mean that the location needs to be specified somewhere on the system. However, if the network is a company that spans over many locations, then this may cause some issues with this approach.
- Have the log files generated by the OpenWeatherMap context application write the location along with the weather code into the log files. Then have a custom script get the location from OSSIM and apply the appropriate weather severity according to the location.

### 7.3.3 Meteorological Information Implementation Results

The following results are recorded from the test using 10000 randomly generated security events augmented with a meteorological context feed. This means that OSSIM's standard bruteforce attack rules are in place with the addition of the embedded weather context information rules. This test is repeated with current weather information such that it would trigger the system to adapt its sensitivity according to the nature of the contextual data. There is an elevated rule sensitivity state linked to the nature of the weather returned from the weather context application, hence the test is run for this sensitivity level to determine whether the new meteorological context directives are effective.

Table 7.5 records the results from this test. In this test the weather context returns an 'EXTREME' state. This state is linked to extreme weather such as heavy snow, hurricanes etc because this is the type of weather that would mean no work on that day. Hence the rule increases the bruteforce directive sensitivity by adding five to the current reliability used to calculate the risk.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	46
2	100000	49
3	100000	48

**Table 7.5:** Table of event number with number of alarms raised during each test with an Extreme Meteorological Context Level

## 7.4 Calendar Event Information Implementation Testing and Results

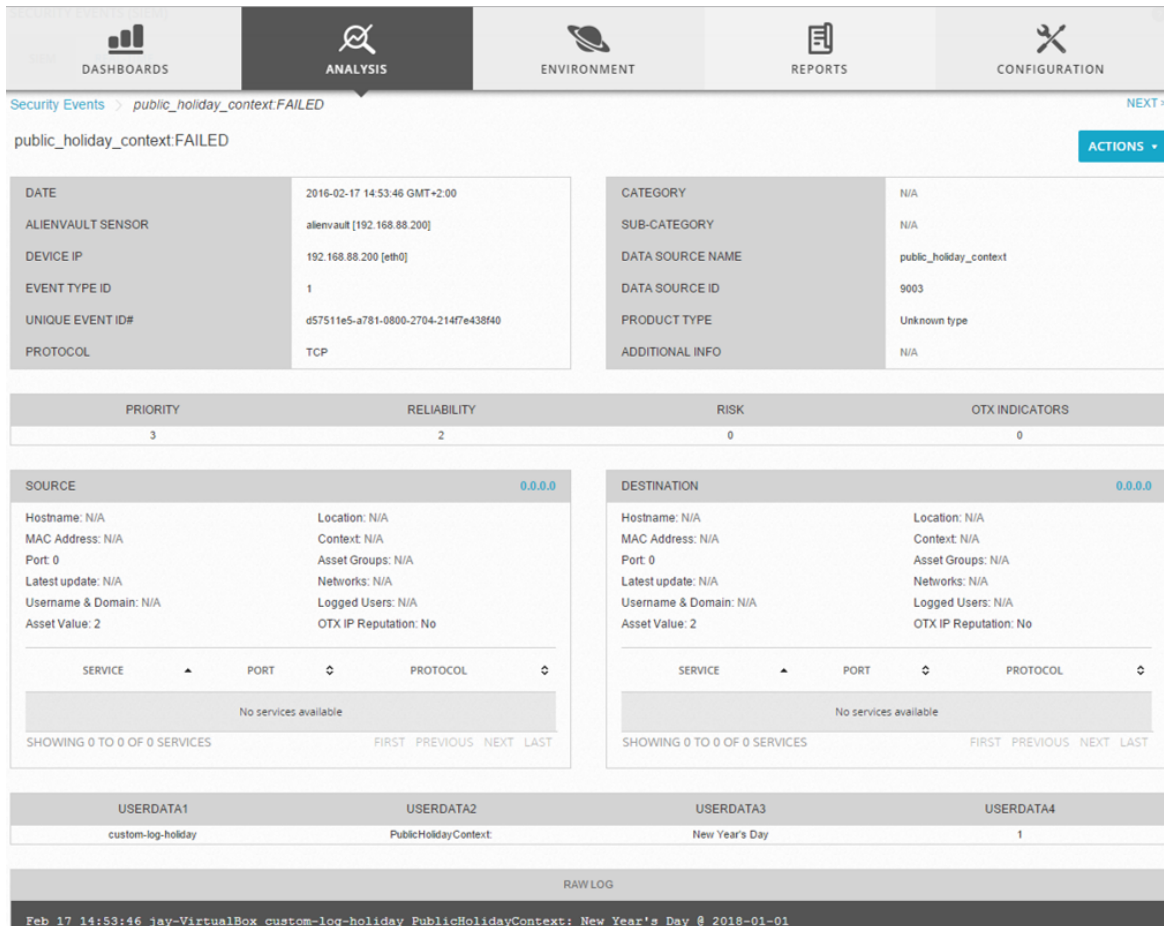
### 7.4.1 Calendar Event Information Implementation Testing

Figure 7.8 shows OSSIM’s web interface recognising our calendar event contextual data feed. The web interface shows the detector plugin is enabled and working. See the Calendar Event Information detector plugin configuration file in Chapter 6 - Implementation.

DATE	EVENT NAME	RISK	GENERATOR	SENSOR	OTX	SOURCE IP	DEST IP
2016-02-17 14:53:46	public_holiday_context:FAILED	0	public_holiday_context	alienvault	N/A	0.0.0.0	0.0.0.0

**Figure 7.8:** Public Holiday Contextual Feed in the OSSIM Web Interface

Figure 7.9 displays the information that the calendar event detector plugin extracts from the appropriate log file. The important information extracted from the log file is the date of the public holiday and the name. This allows OSSIM to run a custom script to take that public holiday date and compares it with the current date. Our custom script returns 'False' when the public holiday date differs from the current one. If the custom script returns 'True', then the OSSIM correlation engine sensitivity is heightened with regards to bad login attempts.



