BYOD Security Enhancement using Log Correlation in Corporate Environments

A dissertation submitted for the Degree of Master of Science in Information Security

G. Y. C. L. Gunaratne
University of Colombo School of Computing 2017

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by any another person, except as acknowledged in the test.

G. Y. C. L. Gunaratne

Signature Date: 09/03/2017

This is to certify that this thesis is based on the work of

G. Y. C. L. Gunaratne

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certify by:

Dr. Kasun De Zoyza

Signature Date: 09/03/2017

Acknowledgements

I take this opportunity to express my heartiest gratitude to who helped and contributed to bring this project to greater heights. First and foremost I would like to thank all the academic and administrative staff of University of Colombo School of Computing (UCSC) for the support given when necessary resources were required to complete this project.

Secondly, I would like to extend my gratitude to the lecturer-in-charge, Dr. Manjusri Wickramasinghe who gave me proper guidance other resources when necessary in order to complete the projects successfully.

Also, my heartfelt gratitude goes to my project supervisor Dr. Kasun De Zoysa for the guidance, immense support given and for the ideas given in all stages of the project right though from very beginning to the end to bring this project to the next level.

Further, I would like to express my sincere gratitude for all the staff, students for the valuable ideas, comments, criticisms, opinions and suggestions given during in the early stages of the project and even after the project was commenced to get this project in to greater heights.

Finally, I extend my thanks to my parents, friends, and all who rendered their thoughts, patience and co-operation which motivated me intellectually to reach greater heights.

Abstract

Bring Your Own Device (BYOD) is a concept in information technology that has become a prominent topic in the recent past. This concept continues to gain popularity due to its ability to give mobility and flexibility to IT operations in organizations. However, due to the rapid growth in the usage of mobile devices in corporate environments, many security concerns and risks have risen, which can easily compromise the business information and cause IT processes in organizations to malfunction. In my opinion, the effective use of BYOD can be beneficial to both the organization as well as the employees of that organization. In the one hand, corporate organizations will save money because it is no longer necessary to invest on purchasing electronic devices for every employee. On the other hand, employees will also find more satisfaction in getting their personal devices such as smart phones, tabs and laptops involved in work. However, this would also mean that employees will be able to access corporate information using personal devices, which are not always monitored by the organization. Further, employees may intentionally or unintentionally perform vulnerable activities using the BYOD equipment that can breach the security of organizational information. Such activities may also expose the corporate network and its information and assets to unauthorized parties. Thereby, even though BYOD brings mobility, convenience and more satisfaction to the work environment, the confidentiality and integrity of corporate information will be at risk. Therefore, to gain the maximum benefits from BYOD, special security measures that can ensure the safety of the organization’s information should be implemented. This is the main challenge in using the BYOD concept in corporate organizations. In spite of these challenges, BYOD is adopted in many corporate organizations at present as an accepted and properly defined concept because of its potential to contribute towards the organization’s efficiency. This study will focus on addressing the security concerns that threaten the effective use of this concept in organizations. In this study, I will look into analysing patterns in the traffic generated from BYOD equipment and focus on methods that can mutually relate these facts for suspected activities. The information gathered from this analysis will be helpful to enhance the security of BYOD equipment in corporate environment.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronym	**Definition**
BYOD	Bring Your Own Device
TCP	Transmission Control Protocol
IP	Internet Protocol
SIEM	Security Information and Event Management
AP	Access Point
DoS	Denial of Service
CPU	Central Processing Unit
RAM	Random Access Memory
CIO	Chief information officer
CISO	Chief information officer
MDM	Mobile Device Management
MIM	Mobile Information Management
MAM	Mobile Application Management

1 Introduction

Advancement of the mobile devices such as smart phones, tabs has made people and organisation to categorise these devices as essential devices. Most of these devices which have the capability of accessing internet faster and more reliable, employers are motivated to use these devices in their organisations. Bring Your Own Devices has become a phenomenon in the present where employee connect their personal mobile devices to corporate network of the organization in order to execute their daily business function. This will allow the users or the employees to easily connect their devices from any geographical area. Thus BYOD has helped the employees and employers to gain substantial advantages such as efficiently and flexibly.

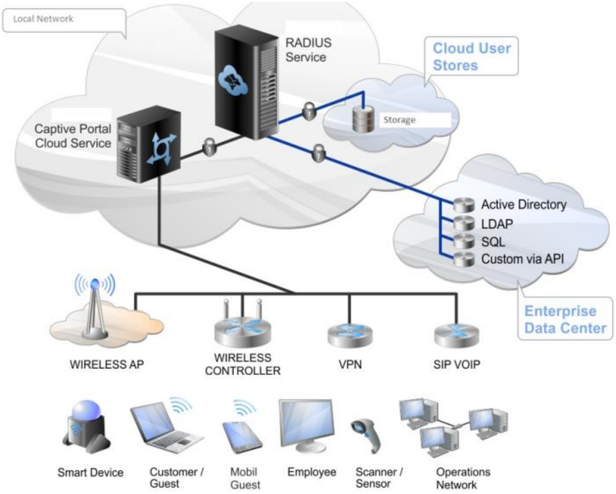


Figure 1 – BYOD in Corporate Network [9]

BYOD is an organisational IT concept that enables employees to access the business related data such as e-mails, intranet, HR systems, meeting schedules and any other business related information using the employee's personally owned devices [1]. Before the organisations absorbed BYOD concept, the employees of the organisations are given a corporate owned devices to perform the business related activities, whereas devices are often managed by IT personnel or by an IT department in the organisation. BYOD concept provides cost effectiveness to the organisations since the devices are not supplied by the organisation to the employee. Therefore, organisations will be benefited by highly productive employees from using their own devices for daily business activities [2].

However, with all the benefits offered by the BYOD concept, there are quite a few security concerns which need to be addressed. In other words BYOD can introduce plenty of risks in terms of IT to the organisation [3]. For example, misplacing a mobile device which does not have a password, configured the organisation email in it may led to compromise of sensitive data owned by the organisation.

1.1 Research Domain

1.1.1 Research Problem

Since there are many devices available in the market, based on the technological factors such as hardware and operating system, people have many choices to purchase these devices at their discretion. These portable devices have become very much closer to peoples' lives nowadays. Some people are immensely attached to their personally own devices and they are very much happy to perform organisational related professional tasks through these devices.

Researchers depict that employee efficiency will rapidly increase when employees' use their personally owned devices for corporate work and access organisational information [4]. Fact is that employees' are well conversant with the system user interfaces and freely available software in the internet.

One of the employer's main expectations is to make employees more productive. An organisation which supports BYOD concept has many advantages than an ordinary organisation which does not support BYOD concept. These devices are owned by users which has high portability that enables the users to work from anywhere in the world if they have been granted with the access. Good thing here is the cost factor. As the user is solely bearing the cost for these devices the organisation need not to be concerned about the finance cost for these devices. However, the organisation has raise its concern about the security measurements and considerations in terms of BYOD.

This is the point where many organisations are failing. Since these devices are personally owned and devices which have high portability, it is practically infeasible to monitor the users in terms of what are they doing using BYOD? Which applications are they accessing? What apps are being installed in these devices? Are they secured apps? Do these devices have anti-virus or security applications? [5] Many questions can be raised by the management to the IT department in terms of BYOD. At this point the organisation can support the users with software protection applications which are cheaper than purchasing portable devices. In the other hand if the organisations need to monitor these devices they can simply implement a Mobile Device Management (MDM) solution and add these devices to MDM under users consent where monitoring can be available to IT department of the organisation [6].

However, with many advantages comes with BYOD concepts in corporate world, it is highly necessary to bring the employers attention to corporate security risks. These risks can be in terms of corporate information being leaked or go to the hands of unauthorised parties and competitors who can get to know about the company information assets such as company employee details, financial records, company strategies etc. This can occur because of malware being installed in devices, vulnerable applications being installed in devices, devices being unattended without applying proper locking mechanism and devices being physically compromised [7]. Considering these factors many organisations will have to update their BYOD policies, security measurements and implementations.

1.1.2 Significance of the Research

This research was inaugurated due to many security flaws being recorded in terms of BYOD concept. At present employees whom recruited are highly technically savvy. Technological understanding and knowledge help individual productivity. Corporate organisations are also set immense pressure to their employees with expectation of high productivity and standards. This is one of the main reasons why BYOD concept is well accepted in corporate environments. Through this concept employees can ease their work using their own personal devices to fulfil the business requirements.

Rapid growth of using personal devices in corporate environments has led to many risks and potential threat in terms of security. Using BYOD devices in corporate environment significantly bring concerns to organisation's network perimeter. Administration part of the devices would not be handled by the organisation. In this case, organisations struggle to build implications and procedures in terms of security and risks since these devices are being used to access business related applications.

However, the problem domain of this research is to address *“BYOD security enhancement using log correlation in corporate environments”.*

1.1.3 Goals and Objectives

- Identify current problems in BYOD in terms of security threats and potential risks within the corporate network and corporate perimeter
- Propose a method to prevent security loopholes. These loopholes will be identified by comparing and correlating BYOD network traffic and log records

1.2 Limitations and Assumptions

One of the main difficulties was to gather BYOD network traffic data to analyse. The scope of this research and solution is limited to small and medium scale organisations.

2 Literature Survey

This chapter articulates the basis of the research project. It also aims to provide a notion of the researches, analyses and cohesive information gathered, which further explains BYOD security concepts, security vulnerabilities of BYOD devices, current BYOD security flaws, corporate information security and BYOD security governance.

2.1 Risk Analysis

Risk analysis provides a study of potential internal and external risks. The section also elaborates the possibilities of mitigating the possible risks which are not inherent. This also provides any interested parties/corporations an analysis and deeper understanding of the feasibility of the application. It is more emphasized towards the governing area of BOYD security where internal staff of the organisation should be aware of managing such threats.

2.1.1 Security Management

The responsibility of assuring information security of an organisation is vested upon the company CIO/CISO. Though it is the responsibility of the Board to oversee the overall strategic direction and the operation of the company, in many instances there is a lack of understanding of business information security.

This, unfortunately, has led to a lack of direction and in turn poor allocation of resources towards information security. If an organisation structure is such, where the executive management is unable to convince the Board of the importance and the need for adequate IT security infrastructure, it is emphasised that the organisation should re-think, cultivating a risk-based culture. If a cultural mismatch exists, it is suggested that the executive management contemplates in improving the communication process between the board and the organisation. This will enable resolving major threats and issues an organisation will have to face with the advancement of the technology.