**Figure 7.9:** Event data extracted from a public holiday event

This satisfies our use-case of checking whether employees should be trying to login on days that are public holidays. OSSIM can now alert the necessary personnel when employees are trying to use the company network on public holidays or even on days such as a Saturday or Sunday.



**Figure 7.10:** Event data extracted from a public holiday event

Figure 7.10 is the rule that is added to OSSIM so that it can handle the calendar context data. The rule adds to the current reliability of an attack and it will trigger only once OSSIM receives logs containing public holiday information sent from the public holiday context application. This rule increases the reliability on days that are public holidays so that OSSIM is more sensitive to failed login attempts since there should not be work personnel attempting to login on these days.

### 7.4.2 Limitations of the Calendar Event Information Implementation Testing

A limitation of using calendar information is location. Calendar event information differs greatly across locations, which means that we must have a specific location in our request when using the Google Calendar API. Although the Google Calendar API does allow multiple locations to be specified, this may cause more problems because we would get multiple public holidays with the same date.

This would not be too efficient since OSSIM would have to run the same script multiple times on the same date and additional redundant processing is not recommended. Also using multiple locations would mean that OSSIM would have to have location specific data and apply rules to location specific generated security events. Although this is not impossible, it would require a lot of extra work - and weigh down the value of adding this contextual feed type because more resources would be needed to do the extra configuring.

### 7.4.3 Calendar Event Information Results

The following results are recorded from the test using 10000 randomly generated security events augmented with a calendar events context feed. This means that OSSIM's standard bruteforce attack rules are in place with addition to the embedded public holiday context rules pertaining. This test is repeated with public holiday information in a way that would trigger the system to heighten its sensitivity. There is one level of rule sensitivity linked to whether the current date is a public holiday. This is returned from the public holiday application. The test is run for this sensitivity level to determine whether the new public holiday directives are effective.

Table 7.6 was recorded in this test. Since a non-public holiday would not alter OSSIM's correlation rules, this test is run with a public holiday returned from our public holiday context application. Hence the rule increases the bruteforce directive sensitivity by adding five to the current reliability used to calculate the risk.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	48
2	100000	50
3	100000	46

**Table 7.6:** Table of event number with number of alarms raised during each test with a current day as a public holiday

## 7.5 Terror Threat Level Information Implementation Testing and Results

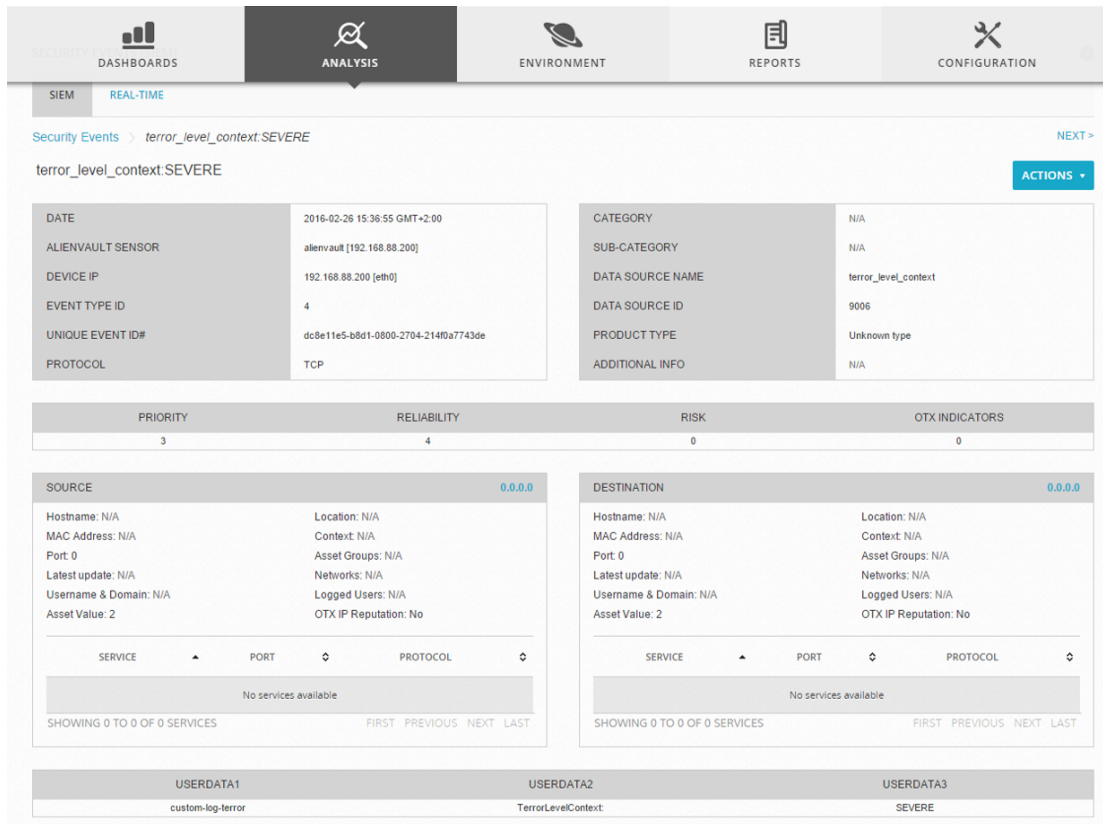
### 7.5.1 Terror Threat Level Information Implementation Testing

Figure 7.11 shows the terror threat level plugin detecting a threat level log file. OSSIM's web interface shows that the terror threat level plugin is functioning properly. See the actual configuration file of the Terror Threat Level detector plugin in Chapter 6 - Implementation.

DATE	EVENT NAME	RISK	GENERATOR	SENSOR	OTX	SOURCE IP	DEST IP
2016-02-26 15:36:55	terror_level_context:SEVERE	0	terror_level_context	alienvault	N/A	0.0.0.0	0.0.0.0
2016-02-26 15:36:35	weather_status_context:EXTREME	1	weather_status_context	alienvault	N/A	0.0.0.0	0.0.0.0
2016-02-26 15:34:35	rsyslog_auth_event:SUCCESS	0	rsyslog_auth_event	alienvault	N/A	0.0.0.0	0.0.0.0
2016-02-26 15:34:35	rsyslog_auth_event:FAILED	0	rsyslog_auth_event	alienvault	N/A	0.0.0.0	0.0.0.0
2016-02-26 15:34:35	rsyslog_auth_event:SUCCESS	0	rsyslog_auth_event	alienvault	N/A	0.0.0.0	0.0.0.0

**Figure 7.11:** Terror Threat Level log displayed on OSSIM's Web UI

Figure 7.12 shows the OSSIM web interface's display of each log file line that it detects to be relevant and the information that it has extracted from that log file via the use of the terror level threat plugin. The important information that this plugin extracts is the actual threat level. This does not need to be fed through a custom script, but rather the severity can be used directly in the rules to heighten OSSIM's correlation engine's sensitivity. The more extreme the severity extracted from the log file, the higher the sensitivity will be raised.



**Figure 7.12:** Event data extracted from a Terror Threat Level contextual feed

The terror threat level information use case is conceptually satisfied by the augmenting of rules relating to brute force attacks. When the threat level RSS feed is updated to a new severity level, OSSIM can change its correlation engine's sensitivity to match. This means that OSSIM is contextually aware of the chance of a cyber attack.

terror threat level low	+1	None	1	ANY	ANY	terror_level_context (9006)	SIDs: 1
terror threat level moderate	+2	None	1	ANY	ANY	terror_level_context (9006)	SIDs: 2
terror threat level substantial	+3	None	1	ANY	ANY	terror_level_context (9006)	SIDs: 3
terror threat level severe	+4	None	1	ANY	ANY	terror_level_context (9006)	SIDs: 4
terror threat level critical	+5	None	1	ANY	ANY	terror_level_context (9006)	SIDs: 5

**Figure 7.13:** Rules added to OSSIM for the Terror Threat Context

Figure 7.13 shows the rules added to OSSIM to enable it to handle the events sent through by the terror threat level contextual application. Each rule is labelled appropriately with its terror threat level. As the terror threat level increases, the corresponding rule increases

the attack reliability. These rules directly relate to the terror threat levels possible from the RSS feed.

### 7.5.2 Limitations of the Terror Threat Level Information Implementation Testing

The use of the terror threat level to give OSSIM a contextual awareness has two limitations worth noting. Since an RSS feed is used to monitor the level of the terror threat of the British Government, or any government that offers this service, the RSS feed's URL needs to be known before hand. This is also going to be linked with the location of the network that OSSIM is monitoring. Unfortunately however not every government has a RSS feed linked with terror threat levels. Hence if a country or location does not have a terror threat level, this contextual data feed type will be rendered useless.

### 7.5.3 Terror Threat Level Information Results

The following results are recorded from the test using 10000 randomly generated security events augmented with a terror threat level context feed. This means that OSSIM's standard bruteforce attack rules are in place in addition to the terror threat level context rules. This test is repeated with varying terror threat level information, in order to trigger the system to heighten its sensitivity. Due to the varying levels of rule sensitivity linked to the different levels of Terror Threat returned from the terror threat level application, the test is run for each sensitivity level to determine whether the new terror threat level directives are effective.

Table 7.7 was recorded with a terror threat level of low. In this case our rule sensitivity is only increased by one because OSSIM recognises that the terror threat level is only low as received by the terror threat level application.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	38
2	100000	36
3	100000	35

**Table 7.7:** Table of event number with number of alarms raised during each test with a low terror threat level

Table 7.8 was recorded with a terror threat level of moderate. In this case our rule sensitivity is also only slightly increased by two because OSSIM recognises that the terror threat level is moderate as received by the terror threat level application.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	40
2	100000	39
3	100000	35

**Table 7.8:** Table of event number with number of alarms raised during each test with a moderate terror threat level

Table 7.9 was recorded with a terror threat level of substantial. In this case our rule sensitivity is increased by three because OSSIM recognises that the terror threat level is substantial as received by the terror threat level application.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	39
2	100000	40
3	100000	42

**Table 7.9:** Table of event number with number of alarms raised during each test with a substantial terror threat level

Table 7.10 was recorded with a terror threat level of severe. In this case our rule sensitivity is increased by four because OSSIM recognises that the terror threat level is severe as received by the terror threat level application.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	43
2	100000	44
3	100000	46

**Table 7.10:** Table of event number with number of alarms raised during each test with a severe terror threat level

Table 7.11 was recorded with a terror threat level of critical. In this case our rule sensitivity is increased by five because OSSIM recognises that the terror threat level is critical as received by the terror threat level application.

<u>Attempt No.</u>	<u>Number of Events</u>	<u>Alarms Raised</u>
1	100000	49
2	100000	52
3	100000	48

**Table 7.11:** Table of event number with number of alarms raised during each test with a critical terror threat level

# Chapter 8

## Analysis of Results

In this chapter the results that have been observed and recorded during the testing phase are analysed. Through the implementation described in Chapter 6, along with the design documented in Chapter 5, the results are carefully recorded and then discussed. This results chapter aims to show that the proposed use of contextual data feeds augmented with an open source SIEM system, OSSIM, will improve the accuracy of the system's detection capabilities.

The hypothesis is split into two parts. The first part aims to prove that it is in fact possible to add contextual data feeds to an open source SIEM in such a way that it is useful - otherwise known as a proof of concept. The second part aims to prove that adding these contextual data feeds can improve OSSIM's functionality - this is why tests were necessary.

It should be noted that the number of alarms is not relevant. The number of alarms is the metric that we use to compare OSSIM's detection capabilities. An increase in alarms would indicate that OSSIM is reaching its risk threshold more often. Hence OSSIM's rule set is adjusting to the current set of events being run through it. It is not adjusting on its own however, so we can attribute this adjustment to the addition of the context information based rules.

### 8.1 Common Proof of Concept Results in each Contextual Data Feed Implementation

The proof of concept results show that the addition of contextual data feeds to OSSIM was successful. Using the *rsyslog* functionality, it was possible to have an external contextual data feed application run on a system on the network and send through specific contextual data. The contextual data is received in a log file format and saved in individual log files on the OSSIM system - these individual log files are named after their contextual data type.