2.1.2 Mobile Device Management (MDM)

In BYOD concept, MDM [2] is a solution to monitor the status of the mobiles. MDM is also providing the facility of controlling the mobile device remotely. An MDM has two major entities. Those are namely MDM agent and MDM server. The agent which is an application is installed in the mobile devices and it will update the mobile status in the server side. The MDM server is able to execute commands in the mobiles remotely such as lock, erase, encrypt, locate and etc [2]. MDM system is a collaboration of several other components such as MDM server, the gateway server, MDM console, MDM agent and etc. Figure 2 elaborates the general schema of MDM architecture. The main task of the MDM solution is that the agent sends the mobile device data to the MDM server and it will perform the administrative functions in the mobiles remotely.

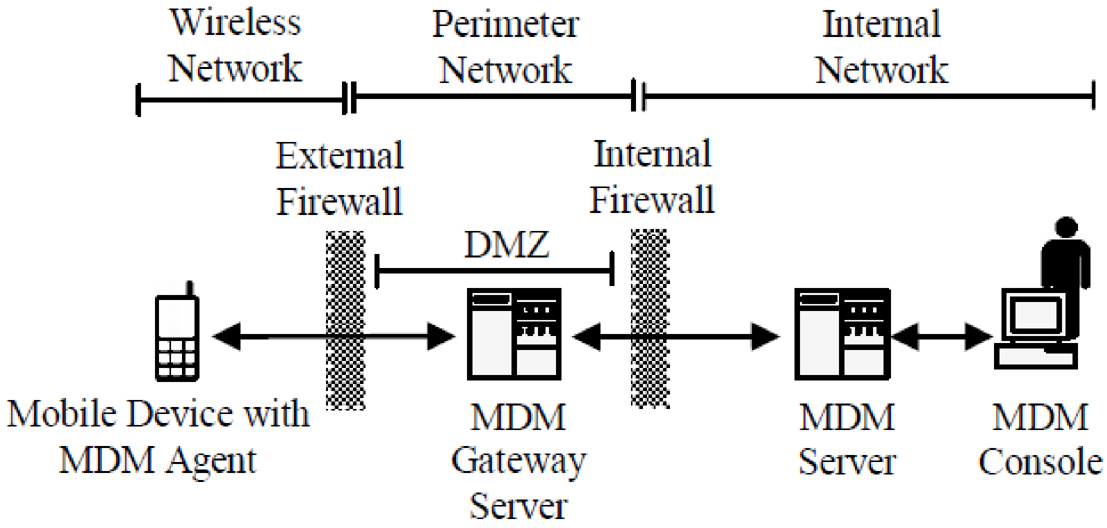


Figure 2 – MDM Architecture [2]

2.1.3 Mobile Information Management (MIM)

MIM is a concept where it secures the corporate information instead of the mobile devices. The basic idea behind this concept is the corporate information being saved in the central location (e.g. cloud environment) and share the information securely among the end devices such as mobile phones and tabs [2]. The MIM is only allowing trusted applications to access the corporate data securely. There is a limitation for these applications which runs in the mobile devices.

2.1.4 Risk of Installing Malicious Applications

The expansion of users' needs and wants has shifted towards customising one's own device in accordance with their personal requirements. These necessities are fulfilled by many application markets such as Apple store, Samsung, Google play, etc. During the application installation process, users are forcefully being asked to provide permission to gallery access, location services access and etc. The users are trading off the security concerns with the benefits they get by the application [8].

This is where the problem arises. When various applications with different levels of security are installed to the same device security risks also increases. E.g. a low level of security application of a free game application and a highly trusted banking application nowadays may exist in any device. The free game application could be malicious [8]. This application has a high probability of modifying , stealing even destroying inter-application messages and, therefore, compromising organisational information security. When an application has been installed in the device, the users are unable to rate the trust of the application. The organisation reputation, secrecy, and the general information security will be directly impacted by those applications. In such case the organisation needs to have a control mechanism on the applications which are being installed on the BYOD devices.

2.1.5 Risk relating to customising BYOD

It is more common at the present the users are using devices with custom configurations. "Jailbreaking", "root", and “unlock” are the most popular custom configuration which eliminates the vendor configuration of the devices. Customised devices are more vulnerable for malware, viruses, malicious applications compared to normal devices. Also when devices are customised, an insecure application could access the resources of the device without a prior approval by the user. E.g. the application can access microphone, camera and storage where it may contain sensitive details of the organisation [8]. This may lead the organisation to deal with a security incident.

2.1.6 Advanced Persistent Threats

An Advanced Persistent Threat (APT) [12] means a network attack prolonged undetected for a considerable period of time that has been caused by an unauthorised person gaining access to the network system. E.g. Users being asked to install malicious apps or transfer money through phishing sites. Fake websites that attempts to install app on mobile for providing various services or monetary benefits.

2.1.7 Malware

Malware is a software having a malicious intent to destroy/damage the operation of a system. This may occur due to poor programming, unintended fault or even usage related programming methods in relation to developing software. Malware can exists in all forms e.g. mobile phones, applications, websites

2.1.8 Local Network Compromise

The local area network can be exposed to the outside world through a device connected to the internal network of the organisation. This happens mostly with a connectivity of a device which belongs to a trusted employee, supplier or a partner. An employee can intentionally or unintentionally compromise the network [8]. Organisations should only allow the devices which meets the security requirements and standards, to connect to the corporate network. It is a best practice that the organisations run a check in the user’s device and then grant approval to use the corporate network based on the results gathered by the check.

3 Design

3.1 Overview

The design of this project work will help the corporate business to secure their network when BYOD concept is used in their corporate environments.

The project work design is a compilation of configuring servers and active and passive network devices. Log entries of the related servers and applications will be sent to central repository in order to analyse the log entries. Log correlation mechanisms will be performed in SIEM environment.

3.2 Methodology

3.2.1 IP Fire

IP Fire is an open source linux distribution which can act as a gateway router and a firewall. IP Fire is integrated with many services such as VPN server, web proxy server (Squid), intrusion detection and etc. Ip fire provides a web GUI to manage and monitor the inbuilt services. In my project work I have used Ip fire as a gateway router and also as web proxy server.

3.2.2 Gateway setup

IP Fire is installed and configured as the gateway to the internal network. In other words all the network traffic has been routed through this server. IP Fire is integrated with an in-built SQUID proxy server. In this project IP Fire proxy server is configured as a transparent proxy to the inter web clients. Event logs of the proxy server generated by the web requests and the logs of the network traffic events have been configured in order to take these log entries for further analysis.

3.2.3 Forwarding logs to OSSIM server

OSSIM [13] server has many ways to acquire events from remote and local sources. Generally events are forwarded to the OSSIM server. OSSIM has the ability to analyse the incoming/outgoing network traffic patterns and OSSM has the ability to analyse the system logs which are sent to the OSSIM agents/sensors. Collection of system logs is an effective method of analysing instead of network traffic being sent to OSSIM sensors for analysis. This is mainly because of amount of resources required to process and analyse network packets such as memory, processing and storage. There are two main methods of log collection done in OSSIM; agentless and agent based.

3.2.4 Agentless log collection

This involves a remote connection to the source server, whereas a service or a process login into the source server and access the event log data. This may lead to high resource spike such as CPU process and memory. In this scenario the Security Incident and Event Management (SIEM) systems will be configured to communicate with the source server API’s and request the relevant logs. However, this will be an expensive method of pulling out log records since it will involve user authentication during the process. Such as system administrators will have an administrative overhead to setup services, apply firewall rules and various other kinds of setups. Furthermore, agentless log collection method with face challenges such as system audit policies locally being changed in the source servers and also some of the critical information might not being sent to SIEMs.

3.2.5 Agent based log collection

With agent based log collection method the logs can be sent to the SIEM in real time in a rapid manner. This method will ensure that there will be less opportunity for an intruder to modify or delete the logs in order to conceal the evidences of the attacks. When using the agent based log collection method an agent should be actively setup and running in the source server where the agent will communicate the log related entries to the destination SIEM solution. There are many agents which we can use based on the platform of the operating system. E.g. for windows based operating systems agents like Cygwin, Datagram, Snare/Epilog can be used. For unix based operating systems syslog will help to forward the log entries to a destination server.

3.2.6 What is Syslog?

Syslog is a protocol that is used to communicate event messages between computers, servers and other networking devices. Certain software applications also use this method to communicate incidents to another destination server. The ISO/OSI architecture is used for this purpose which is similar to other transport layer protocols that are used in communicating and transmitting Syslog messages. Syslog independently supports major platforms such as Windows, Unix and Mac. Further, Syslog is also supported by many of the open source event logging applications.

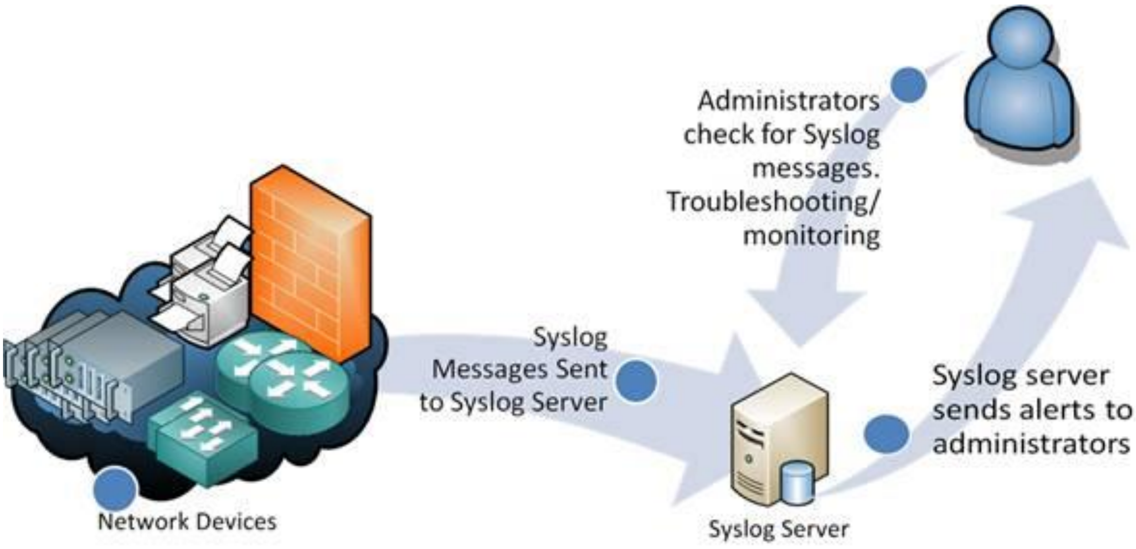
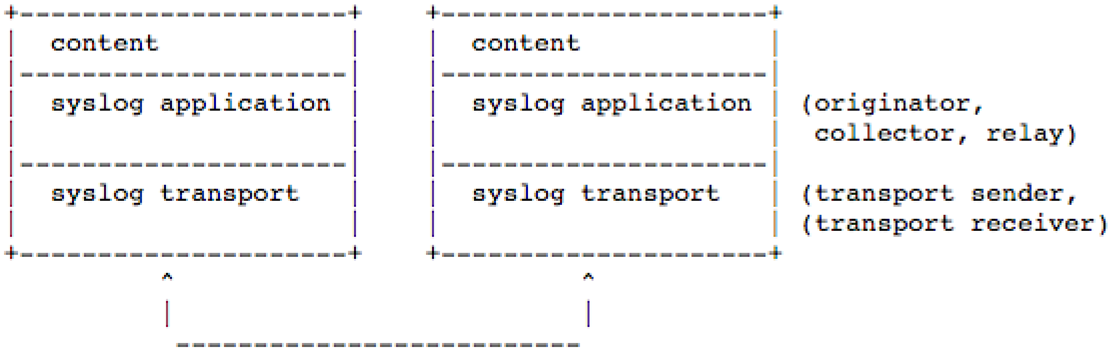


Figure 3 – Syslog Architecture [10]

At present, in the industry the best solution for logging is capturing local logs generated by different hardware devices and software applications through a centralised server and store in it. This is an effective technique to analyse logs during an event. It is a simplified method where the administrators have to correlate the logs which are stored in the central server rather, analysing local logs of respective servers in order to investigate and analyse an event. Therefore, forwarding local logs to a centralised location for analysis has become a standard method of log analysing among the industry IT professionals.

3.2.7 How Syslog works

At present in the industry the best solution for logging is capturing local logs generated by different hardware devices and software applications by a centralised server and store in it. This is an effective technique to analyse logs during an event. It is a simplified method where the administrators have to correlate the logs which are stored in the central server rather, analysing local logs of respective servers in order to investigate and analyse an event. Therefore, forwarding local logs to a centralised location for analysis has become a standard method of log analysing among the industry IT professionals.

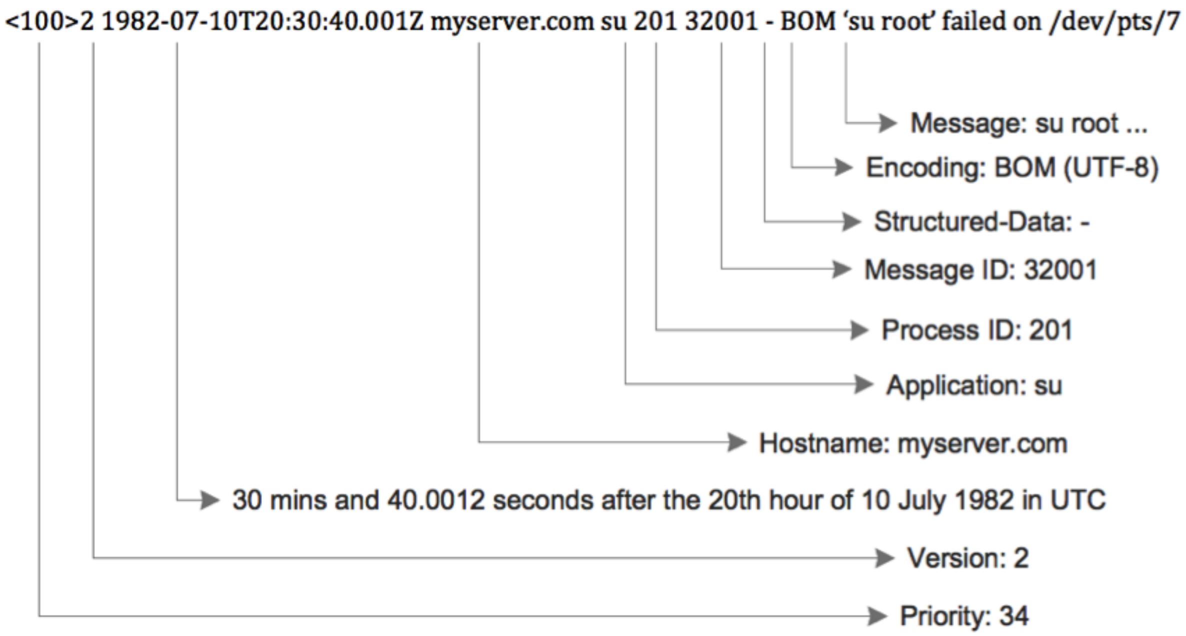


```
+----------------------+    +----------------------+
|  content             |    |  content             |
|----------------------|    |----------------------|
|  syslog application  |    |  syslog application  | (originator,
|                      |    |                      |  collector, relay)
|----------------------|    |----------------------|
|  syslog transport    |    |  syslog transport    | (transport sender,
|                      |    |                      | (transport receiver)
+----------------------+    +----------------------+
           ^                           ^
           |                           |
            --------------------------
```

Figure 4 – Syslog Layers [14]

A syslog server is configured to capture messages sent over the network. A process which listen UDP 514 is set to capture messages. But these messages are not acknowledged since it is an UDP. Some devices will send the syslog messages through TCP1468 in order to maintain the reliability.

The syslog messages have its standard format; the header section, structured data section and the message section. The header section consists with priority, version, timestamp, hostname, application, process id and message id. The next section which is the standard data section consists of data blocks in the format of key=value in between square brackets. E.g. [SDID@0 utilization="low" os="linux"]. The null values will be represented by a hyphen "-". While in the next and the last section, the detailed message will be shown. E.g "su root failed on /dev/pts/5"



<100>2 1982-07-10T20:30:40.001Z myserver.com su 201 32001 - BOM ‘su root’ failed on /dev/pts/7
Message: su root ...
Encoding: BOM (UTF-8)
Structured-Data: -
Message ID: 32001
Process ID: 201
Application: su
Hostname: myserver.com
30 mins and 40.0012 seconds after the 20th hour of 10 July 1982 in UTC
Version: 2
Priority: 34

Figure 5 – Syslog Header [14]

3.2.8 Advantages of using syslog

Syslog log is a commonly available logging tool mainly used in Unix based operating systems. Syslog clients are also available for Windows based operating systems as well. Using an agent based log management tool will eliminate the overhead of setting up the network, firewall, user authentication, service authentication. This will simplify the process for systems/network administrators in monitoring and identifying an incident that has been occurred.

At present, IT security related policies in many corporate environments do not allow the system/network administrators and other IT staff to install third party tools and applications. Moreover, these policies may not also allow unlocking unknown ports in devices as a security measure to ensure proper function of the log management tools

Hence, Syslog is well considered as a reliable and a secure tool in the IT industry where it can be used for log management, monitoring and investigating during an incident.

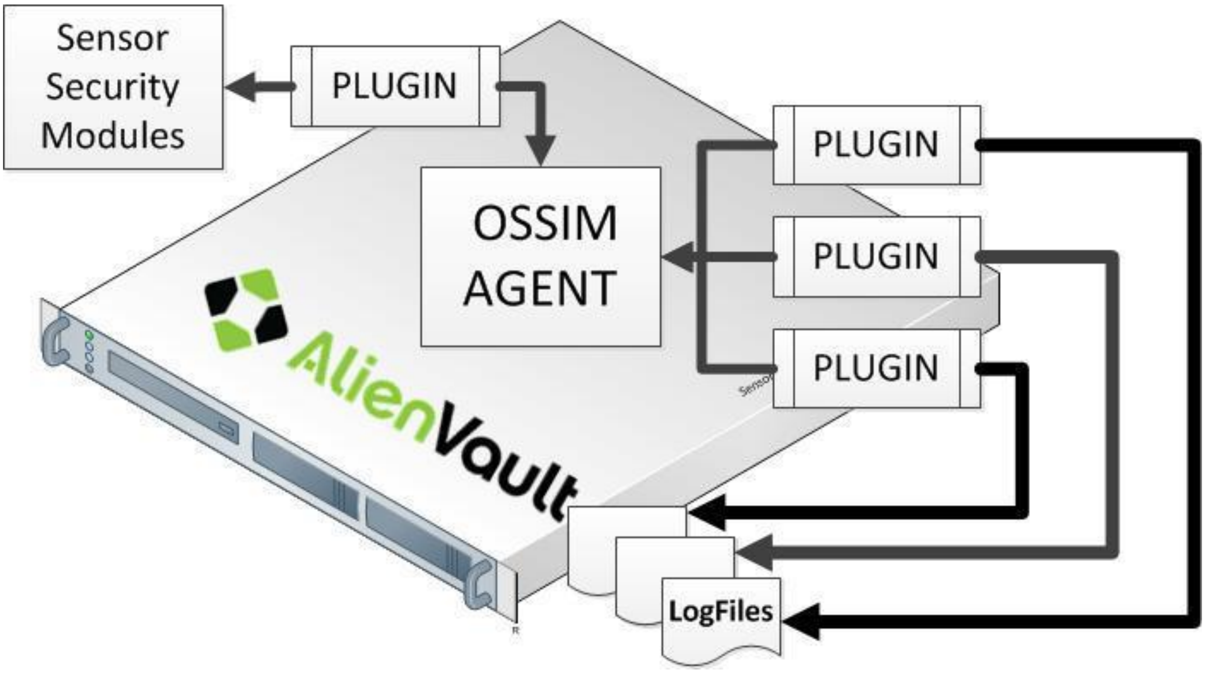


Figure 6 – Alien Vault OSSIM Agent Overview [11]

3.3 Correlating the collected events

The proposed solution is configured in way that all web traffic logs and iptables logs are received by the OSSIM rsyslog daemon. Logs received by the remote server which is the firewall server in this scenario are taken into account at the OSSIM server. We can create or use the existing plugins in OSSIM in order to find the unusual or suspicious behaviours of occurred events in the remote server.

My main focus in this project work is to monitor the web traffic and the network traffic. In order to execute this monitoring process I have used the log entries acquired by squid proxy service for web traffic and the log entries generated by Iptables are used for network traffic.

4 Implementation

4.1 Overview

In the implementation phase we have to parse the syslog entries for iptables and squid access log to the OSSIM server. In the OSSIM server we have to setup the plugins in order to detect suspicious behaviours in the gathered logs which will facilitate in providing the prompt results through OSSIM.

4.2 What OSSIM does?

OSSIM is an open source SIEM software. This product is developed by AlienVault and they distributed the product freely. The OSSIM is used by many corporate companies. OSSIM is a Debian Linux (64-bit) based distribution. OSSIM consists with major 4 components.

Framework

Using the framework the OSSIM users are able to monitor and manage the OSSIM tools and components. Administration has become easier because OSSIM provides an inbuilt web GUI to manage OSSIM.