The use of *rsyslog* allows that the contextual data applications can be run at any time without the risk that some contextual data would be missed because *rsyslog* is set to monitor certain local log files - these are the log files generated by our contextual data application. The facility known as *logrotate* helped produce positive results because it allows us to state

the frequency in which our log files are rotated or removed - this is important because the collection of extra log files bring with it the danger of unnecessary disk space usage.

By using *logrotate*, we ensure that the contextual data log files are kept for the necessary time only. Since context relates to a state of something at a certain time, it isn't necessary that we keep contextual data logs longer than a few days. This means that implementing contextual data feeds adds to the requirement of more available disk space on OSSIM, but because of *logrotate* the requirement is much less. This is especially important when considering that some of the contextual data feeds would produce more log file data than others - namely public holiday information would result in larger log files than the threat level information.

## 8.2 Social Media Results Analysis

In section 7.2.3, the social media results are presented. This section discusses those results. The increased sensitivity of the correlation engine by increasing the reliability does allow OSSIM to adapt depending on the data feed into OSSIM by the Social Media application.

Table 7.2 presents the results of a 'LOW' social media context level. It is important to understand that these levels represent the number of results returned from a Social Mention API query using the search term 'brute force'. The 'LOW' level means that the number of results return was very little and this is important because the attack is not trending. Many attackers use social platforms to boast or brag about attacks or even discuss them because of the anonymity.

Simply using the 'brute force' term allows us to see whether it is an attack that is being discussed and potentially planned. The 'LOW' level alarms raised were not significantly more than those of the control test. This is a good indication that the rule changes linked with the 'LOW' level, increase OSSIM's sensitivity to the brute force attack's pattern by only a small amount. In table 7.3, the 'MEDIUM' social media context level results are presented. The 'MEDIUM' results show a larger increase in alarms raised by OSSIM. This is a good sign because 'MEDIUM' social media context level means that the attack is being discussed more than 'LOW' and hence OSSIM should be more sensitive to the brute force attack pattern.

Lastly, table 7.4 presents the 'HIGH' social media context level. The number of alarms raised during these tests is significantly higher than the control test, and also higher than the 'MEDIUM' social media context level. These results show that OSSIM is adapting its brute force attack pattern reliability, to be more sensitive to this attack pattern because it is being discussed, with an alarming frequency, on social platforms.

These results show that the use of Social Media (in this context and manner), can be useful in giving OSSIM the ability to use context to its advantage. It is also clear, by compar-

ison, that the increased reliability based on the context, works effectively to alter OSSIM's correlation engine's sensitivity. As the reliability of the events increased, the number of alarms increased too.

### 8.3 Meteorological Results Analysis

The single set of tests that were run for this extreme weather context state shows that the increased reliability (plus five) effectively makes the correlation engine more sensitive to the brute force attack attempts. The increased reliability is done through the addition of the context based rule set directive.

Table 7.5 presents the results of the meteorological tests where the weather status context application returns an 'EXTREME' weather state. This test helped us confirm that weather status context can help OSSIM. In the situation that weather is deemed extreme, such as a hurricane, a network would have much less traffic because less people would be at work using the network. This applies to login attempts.

Table 7.5 shows that the presence of the weather status rules in OSSIM with the combination of our weather status application returning an 'EXTREME' weather status, increases OSSIM's sensitivity to the brute force attack pattern. The table clearly shows an increase in alarms over the control tests' results. Since no personnel should be at work, OSSIM's risk threshold is reached more often because the reliability of the brute force attack is increased by a noticeable amount. This amount is seen in the rule, plus 5 to the existing reliability of the brute force attack pattern when an 'EXTREME' weather status context state is returned.

Table 7.5 shows an increased number of alarms over the control tests which were using no additional context information. This then proves that the meteorological context feed can give OSSIM the ability to adapt its sensitivity based on the weather state returned from the weather context application.

### 8.4 Calendar Events Results Analysis

The single set of tests that were run for this calendar event context as a public holiday shows that the increased reliability effectively makes the correlation engine more sensitive to the brute force attack attempts.

In table 7.6, the average number of alarms raised is more than that of the control test. This shows that OSSIM can be configured to use public holidays as a context in determining whether its correlation engine should be more sensitive to failed login events. Once OSSIM detects that the public holiday context application has returned a list of dates, one of which coincides with the current date, OSSIM has the contextual awareness that the current day is a public holiday.

OSSIM raises its reliability of predicting a brute force attack pattern, because the underlying circumstance of a public holiday determines that little activity on the network. The social media context rule that is embedded in OSSIM's standard brute force attack pattern rule would cause OSSIM to raise the reliability of a brute-force attack by 5, which is the same increase used in the meteorological context tests for an 'EXTREME' event. This is why the results are quite similar.

On days that are not public holidays, the results are very similar to the control test. This is a good sign because it means the new calendar event rule does not interfere with OSSIM's standard operation. This proves that OSSIM will adjust its sensitivity according to the calendar event context when it is necessary, otherwise it will run as a standard OSSIM installation.

## 8.5 Terror Threat Level Results Analysis

The multiple sets of tests that were run for these terror threat level contexts show that the increasing of reliabilities, directly relating to the increasing terror threat levels, makes the correlation engine more sensitive to the brute-force attack attempts. As each terror threat level gets worse, the reliability is raised to ensure that OSSIM uses this context effectively.

In table 7.7, the results of testing when the terror threat level is low are presented. This level of terror threat causes OSSIM's reliability of a brute force attack to increase by only one. This is why the number of alarms raised is only slightly higher than the control test's results. The low terror threat level is the lowest terror threat level so this rule would actually always affect OSSIM's sensitivity. In table 7.8, the results of testing when the terror threat level is moderate are presented. This level of terror threat causes OSSIM's reliability of a brute force attack to increase by two.