Server

The important SIEM functions are handled by the server. Aggregation, risk assessment and correlation of events are captured by the OSSIM sensors through TCP 40001 port. Additionally the server is responsible of sending event to the database in order store the events which has been already occurred.

Database

OSSIM has a inbuilt MYSQL database. This database will basically store the event related details and the OSSIM configuration data.

Sensor

OSSIM sensors are responsible for mapping the events correctly. OSSIM sensor has two major components; OSSIM agent services and the rsyslog servers.

OSSIM agent services are a set of applications called plugins which perform the event log analysis and then normalises the event logs. Afterwards the processed ones will be sent to the server component.

Rsyslog service is a process which is listening on TCP514 and UDP 514 in order to capture the incoming event log details by the devices in the network. The logs then will be stored in the local server according to the rsyslog configuration.

The above components can be installed and setup in either a physical or a virtual machine which is the default installation. However, depending on the requirement the above components can be installed and setup on physical or virtual machines. This will depend on the size and configuration of the network to monitor and other tools. For my project work I have installed the OSSIM on a single machine which would a virtual machine.

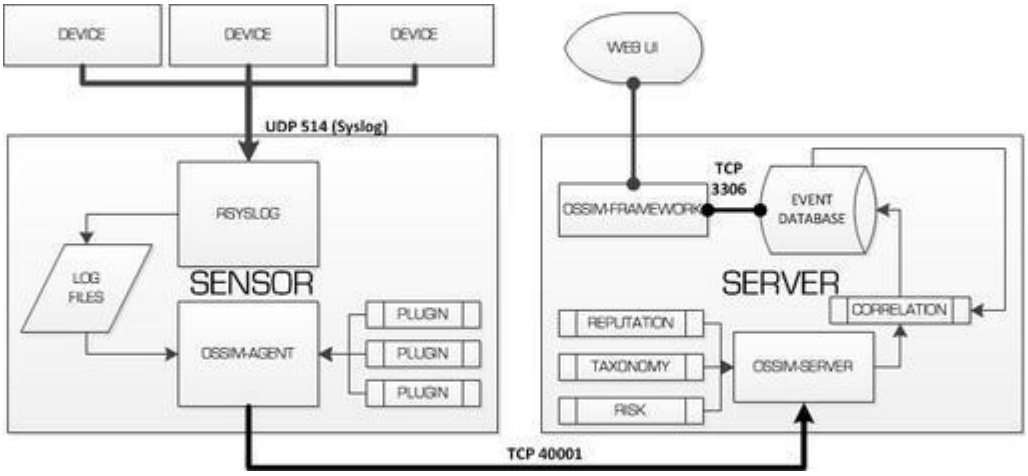
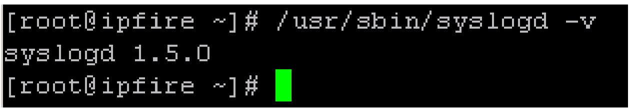


Figure 7 – OSSIM Log Correlation Overview [11]

4.3 Setting up Syslog

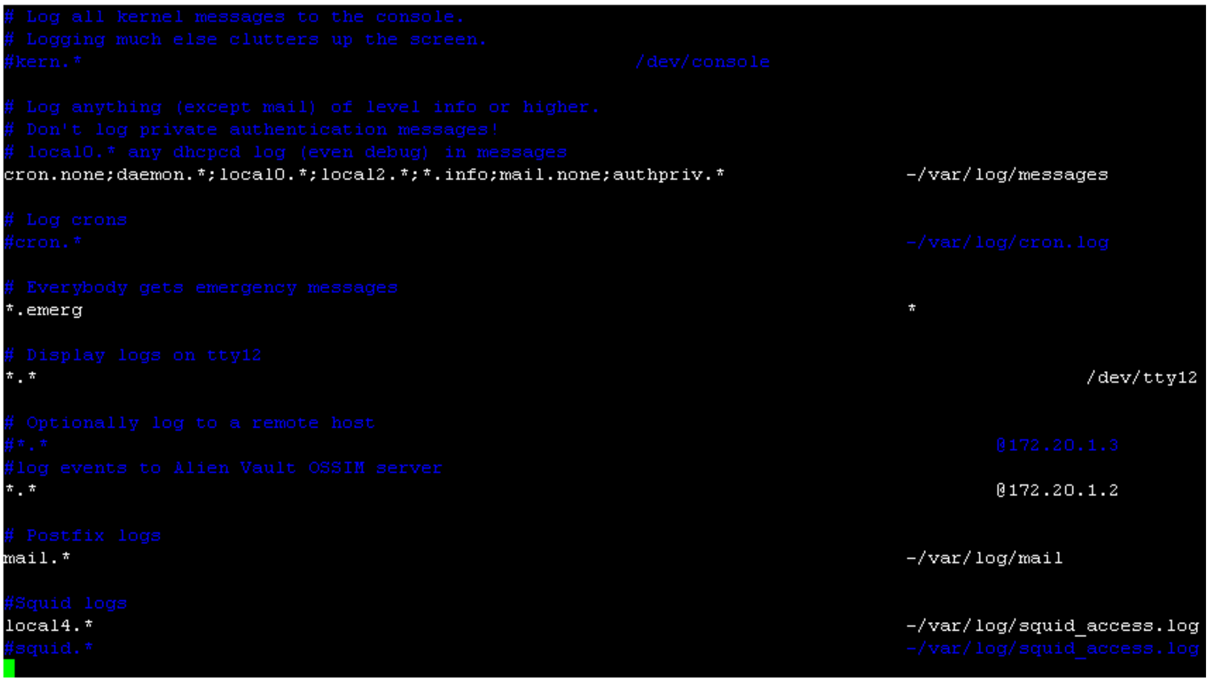
As the inaugural process, the event log setup needs to be implemented. Therefore in my project scenario the logs are collected at the gateway server. In the gateway server the event logs related to squid proxy and the iptables was set to parse to syslogd daemon for capturing. Syslogd daemon is process which is responsible for capturing event logs and transmitting the logs. In the gateway server syslog 1.5.0 is installed.



```
[root@ipfire ~]# /usr/sbin/syslogd -v
syslogd 1.5.0
[root@ipfire ~]# 
```

Figure 8 – Syslog Version in Gateway Server

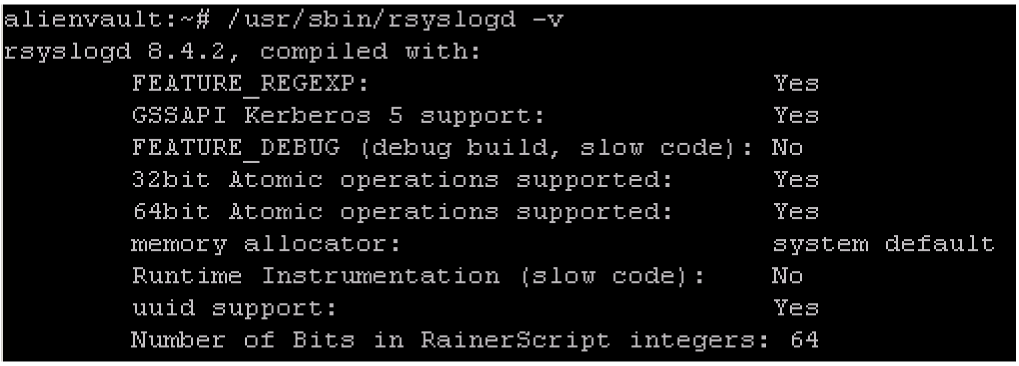
In this project work all the network traffic logs and the web proxy logs acquired by the syslogd daemon is sent to the SIEM server. Therefore log events are transmitted to the SIEM server by the syslogd daemon. Syslog uses UDP traffic to transmit these log events to remote servers.



```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
# local0.* any dhcpcd log (even debug) in messages
cron.none;daemon.*;local0.*;local2.*;*.info;mail.none;authpriv.*                -/var/log/messages

# Log crons
#cron.*                                                                         -/var/log/cron.log

# Everybody gets emergency messages
*.emerg                                                                         *

# Display logs on tty12
*.*                                                                                             /dev/tty12

# Optionally log to a remote host
#*.*                                                                                    @172.20.1.3
#log events to Alien Vault OSSIM server
*.*                                                                                     @172.20.1.2

# Postfix logs
mail.*                                                                          -/var/log/mail

#Squid logs
local4.*                                                                        -/var/log/squid_access.log
#squid.*                                                                        -/var/log/squid_access.log
```

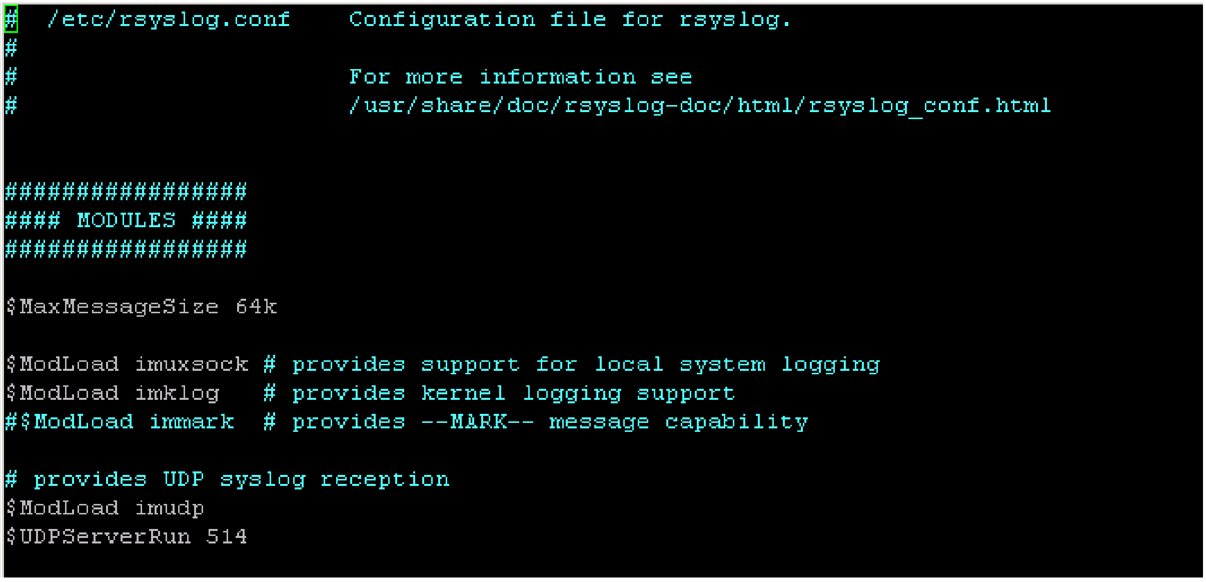
Figure 9 – Syslogd Configuration in Gateway Server

The remote server which is the OSSIM SIEM server in my scenario is configured to receive logs from other hosts. OSSIM SIEM unix distribution supports rsyslogs. Rsyslog is an advanced version of syslog. Both utilities have similar kind of operation however rsyslog has more advanced features. In rsyslog configuration of the OSSIM server is set to receive the event logs from remote hosts using the UDP 514 port. Here in the OSSIM server rsyslog version 8.4.2 has been installed and configured.



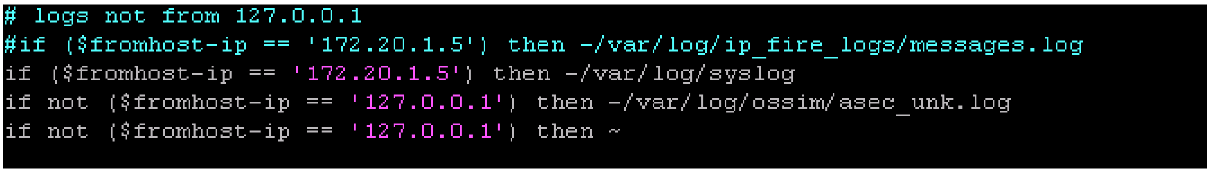
```
alienvault:~# /usr/sbin/rsyslogd -v
rsyslogd 8.4.2, compiled with:
        FEATURE_REGEXP:                         Yes
        GSSAPI Kerberos 5 support:              Yes
        FEATURE_DEBUG (debug build, slow code): No
        32bit Atomic operations supported:      Yes
        64bit Atomic operations supported:      Yes
        memory allocator:                       system default
        Runtime Instrumentation (slow code):    No
        uuid support:                           Yes
        Number of Bits in RainerScript integers: 64
```

Figure 10 – Rsyslog Version in OSSIM Server



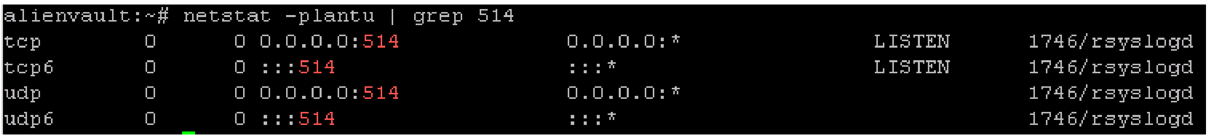
```
#  /etc/rsyslog.conf     Configuration file for rsyslog.
#
#                        For more information see
#                        /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html


#################
#### MODULES ####
#################

$MaxMessageSize 64k

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
```

Figure 11 – Rsyslogd Configuration in OSSIM Server (UDP Allow)



```
# logs not from 127.0.0.1
#if ($fromhost-ip == '172.20.1.5') then -/var/log/ip_fire_logs/messages.log
if ($fromhost-ip == '172.20.1.5') then -/var/log/syslog
if not ($fromhost-ip == '127.0.0.1') then -/var/log/ossim/asec_unk.log
if not ($fromhost-ip == '127.0.0.1') then ~
```

Figure 12 – Rsyslogd Configuration in OSSIM Server (Log Location)



```
alienvault:~# netstat -plantu | grep 514
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      1746/rsyslogd
tcp6       0      0 :::514                  :::*                    LISTEN      1746/rsyslogd
udp        0      0 0.0.0.0:514             0.0.0.0:*                           1746/rsyslogd
udp6       0      0 :::514                  :::*                                1746/rsyslogd
```

Figure 13 – OSSIM Server Listens for Logs

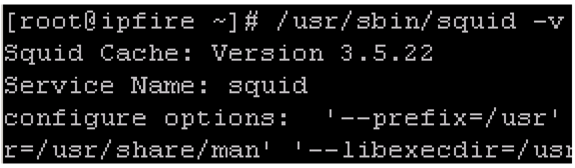
4.4 Setting up the OSSIM plugin for Squid and Iptables

As part of the process we need to acquire the squid and iptables logs from the IP Fire server and then using the squid and iptables plugins at the OSSIM server these logs should be imported to OSSIM event database.

Before configuring the sensor plugins an understanding of the log entries, recognising the patterns of the log entries are essential because the sensor plugin configuration is based on these patterns.

4.4.1 Setting up plugins for Squid

Squid is a utility that works as a forward proxy server and also can act as a web caching server which supports HTTP and FTP traffic. In this project I have configured a squid proxy server as forward proxy server to the local network. Simply speaking the web traffic generated from the BYOD devices is routed through the squid proxy server. In the gateway server squid version 3.5.22 has been installed and configured.



```
[root@ipfire ~]# /usr/sbin/squid -v
Squid Cache: Version 3.5.22
Service Name: squid
configure options:  '--prefix=/usr'
r=/usr/share/man' '--libexecdir=/us
```

Figure 14 – Squid Version in Gateway Server

Squid Configuration [APPENDIX B]

4.4.2 Setting up plugins for Iptables

Iptables is a firewall utility for unix based servers. In most unix based distributions by default iptables is pre-installed. This is a command line IP filtering tool which is used to allow or block the incoming and outgoing network traffic. Net filter is the intermediary tool which acts in between iptables and the kernel. Iptables is rule based where it matches the IP address against the rules. Furthermore iptables uses policy chains for its functionality.

Default tables are as below

1 Raw
2 Mangle
3 NAT
4 Filter

Defaults chains are as below

1 PREROUTING
2 INPUT
3 FORWARD
4 OUTPUT
5 POSTROUTING

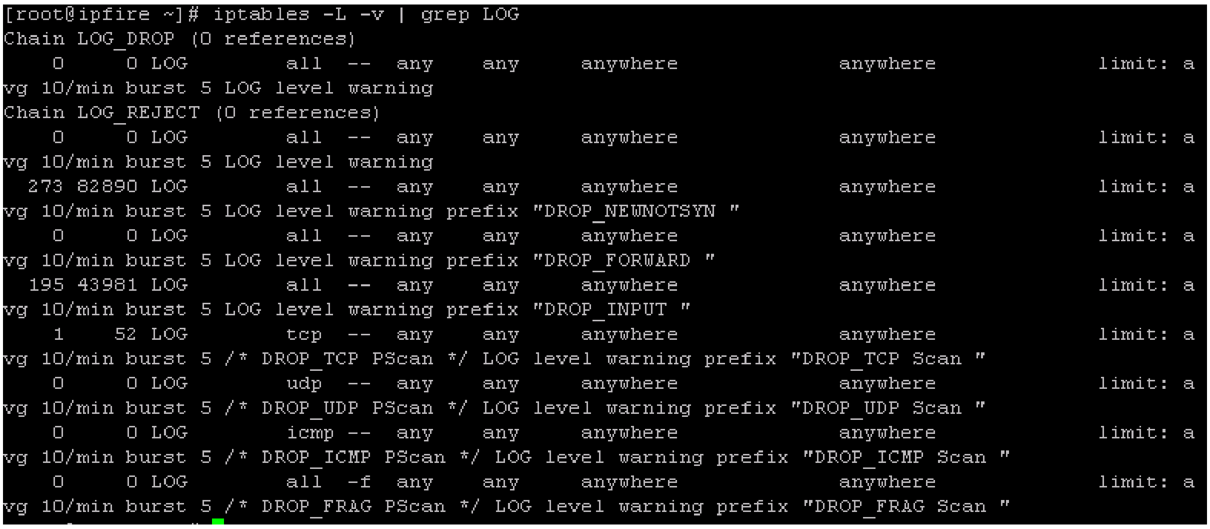
Mainly, INPUT chain is for the all the IP packets which comes to a certain device. If simplified, the incoming traffic to a certain device. The reply packets of any connection made through a particular device will be received through the INPUT chain. The OUTPUT chain is all about the outgoing traffic from a particular device. For example a web request made by a computer falls under OUTPUT chain. FORWARD chain is for the IP packets which are not to a given device but to another device. For example, default gateways. It simply forward the IP packets to the relevant device once received.

As explained earlier, iptables is a rule based utility which is matched against the rules which has been in place against a certain criteria. Following are the most commonly used targets in iptables.

1 ACCEPT: Packets are accepted and acknowledged
2 DROP: Packets are discarded
3 REJECT: Packets are discarded and informed to the sender
4 LOG: Traffic events are sent to the syslogd
5 DNAT: Overrides the destination IP
6 SNAT: Overrides the source IP

In the gateway server iptables have been configured to log all the IP traffic generated by the BYOD devices in the network. Also iptables dropped requests have been logged to the default log.

Iptables configuration [APPENDIX C]



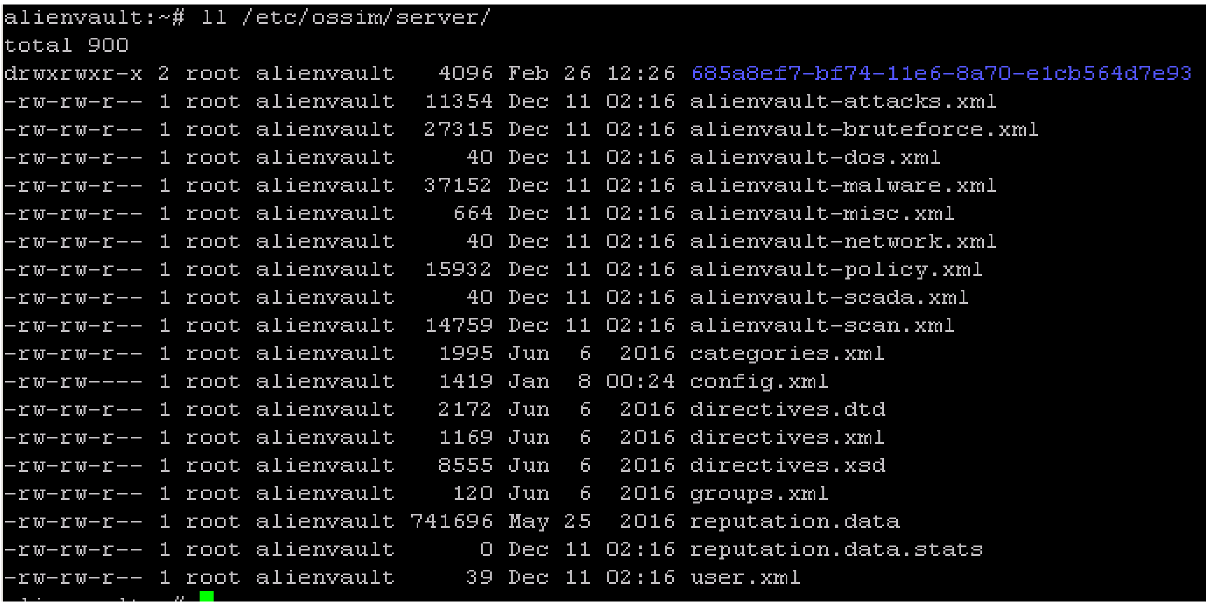
```
[root@ipfire ~]# iptables -L -v | grep LOG
Chain LOG_DROP (0 references)
    0     0 LOG        all  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 LOG level warning
Chain LOG_REJECT (0 references)
    0     0 LOG        all  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 LOG level warning
  273 82890 LOG        all  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 LOG level warning prefix "DROP_NEWNOTSYN "
    0     0 LOG        all  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 LOG level warning prefix "DROP_FORWARD "
  195 43981 LOG        all  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 LOG level warning prefix "DROP_INPUT "
    1    52 LOG        tcp  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 /* DROP_TCP PScan */ LOG level warning prefix "DROP_TCP Scan "
    0     0 LOG        udp  --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 /* DROP_UDP PScan */ LOG level warning prefix "DROP_UDP Scan "
    0     0 LOG        icmp --  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 /* DROP_ICMP PScan */ LOG level warning prefix "DROP_ICMP Scan "
    0     0 LOG        all  -f  any    any     anywhere             anywhere             limit: a
vg 10/min burst 5 /* DROP_FRAG PScan */ LOG level warning prefix "DROP_FRAG Scan "
```