Again, the number of alarms returned is only slightly higher than the previous terror threat level test of low. In table 7.9, the results of testing when the terror threat level is substantial are presented. In table 7.10, the results of testing when the terror threat level is severe are presented. In table 7.11, the results of testing when the terror threat level is critical are presented. It is clear by the alarms raised in each subsequent terror threat level test that as the terror threat level is elevated, the number of alarms raised is slightly elevated too.

Much like the tests for terror threat level low and moderate, the tests for terror threat level substantial, severe and critical all have embedded rules within OSSIM's standard brute force attack rule set which increase the reliability of predicting a brute force attack accordingly. If we take the results from table 7.11 where OSSIM's reliability is increased to match the terror threat level of critical, the number of results returned is similar to that of the weather status context test condition of 'EXTREME'. This is not a random occurrence, but rather the result of each context's rule increasing OSSIM's reliability by the same amount of five.

As you can see in the above tables of results, as the severity of threat level increases, so does the number of alarms raised by OSSIM's correlation engine. This proves that OSSIM can use this type of contextual data feed to manipulate its interpretation of the other incoming events.

# Chapter 9

## Conclusion

This final chapter will briefly outline the steps in the process of compiling the data, and drawing conclusions based on the results. A discussion of the results acquired through the implementation and tests of multiple contextual applications streaming relevant results to OSSIM will show that adding contextual data to a SIEM context can greatly support the 'decision-making' process that the SIEM goes through. Each of the above chapters will be summarised into a concise evaluation of why this thesis succeeds in showing how and why contextual information is an important resource for cyber security.

### 9.1 Summary Overview

The cyber security industry is constantly adapting and evolving threat and intrusion detection to ensure that digital information is safe. As networks and connectivity grow throughout the world, the security of these networks needs to become a priority, as well as a critical infrastructure, considering the sensitive nature of the information on these networks.

The importance of the cyber security of enterprises has grown into a massive industry because enterprise networks contain sensitive digital information as well as allow many different hierarchies of users to access that data. This means that there are potential vulnerabilities throughout the network hence a need to have complete visibility of that network.

Security information and event management systems collect data from devices throughout your network, parse that data into readable and useful information, stores that information and runs the events through a complex correlation engine. The SIEM presents its findings through a web terminal that is human readable.

The SIEM used in this thesis is the industry de facto open source SIEM known as OSSIM. OSSIM is maintained and developed by Alienvault. The hypothesis,

*The augmenting of security event data with contextual data will improve a Security Information and Event Management System's threat detection capabilities.*

can be split into two separate test cases: proof of concept and tests. The proof of concept case aimed to show that it was possible to integrate external contextual data applications with OSSIM in such a way that it would better OSSIM's performance. In this sense, performance is based on its detection capabilities. The tests aimed to show that using generated events would yield positive results in the sense that OSSIM would adapt appropriately to the incoming contextual information.

Further research into SIEMs showed that to optimise the workflow of such a complex collection of tools, careful consideration was needed when adding extra components. Contextual data applications needed to be seamlessly integrated into OSSIM. SIEMs go through the process of collecting information from the surrounding devices on the network, normalising and parsing this data into events, recognising the events and the data contained in each event and using that meta data in a correlation engine.

With this process in mind, the contextual data applications were developed in such a way that would minimise the work load on OSSIM's side, hence the log file formats sent to OSSIM from the context applications were concise and accurate. The chosen contextual data types are tabularised in Table 9.1.

<u>No.</u>	<u>Context Type</u>	<u>Context Test Case</u>
1	Social Media	Social Mention Result Number
2	Meteorological	Weather Result Severity
3	Calendar Events	Public Holiday Event
4	Terror Threat Level	British Gov. Terror Threat Level

**Table 9.1:** Context types used for the Contextual Data Applications

These contextual data types were chosen because they showed a variety of use cases - each different from the next. Each context type used a different openly available medium of information and harnessed this information into OSSIM useful information. The importance of contextual data in general is discussed indepth throughout this thesis. In summary, contextual information gives an insight into an event by providing extra information regarding the event. In many cases, context can motivate a decision entirely. As discussed, humans rely heavily on contextual cues in conversation or decision making by taking in information around the decision and using the information to better understand the decision itself.

This is the goal of using contextual information with OSSIM. By adding contextual information, we are giving OSSIM more information surrounding the events that it must use in its correlation engine. The correlation engine will use this extra information to adapt how

it receives the usual information, and make more accurate decisions. The design to ensure that the information would be useful to OSSIM will be well documented in chapter 5. The design process is split into a number of components to help with readability and modularity. OSSIM was set up on a local network and kept constant with a static IP address and access to the internet. OSSIM was run inside of an Oracle VM VirtualBox because it runs as its own system and hence requires its own environment and resources.

The next design considerations were those of the development of the contextual applications. The information retrieved by each application needed to be useful and usable. This meant that reliable APIs were to be used. Table 9.2 is an overview of the contextual data APIs:

<u>No.</u>	<u>Context Type</u>	<u>Context API</u>	<u>Description</u>
1	Social Media	Social Mention API	This API allowed a search for key words across a variety of social media including social networks and news reports.
2	Meteorological	OpenWeatherMap API	This API allowed for a query for weather in a certain area
3	Calendar Events	Google Calendar API	This API allowed for a query of public holidays in a certain country
4	Terror Threat Level	British Gov. Terror Threat RSS Feed	This RSS feed returns the current terror threat level

**Table 9.2:** Context type APIs used for the Contextual Data Applications

Once the context application retrieved the information, it was parsed and written to a log file. The log file formats are documented in chapter 5. These formats were designed to be easily readable by OSSIM once the log file was sent through the Rsyslog functionality set up to relay all new context application log files.

Lastly, the rulesets needed to be designed in such a way as to optimise OSSIM when using them. Rulesets were copied from existing ones and altered to use the contextual information. This is the recommended process for creating new rules for OSSIM. The rules were designed to fit the test use cases for each contextual data. The test use case for each contextual data is tabularised in Table 9.3.