Figure 15 – Iptables log rules in Gateway Server

4.4.3 Event log correlation

Real time event data in OSSIM web interface can be seen once the OSSIM plugin configurations are properly in place for squid and iptables. These captured data will help event log correlation. In this implementation, the next challenge would be how to correlate the squid and iptables events generated by the BYOD devices connected the local network.

OSSIM SIEM comes with integrated correlation rules which come with OSSIM installation. These rules are written in XML and these can be found the below location in the OSSIM server. OSSIM provides and options to edit these correlation rule in the web interface. And interface named by the directive editor will help to edit the correlation rules for the best optimisation.



```
alienvault:~# ll /etc/ossim/server/
total 900
drwxrwxr-x 2 root alienvault   4096 Feb 26 12:26 685a8ef7-bf74-11e6-8a70-e1cb564d7e93
-rw-rw-r-- 1 root alienvault  11354 Dec 11 02:16 alienvault-attacks.xml
-rw-rw-r-- 1 root alienvault  27315 Dec 11 02:16 alienvault-bruteforce.xml
-rw-rw-r-- 1 root alienvault     40 Dec 11 02:16 alienvault-dos.xml
-rw-rw-r-- 1 root alienvault  37152 Dec 11 02:16 alienvault-malware.xml
-rw-rw-r-- 1 root alienvault    664 Dec 11 02:16 alienvault-misc.xml
-rw-rw-r-- 1 root alienvault     40 Dec 11 02:16 alienvault-network.xml
-rw-rw-r-- 1 root alienvault  15932 Dec 11 02:16 alienvault-policy.xml
-rw-rw-r-- 1 root alienvault     40 Dec 11 02:16 alienvault-scada.xml
-rw-rw-r-- 1 root alienvault  14759 Dec 11 02:16 alienvault-scan.xml
-rw-rw-r-- 1 root alienvault   1995 Jun  6  2016 categories.xml
-rw-rw---- 1 root alienvault   1419 Jan  8 00:24 config.xml
-rw-rw-r-- 1 root alienvault   2172 Jun  6  2016 directives.dtd
-rw-rw-r-- 1 root alienvault   1169 Jun  6  2016 directives.xml
-rw-rw-r-- 1 root alienvault   8555 Jun  6  2016 directives.xsd
-rw-rw-r-- 1 root alienvault    120 Jun  6  2016 groups.xml
-rw-rw-r-- 1 root alienvault 741696 May 25  2016 reputation.data
-rw-rw-r-- 1 root alienvault      0 Dec 11 02:16 reputation.data.stats
-rw-rw-r-- 1 root alienvault     39 Dec 11 02:16 user.xml
```

Figure 16 – OSSIM Correlation Rules

5 Evaluation

5.1 Overview

For the testing environment various types of BYOD devices had been connected to the local area network and the network traffic of these devices have been monitoring and taken into account. If the objectives of my project work are accomplished, the administrators of the network will be alerted with the suspicious events taken place within the network.

5.2 Testing environment

As the testing bed I have replicated a local area network with servers in a virtual environment. OSSIM which is a SIEM developed by Alien Vault has been used for evaluate the test results.

5.2.1 SIEM server (Alien Vault)

Below table shows the SIEM server specifications and the configurations.

Table 1 – OSSIM Server Specification

Type of the server	Virtual server running on VMWare ESXi 5.1.0 and managed by vSphere 5.1.0
Server speciafication	Intel(R) Core(TM)2 Duo CPU 3.00GHz 2 Virtual Cores 4GB RAM 50GB Hard Disk
Operating system	Linux alienvault 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-1 (2016-03-06) x86_64 GNU/Linux
Software versions	OSSIM 5.2.5 Rsyslog 8.4.2

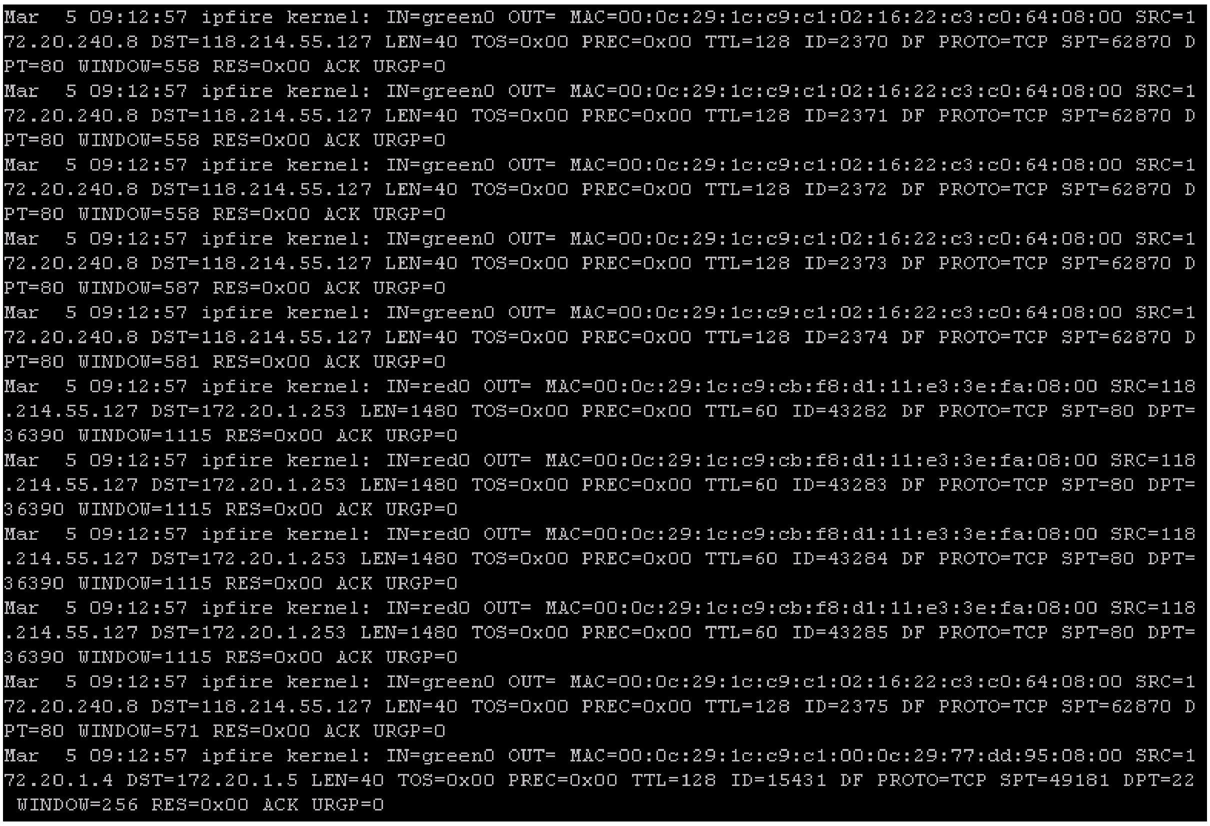
5.2.2 Gateway server (IP fire)

Table 2 – Gateway Server Specifications

Type of the server	Virtual server running on VMWare ESXi 5.1.0 and managed by vSphere 5.1.0
Server speciafication	Intel(R) Core(TM)2 Duo CPU 3.00GHz 2 Virtual Cores 1GB RAM 10GB Hard Disk
Operating system	Linux ipfire 3.14.79-ipfire #1 SMP Wed Dec 14 01:15:33 GMT 2016 x86_64 GNU/Linux
Software versions	Squid 3.5.22 Iptables 1.4.21 Syslog 1.5.0

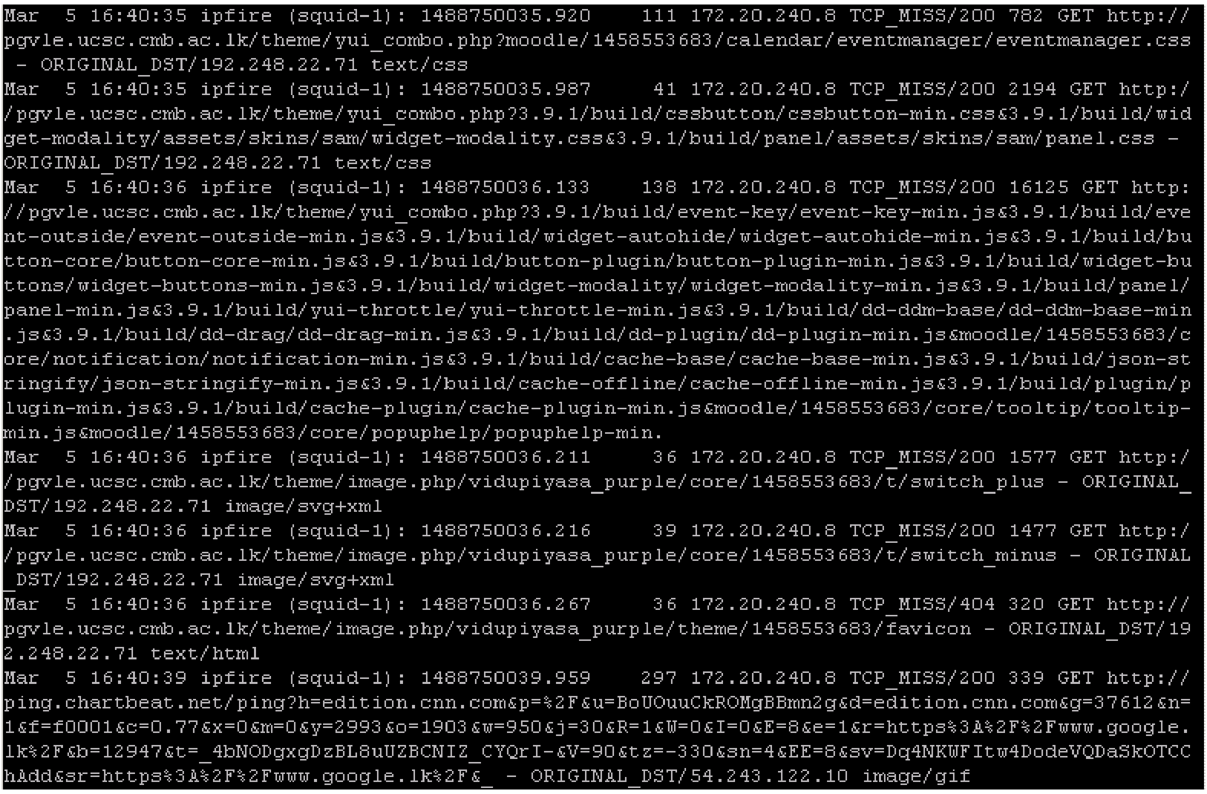
5.3 Event logs setup

In this testing environment event logs from squid and iptables are logged to syslog at the gateway server. First of all we have to make sure that the squid and iptables are sending their events to the syslog. This part has to be done successfully before sending the syslogs to the OSSIM server. Figure 17 and Figure 18 are exhibits of Iptables and Squid.



```
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2370 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=558 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2371 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=558 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2372 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=558 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2373 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=587 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2374 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=581 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=118
.214.55.127 DST=172.20.1.253 LEN=1480 TOS=0x00 PREC=0x00 TTL=60 ID=43282 DF PROTO=TCP SPT=80 DPT=
36390 WINDOW=1115 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=118
.214.55.127 DST=172.20.1.253 LEN=1480 TOS=0x00 PREC=0x00 TTL=60 ID=43283 DF PROTO=TCP SPT=80 DPT=
36390 WINDOW=1115 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=118
.214.55.127 DST=172.20.1.253 LEN=1480 TOS=0x00 PREC=0x00 TTL=60 ID=43284 DF PROTO=TCP SPT=80 DPT=
36390 WINDOW=1115 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=118
.214.55.127 DST=172.20.1.253 LEN=1480 TOS=0x00 PREC=0x00 TTL=60 ID=43285 DF PROTO=TCP SPT=80 DPT=
36390 WINDOW=1115 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=1
72.20.240.8 DST=118.214.55.127 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=2375 DF PROTO=TCP SPT=62870 D
PT=80 WINDOW=571 RES=0x00 ACK URGP=0
Mar  5 09:12:57 ipfire kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:00:0c:29:77:dd:95:08:00 SRC=1
72.20.1.4 DST=172.20.1.5 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=15431 DF PROTO=TCP SPT=49181 DPT=22
 WINDOW=256 RES=0x00 ACK URGP=0
```

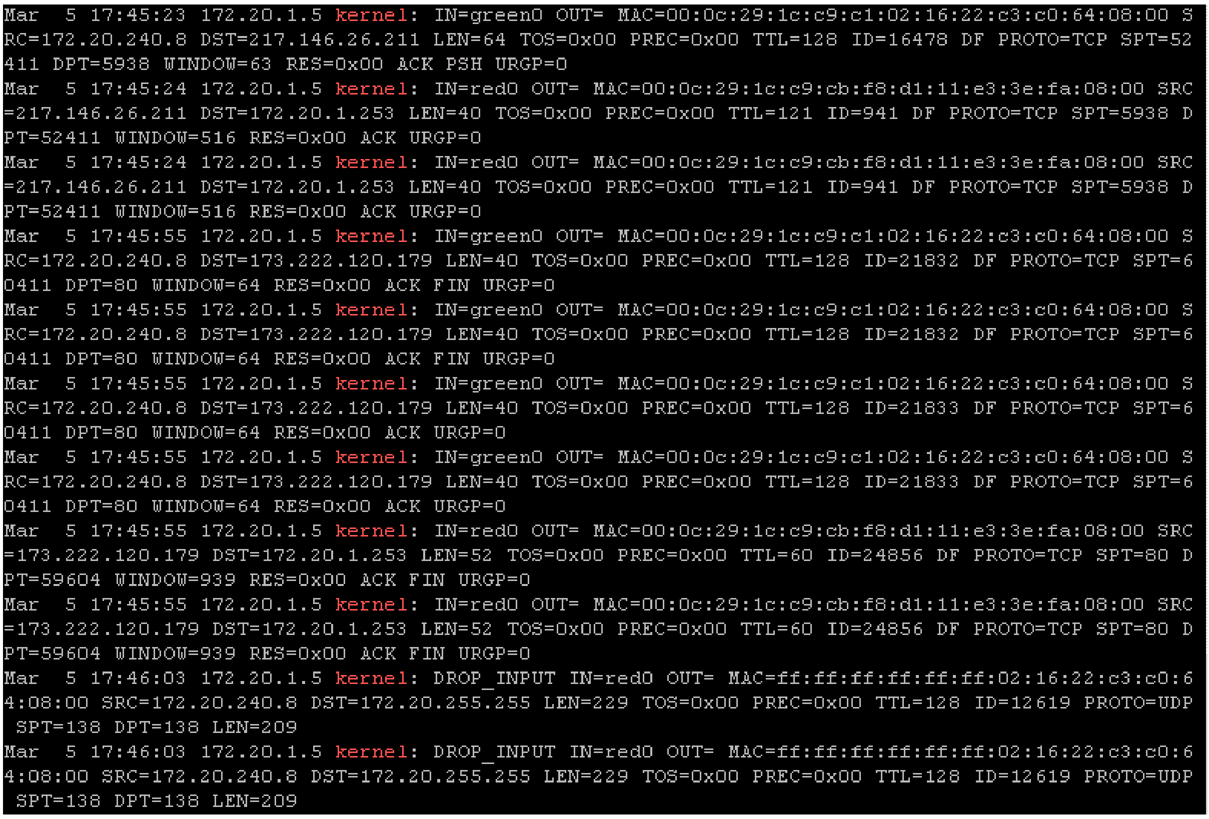
Figure 17 – Iptables log events in Gateway Server



```
Mar  5 16:40:35 ipfire (squid-1): 1488750035.920    111 172.20.240.8 TCP_MISS/200 782 GET http://
pgvle.ucsc.cmb.ac.lk/theme/yui_combo.php?moodle/1458553683/calendar/eventmanager/eventmanager.css
 - ORIGINAL_DST/192.248.22.71 text/css
Mar  5 16:40:35 ipfire (squid-1): 1488750035.987     41 172.20.240.8 TCP_MISS/200 2194 GET http:/
/pgvle.ucsc.cmb.ac.lk/theme/yui_combo.php?3.9.1/build/cssbutton/cssbutton-min.css&3.9.1/build/wid
get-modality/assets/skins/sam/widget-modality.css&3.9.1/build/panel/assets/skins/sam/panel.css -
ORIGINAL_DST/192.248.22.71 text/css
Mar  5 16:40:36 ipfire (squid-1): 1488750036.133    138 172.20.240.8 TCP_MISS/200 16125 GET http:
//pgvle.ucsc.cmb.ac.lk/theme/yui_combo.php?3.9.1/build/event-key/event-key-min.js&3.9.1/build/eve
nt-outside/event-outside-min.js&3.9.1/build/widget-autohide/widget-autohide-min.js&3.9.1/build/bu
tton-core/button-core-min.js&3.9.1/build/button-plugin/button-plugin-min.js&3.9.1/build/widget-bu
ttons/widget-buttons-min.js&3.9.1/build/widget-modality/widget-modality-min.js&3.9.1/build/panel/
panel-min.js&3.9.1/build/yui-throttle/yui-throttle-min.js&3.9.1/build/dd-ddm-base/dd-ddm-base-min
.js&3.9.1/build/dd-drag/dd-drag-min.js&3.9.1/build/dd-plugin/dd-plugin-min.js&moodle/1458553683/c
ore/notification/notification-min.js&3.9.1/build/cache-base/cache-base-min.js&3.9.1/build/json-st
ringify/json-stringify-min.js&3.9.1/build/cache-offline/cache-offline-min.js&3.9.1/build/plugin/p
lugin-min.js&3.9.1/build/cache-plugin/cache-plugin-min.js&moodle/1458553683/core/tooltip/tooltip-
min.js&moodle/1458553683/core/popuphelp/popuphelp-min.
Mar  5 16:40:36 ipfire (squid-1): 1488750036.211     36 172.20.240.8 TCP_MISS/200 1577 GET http:/
/pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/core/1458553683/t/switch_plus - ORIGINAL_
DST/192.248.22.71 image/svg+xml
Mar  5 16:40:36 ipfire (squid-1): 1488750036.216     39 172.20.240.8 TCP_MISS/200 1477 GET http:/
/pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/core/1458553683/t/switch_minus - ORIGINAL
_DST/192.248.22.71 image/svg+xml
Mar  5 16:40:36 ipfire (squid-1): 1488750036.267     36 172.20.240.8 TCP_MISS/404 320 GET http://
pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/theme/1458553683/favicon - ORIGINAL_DST/19
2.248.22.71 text/html
Mar  5 16:40:39 ipfire (squid-1): 1488750039.959    297 172.20.240.8 TCP_MISS/200 339 GET http://
ping.chartbeat.net/ping?h=edition.cnn.com&p=%2F&u=BoUOuuCkROMgBBmn2g&d=edition.cnn.com&g=37612&n=
1&f=f0001&c=0.77&x=0&m=0&y=2993&o=1903&w=950&j=30&R=1&W=0&I=0&E=8&e=1&r=https%3A%2F%2Fwww.google.
lk%2F&b=12947&t=_4bNODgxgDzBL8uUZBCNIZ_CYQrI-&V=90&tz=-330&sn=4&EE=8&sv=Dq4NKWFItw4DodeVQDaSkOTCC
hAdd&sr=https%3A%2F%2Fwww.google.lk%2F&_ - ORIGINAL_DST/54.243.122.10 image/gif
```