<u>No.</u>	<u>Context Type</u>	<u>Context Test Use Case</u>	<u>Description</u>
1	Social Media	Trending Attack	The number of results of a trending attack will determine the sensitivity of the OSSIM correlation engine in dealing with the trending attack's rules
2	Meteorological	Extreme weather	The current weather being extreme or not will determine the sensitivity of the OSSIM correlation engine in dealing with attacks in that location
3	Calendar Events	Public Holidays	The current date being a public holiday or not will determine the sensitivity of the OSSIM correlation engine in dealing with attacks on that date.
4	Terror Threat Level	Terror Threat Severity	The current terror threat level will determine the sensitivity of the OSSIM correlation engine in dealing with any attacks

**Table 9.3:** Context test use cases used for the Contextual Data Applications

The implementation process is documented through chapter 6. The implementation process is closely tied to the testing because of the previously mentioned parts of the hypothesis. The implementation reflects a methodology focused on showing that the addition of contextual data application feeds is both feasible and advantageous. This is discussed in the next section.

## 9.2 Discussion of Proof of Concept

The implementation process detailed the setting up of the testing environment including OSSIM, Rsyslog and the control test directives. These control test directives are used as part of the control tests discussed later. The implementation of each contextual data application takes each contextual data API and implements it in such a way that the application will retrieve the required contextual information. Once the data is retrieved, the application writes the data to a log file in a particular format readable by OSSIM.

To show that we have proved the proof of concept, the contextual data application must link cleanly to OSSIM. Once this link is made, OSSIM must interpret the data from the contextual data application in an applicable manner and use the data for OSSIM's correlation engine. Enabling OSSIM to interpret the incoming data begins with developing a detector plugin for OSSIM's agent. This detector plugin is enabled for each contextual data type and recognises the data as it is sent through from the Rsyslog. The detector plugin defines the regular expression used to interpret the data so that the log data can be normalised into variables used in the correlation engine.

The testing process is documented in chapter 7. As you see in chapter 7, the contextual data feeds are successfully linked to OSSIM. Each contextual data type has a working detector plugin that uses the log file data generated by the application. The detector plugin parses that data correctly into variables that the correlation engine uses in its directives. The attack method, brute force logins, has its correlation directive altered with an embedded check of each contextual data response. This check returns whether the reliability of the current directive should be increased or not. This increase is based on the type of result returned from the contextual data feed.

This proof of concept shows that the addition of external contextual data feeds is possible. The next section looks at why it is advantageous. In conclusion, the augmenting of contextual data with security event data is implementable if the correct data is fed into the SIEM and the correct data formats defined in the detector plugins. It has also been shown that different types of contextual data can be used to augment security event data. The limitations of linking contextual data feeds to OSSIM are also stated in the testing chapter.

## 9.3 Conclusion of Results

The results are presented in chapter 8. The results are separated into the different tests done for each contextual data type and the control test. The control test was done to ensure that we have results for the number of alarms triggered before we add the contextual data feeds to OSSIM. In summary, each contextual data type is run with 10 000 events three times. Since the events are generated randomly, the use of multiple test ensures a fair test.

Table 9.4 summarises the results from the tests.

<u>Context Type</u>	<u>Average Alarms Raised During Testing</u>
Baseline	37
Social Media (Low)	37.6
Social Media (Medium)	41.3
Social Media (High)	48.6
Meteorological (Extreme)	47.6
Calendar Events (Public Holiday)	48
Terror Threat Level (Low)	36.3
Terror Threat Level (Moderate)	38
Terror Threat Level (Substantial)	40.3
Terror Threat Level (Severe)	44.3
Terror Threat Level (Critical)	49.6

**Table 9.4:** Results showing the average number of alarms from each section of contextual data

The second part of the hypothesis, namely whether the augmenting of the contextual data feeds with event data is useful, still remains to be shown. The following discussion will show that the above set of results from the tests show this part of the hypothesis to be true.

The first noteworthy observation is that of the results from the control test, the social media (low) and the terror threat level (low) tests. Noticeably, the results are very similar, in 'alert average'. This is because the low state increases the sensitivity by none and one respectively. It makes sense that the low state from the respective contextual data feeds would alter OSSIM's sensitivity by only a small amount.

The next observation is that of the results from the high states such as social media (high), meteorological (extreme), calendar events (public holiday) and terror threat level (critical). These results have each returned alarm averages that are very close in amount. It makes sense that the high states of the contextual data feeds would yield similar results because their effect on OSSIM's sensitivity is similar (around plus five to the reliability of the attack). The results are substantially higher than the low states. Again this is a positive result because we would not want the low state to cause OSSIM to raise its sensitivity as

much as the high state.

Finally we will discuss the results as a whole. Throughout each contextual data feed, as the severity of the data feed increases, so does the number of alarms. This shows that as severity increases as returned from our contextual data feeds, so does OSSIM's sensitivity. This means that OSSIM now has the ability to use contextual data to adjust its correlation engine when interpreting incoming security events. Although this is a limited example of augmentation of contextual data with security event data, it proves to show a SIEM can use the additional, none security related information, to effectively increase its visibility of its environment.

OSSIM is essentially using surrounding information, augmenting it with the usual security information and making more accurate and appropriate decisions regarding a relevant attack. In conclusion, the hypothesis of effectively augmenting contextual data with security information has been shown to be true. This thesis shows that OSSIM can be configured and developed to use multiple contextual data feeds to supplement its visibility of the surrounding network.

## 9.4 Limitations to Findings

A mention of some limitations related to the above findings is necessary. The main limitation is that of false positives and what it means for these results. False positives are alerts that are not actual attacks. False positives can be produced by a number of methods such as trying the wrong password or username incorrectly too many times. In this thesis we increase the sensitivity of the OSSIM correlation engine with respect to certain correlation rules.

This increase in sensitivity could lead to the generation of more alerts in general and hence more false positives too. This is important to acknowledge because the increase number of alerts does not necessarily mean an increased number of true positives. However, this thesis attempts to use contextual data to help adapt the SIEM with situational awareness. The increased number of alerts when the contextual data warrants an increased correlation sensitivity shows that the SIEM can use contextual data effectively.

False positives also lead to alert fatigue which, in this ages of millions of logs and alerts, is a real issue that security professionals need to consider. To this end, a possible continuation of this thesis could be to check the validity of the alerts being produced to ensure that we are not producing an increased amount of false positives and hence not contributing to alert fatigue.

# Bibliography

- [1] Heartbleed.com, *Heartbleed*, 2016. [Online]. Available: <http://heartbleed.com/>.
- [2] D. Kushner, *The real story of stuxnet*, 2016. [Online]. Available: <http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet>.
- [3] A. e. a. Di Sarno C. Garofalo, “A novel security event and information management system for enhancing cyber security in a hydroelectric dam”, University of Naples, Computer Science Research Group, Institute for Informatics and Telematics, Paper, 2016.
- [4] A. Kibirkstis, *What is the role of a siem in detecting events of interest*, 2009. [Online]. Available: <https://www.sans.org/security-resources/idfaq/what-is-the-role-of-a-siem-in-detecting-events-of-interest/5/10>.
- [5] M. Rouse, *Security information and event management*, 2012. [Online]. Available: <http://searchsecurity.techtarget.com/de%0Cnition/security-information-and-event-management-SIEM>.
- [6] P. Bedwell, “Finding a new approach to siem to suit the sme environment”, AlienVault, Paper, 2014.
- [7] S. Jenkins, “Learning to love siem”, Q1 Labs, Paper, 2011.
- [8] Verizon, *2016 data breach investigations report*, 2016. [Online]. Available: <http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016>.
- [9] C. Constantine, “Big data: An information security context”, *Network Security*, vol. 2014, no. 1, pp. 18–19, 2014.
- [10] P. Dowd and J. McHenry, “Network security: It’s time to take it seriously”, *Computer*, vol. 31, no. 9, pp. 24–28, 1998. DOI: 10.1109/2.708446.
- [11] B. Daya, “Network security: History, importance, and future”, *University of Florida Department of Electrical and Computer Engineering*, 2013.
- [12] O. Adeyinka, “Internet attack methods and internet security technology”, in *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, IEEE, 2008, pp. 77–82.
- [13] J. Inns, “The evolution and applications of siem systems”, Accumuli, Paper, 2014.
- [14] Satisnet, *New Year, New Threats for Q1 Labs to fix*. 2012. [Online]. Available: <http://www.satisnet.co.uk/content/new-year-new-threats-for-q1-labs-to-fix/>.

- [15] J. Sweeney, "Creating your own siem and incident response toolkit using open source tools", SANS Institute, Paper, 2011.
- [16] P. Dairinram, D. Wongsawang, and P. Pengsart, "Siem with lsa technique for threat identification", *2013 19th IEEE International Conference on Networks (ICON)*, 2013. DOI: 10.1109/icon.2013.6781951.
- [17] E. e. a. Suarez-Tangil G. Palomar, "Providing siem systems with self adaptation", Carlos III University of Madrid, Paper, 2013.
- [18] I. Kotenko and A. Chechulin, "Attack modeling and security evaluation siem systems", St. Petersburg Institute for Informatics and Automation, Paper, 2012.
- [19] AlienVault. (2015). Ossim vs usm, [Online]. Available: <https://www.alienvault.com/products/ossim>.
- [20] K. Kavanagh and M. e. a. Nicolett, "Magic quadrant for security information and event management", Gartner, Paper, 2016.
- [21] IBM, *Ibm security qradar siem*, 2016. [Online]. Available: <http://www-03.ibm.com/software/products/en/qradar-siem>.
- [22] H. ArcSight, *Arcsight enterprise security management*, 2016. [Online]. Available: <http://www8.hp.com/us/en/software-solutions/arcsight-esm-enterprise-security-management/index.html>.
- [23] C. Pham, *From Events to Incidents*. 2001. [Online]. Available: <http://www.sans.org/reading-room/whitepapers/incident/events-incidents-646>.
- [24] G. Cater, *Security event management*, 2016. [Online]. Available: [http://www.infosectoday.com/Articles/Security\\_Event\\_Management/Security\\_Event\\_Management.htm](http://www.infosectoday.com/Articles/Security_Event_Management/Security_Event_Management.htm).
- [25] P. Czanik, *The 6 categories of critical log information*, 2006. [Online]. Available: <http://www.sans.edu/research/security-laboratory/article/sixtoplogcategories>.
- [26] I. Eaton, *The ins and outs of system logging using syslog*, 2003. [Online]. Available: <http://www.sans.org/reading-room/whitepapers/logging/logging-ins-outs-system-logging-syslog-1168>.
- [27] F. Essalmi and L. Ayed, "Graphical uml view from extended backus-naur form grammars", in *Advanced Learning Technologies, 2006. Sixth International Conference on*, IEEE, 2006, pp. 544–546.
- [28] D. Howell, "Building better data protection with siem", ManageEngine, Paper, 2015.
- [29] iSightPartners, *What is cyber threat intelligence and why do i need it?*, 2014. [Online]. Available: [http://www.isightpartners.com/wp-content/uploads/2014/07/iSIGHT\\_Partners\\_What\\_Is\\_20-20\\_Clarify\\_Brief1.pdf](http://www.isightpartners.com/wp-content/uploads/2014/07/iSIGHT_Partners_What_Is_20-20_Clarify_Brief1.pdf).
- [30] W. Gragido, *Understanding indicators of compromise (ioc) part i - speaking of security - the rsa blog and podcast*, 2012. [Online]. Available: <https://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/>.
- [31] S. Lynch, *Reinventing threat intelligence*, 2014. [Online]. Available: <http://resources.infosecinstitute.com/reinventing-threat-intelligence/>.

- [32] Openioc.org, *The openioc framework*, 2015. [Online]. Available: <http://www.openioc.org/>.
- [33] Cybox.mitre.org, *Cybox -cyber observable expression?*, 2015. [Online]. Available: <http://cybox.mitre.org/index.html>.
- [34] Stix.mitre.org, *Stix -structured threat information expression?*, 2015. [Online]. Available: <http://stix.mitre.org/>.
- [35] K. Maxwell, *Open source threat intelligence*, 2013. [Online]. Available: [https://digital-forensics.sans.org/summit-archives/DFIR\\_Summit/Open-Source-Threat-Intelligence-Kyle-Maxwell.pdf](https://digital-forensics.sans.org/summit-archives/DFIR_Summit/Open-Source-Threat-Intelligence-Kyle-Maxwell.pdf).
- [36] ReliaQuest, *Siem optimization 101*, 2014. [Online]. Available: [http://cdn2.hubspot.net/hub/247765/file-344637836-pdf/SIEM\\_Optimization\\_101\\_E-book\\_UPDATED.pdf](http://cdn2.hubspot.net/hub/247765/file-344637836-pdf/SIEM_Optimization_101_E-book_UPDATED.pdf).
- [37] I. Kotenko and A. Chechulin, "A cyber attack modeling and impact assessment framework", St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Paper, 2013.
- [38] S. Camtepe and B. Yener, "A formal method for attack modeling and detection", Rensselaer Polytechnic Institutes, Journal, 2006.
- [39] A. Moore, R. Ellison, and R. Linger, "Attack modeling for information security and survivability", DTIC Document, Tech. Rep., 2001.
- [40] J. Dawkins, C. Campbell, and J. Hale, "Modeling network attacks: Extending the attack tree paradigm", in *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, 2002, pp. 75–86.
- [41] A. Goncharov, *Who needs event normalisation?*, 2013. [Online]. Available: <http://www.metanetivs.com/event-normalization/>.
- [42] R. Marty, *Event processing - normalisation*, 2007. [Online]. Available: <http://www.raffy.ch>.
- [43] AlienVault, *Plugins, sid's and log normalisation*, 2014. [Online]. Available: <https://alienvault.bloomfire.com/posts/520572-plugins-sid-s-and-lognormalization/public>.
- [44] A. Lane, *Understanding and selecting a siem/lm: Correlation and alerting*. 2010. [Online]. Available: <https://www.securosis.com/blog/understanding-andselecting-a-siem-lm-correlation-and-alerting>.
- [45] Praetorian, *Regulatory compliance*, 2015. [Online]. Available: <http://www.praetorian.com/regulatory-compliance>.
- [46] A. Networks, *Arbor networks' finds 36% increase in advanced persistent threats*, 2014. [Online]. Available: <http://www.arbornetworks.com/news-and-events/press-releases/2014-pressreleases/%205111-arbor-networks-research-finds-36-increase-in-advanced-persistent-threatsand-%20attacks-against-mobile-networks-doubled>.

- [47] A. Chuvakin, *Siem real-time and historical analytics collide?*, 2014. [Online]. Available: <http://blogs.gartner.com/anton-chuvakin/2014/07/30/siem-real-time-andhistorical-%20analytics-collide/>.
- [48] J. van de Moosdijk and D. Wagenaar, "Addressing siem", 2015.
- [49] E. Kostrecová and H. Bínová, "Research paper security information and event management", *Management*, vol. 4, no. 2, 2015.
- [50] R. Chow and P. Golle, "Faking contextual data for fun, profit, and privacy", in *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, ACM, 2009, pp. 105–108.
- [51] J. Hong, E. Suh, and S. Kim, "Context-aware systems: A literature review and classification", *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.
- [52] A. Dey, "Understanding and using context", *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [53] W. Jung, L. Olfman, T. Ryan, and Y. Park, "An experimental study of the effects of contextual data quality and task complexity on decision performance", in *Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.*, IEEE, 2005, pp. 149–154.
- [54] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems", *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, p. 24, 2012.
- [55] U. Alegre, J. Augusto, and T. Clark, "Engineering context-aware systems and applications: A survey", *Journal of Systems and Software*, vol. 117, pp. 55–83, 2016.
- [56] P. Dourish, "What we talk about when we talk about context", *Personal and ubiquitous computing*, vol. 8, no. 1, pp. 19–30, 2004.
- [57] D. Modha, R. Ananthanarayanan, S. Esser, A. Ndirango, A. Sherbondy, and R. Singh, "Cognitive computing", *Commun. ACM*, vol. 54, no. 8, pp. 62–71, Aug. 2011, ISSN: 0001-0782. DOI: 10.1145/1978542.1978559. [Online]. Available: <http://doi.acm.org/10.1145/1978542.1978559>.
- [58] S. Greenberg, "Context as a dynamic construct", *Human-Computer Interaction*, vol. 16, no. 2, pp. 257–268, 2001.
- [59] K. Henriksen, *A framework for context-aware pervasive computing applications*. University of Queensland Queensland, Australia, 2003.
- [60] P. Hu, J. Indulska, and R. Robinson, "An autonomic context management system for pervasive computing", in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on.*, IEEE, 2008, pp. 213–223.
- [61] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness", in *Handheld and ubiquitous computing*, Springer, 1999, pp. 304–307.

- [62] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey", *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 414–454, 2014.
- [63] B. Hardian, J. Indulska, and K. Henricksen, "Balancing autonomy and user control in context-aware systems-a survey", in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, IEEE, 2006, 6–pp.
- [64] L. Barkhuus and A. Dey, "Is context-aware computing taking control away from the user? three levels of interactivity examined", in *UbiComp 2003: Ubiquitous Computing*, Springer, 2003, pp. 149–156.
- [65] H. Stuart, *Uk threat level lowered to substantial*, 2011. [Online]. Available: <http://henryjacksonsociety.org/2011/07/12/uk-threat-level-lowered-to-substantial/>.
- [66] AlienVault, *Process: Modifying a built-in directive*, 2016. [Online]. Available: <https://www.alienvault.com/documentation/usm-v5/correlation/process-modifying-built-in-directive.htm>.
- [67] ———, *Building collector plugins - admin guide*, 2010. [Online]. Available: <http://docplayer.net/1990582-Building-collector-plugins-admin-guide.html>.