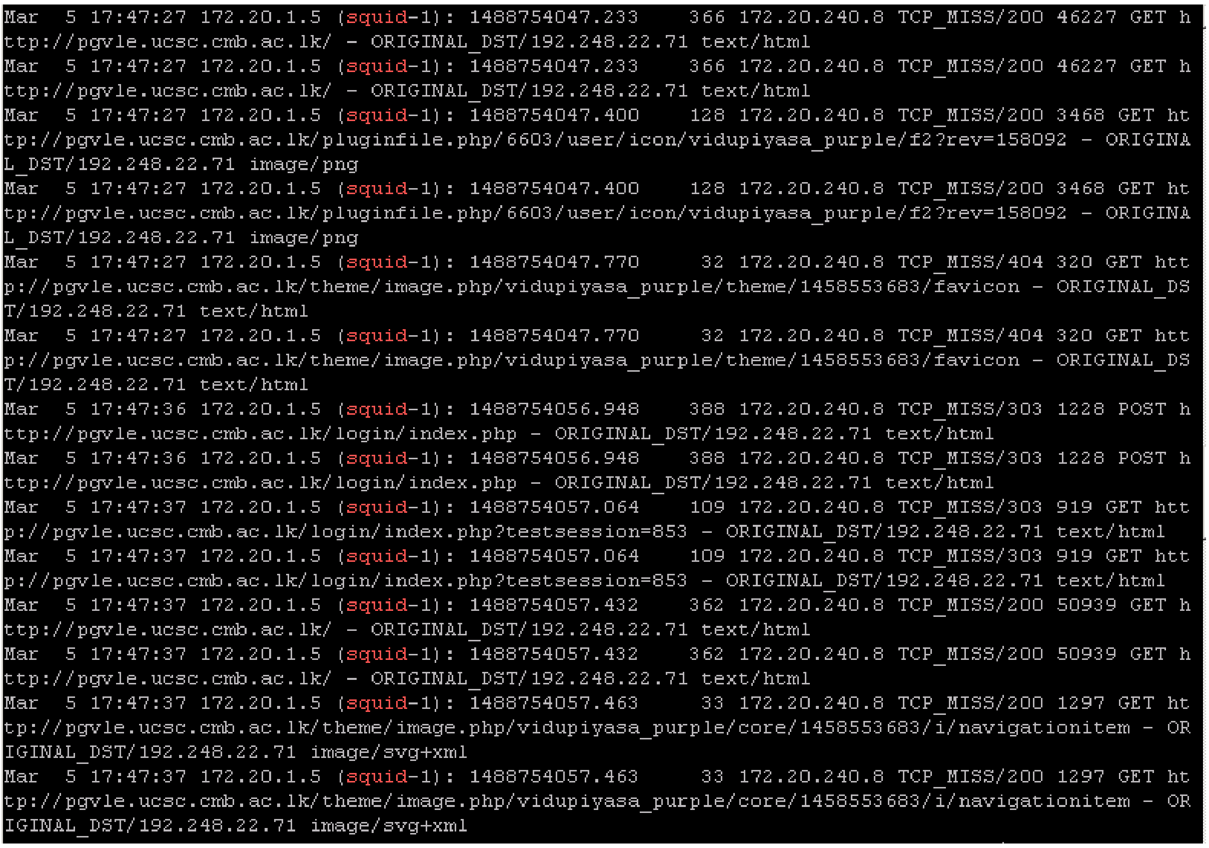
Figure 18 – Squid log events in Gateway Server

Once the events are correctly logged in the syslog of the gateway server, these logs will be sent to the OSSIM SIEM server via UPD traffic. The OSSIM server will be listening to syslog through UDP 514. Below are the event logs captured by the OSSIM server from the gateway server. In this scenario the event logs received from the 172.20.1.5 are the event logs sent from the gateway server.



```
Mar  5 17:45:23 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 S
RC=172.20.240.8 DST=217.146.26.211 LEN=64 TOS=0x00 PREC=0x00 TTL=128 ID=16478 DF PROTO=TCP SPT=52
411 DPT=5938 WINDOW=63 RES=0x00 ACK PSH URGP=0
Mar  5 17:45:24 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC
=217.146.26.211 DST=172.20.1.253 LEN=40 TOS=0x00 PREC=0x00 TTL=121 ID=941 DF PROTO=TCP SPT=5938 D
PT=52411 WINDOW=516 RES=0x00 ACK URGP=0
Mar  5 17:45:24 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC
=217.146.26.211 DST=172.20.1.253 LEN=40 TOS=0x00 PREC=0x00 TTL=121 ID=941 DF PROTO=TCP SPT=5938 D
PT=52411 WINDOW=516 RES=0x00 ACK URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 S
RC=172.20.240.8 DST=173.222.120.179 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=21832 DF PROTO=TCP SPT=6
0411 DPT=80 WINDOW=64 RES=0x00 ACK FIN URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 S
RC=172.20.240.8 DST=173.222.120.179 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=21832 DF PROTO=TCP SPT=6
0411 DPT=80 WINDOW=64 RES=0x00 ACK FIN URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 S
RC=172.20.240.8 DST=173.222.120.179 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=21833 DF PROTO=TCP SPT=6
0411 DPT=80 WINDOW=64 RES=0x00 ACK URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 S
RC=172.20.240.8 DST=173.222.120.179 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=21833 DF PROTO=TCP SPT=6
0411 DPT=80 WINDOW=64 RES=0x00 ACK URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC
=173.222.120.179 DST=172.20.1.253 LEN=52 TOS=0x00 PREC=0x00 TTL=60 ID=24856 DF PROTO=TCP SPT=80 D
PT=59604 WINDOW=939 RES=0x00 ACK FIN URGP=0
Mar  5 17:45:55 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC
=173.222.120.179 DST=172.20.1.253 LEN=52 TOS=0x00 PREC=0x00 TTL=60 ID=24856 DF PROTO=TCP SPT=80 D
PT=59604 WINDOW=939 RES=0x00 ACK FIN URGP=0
Mar  5 17:46:03 172.20.1.5 kernel: DROP_INPUT IN=red0 OUT= MAC=ff:ff:ff:ff:ff:ff:02:16:22:c3:c0:6
4:08:00 SRC=172.20.240.8 DST=172.20.255.255 LEN=229 TOS=0x00 PREC=0x00 TTL=128 ID=12619 PROTO=UDP
 SPT=138 DPT=138 LEN=209
Mar  5 17:46:03 172.20.1.5 kernel: DROP_INPUT IN=red0 OUT= MAC=ff:ff:ff:ff:ff:ff:02:16:22:c3:c0:6
4:08:00 SRC=172.20.240.8 DST=172.20.255.255 LEN=229 TOS=0x00 PREC=0x00 TTL=128 ID=12619 PROTO=UDP
 SPT=138 DPT=138 LEN=209
```

Figure 19 – Iptables log messages received from OSSIM Server



```
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.233    366 172.20.240.8 TCP_MISS/200 46227 GET h
ttp://pgvle.ucsc.cmb.ac.lk/ - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.233    366 172.20.240.8 TCP_MISS/200 46227 GET h
ttp://pgvle.ucsc.cmb.ac.lk/ - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.400    128 172.20.240.8 TCP_MISS/200 3468 GET ht
tp://pgvle.ucsc.cmb.ac.lk/pluginfile.php/6603/user/icon/vidupiyasa_purple/f2?rev=158092 - ORIGINA
L_DST/192.248.22.71 image/png
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.400    128 172.20.240.8 TCP_MISS/200 3468 GET ht
tp://pgvle.ucsc.cmb.ac.lk/pluginfile.php/6603/user/icon/vidupiyasa_purple/f2?rev=158092 - ORIGINA
L_DST/192.248.22.71 image/png
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.770     32 172.20.240.8 TCP_MISS/404 320 GET htt
p://pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/theme/1458553683/favicon - ORIGINAL_DS
T/192.248.22.71 text/html
Mar  5 17:47:27 172.20.1.5 (squid-1): 1488754047.770     32 172.20.240.8 TCP_MISS/404 320 GET htt
p://pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/theme/1458553683/favicon - ORIGINAL_DS
T/192.248.22.71 text/html
Mar  5 17:47:36 172.20.1.5 (squid-1): 1488754056.948    388 172.20.240.8 TCP_MISS/303 1228 POST h
ttp://pgvle.ucsc.cmb.ac.lk/login/index.php - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:36 172.20.1.5 (squid-1): 1488754056.948    388 172.20.240.8 TCP_MISS/303 1228 POST h
ttp://pgvle.ucsc.cmb.ac.lk/login/index.php - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.064    109 172.20.240.8 TCP_MISS/303 919 GET htt
p://pgvle.ucsc.cmb.ac.lk/login/index.php?testsession=853 - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.064    109 172.20.240.8 TCP_MISS/303 919 GET htt
p://pgvle.ucsc.cmb.ac.lk/login/index.php?testsession=853 - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.432    362 172.20.240.8 TCP_MISS/200 50939 GET h
ttp://pgvle.ucsc.cmb.ac.lk/ - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.432    362 172.20.240.8 TCP_MISS/200 50939 GET h
ttp://pgvle.ucsc.cmb.ac.lk/ - ORIGINAL_DST/192.248.22.71 text/html
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.463     33 172.20.240.8 TCP_MISS/200 1297 GET ht
tp://pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/core/1458553683/i/navigationitem - OR
IGINAL_DST/192.248.22.71 image/svg+xml
Mar  5 17:47:37 172.20.1.5 (squid-1): 1488754057.463     33 172.20.240.8 TCP_MISS/200 1297 GET ht
tp://pgvle.ucsc.cmb.ac.lk/theme/image.php/vidupiyasa_purple/core/1458553683/i/navigationitem - OR
IGINAL_DST/192.248.22.71 image/svg+xml
```

Figure 20 – Squid log messages received from OSSIM Server

5.3.1 Iptables event logs received from gateway server to OSSIM server

Mar 5 22:19:09 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=74.125.68.188 DST=172.20.1.253 LEN=52 TOS=0x00 PREC=0x00 TTL=48 ID=20071 PROTO=TCP SPT=5228 DPT=60484 WINDOW=360 RES=0x00 ACK URGP=0

The above log event which has been received by the OSSIM server, can be normalised to a readable manner using regular expressions. A python script named regexp.py [APPENDIX A] has been used to test the data normalisation for the event log messages.

Regular Expression for iptables (Pattern 1)

regexp=(\S+\s+\d+\s+\d\d:\d\d:\d\d)\s+(\S+) (\S+):.*?(?:Iptbl=)?(\S+)\s+IN=(\S+) OUT=(\S*)\s+(?:MAC=(?P<mac>[^\s]*)\s+)?SRC=(\S+) DST=(\S+) LEN=(\d+) \S+ \S+ TTL=(\d+) .*? PROTO=(\S*) SPT=(\d+) DPT=(\d+)

date={normalize_date($1)}

plugin_sid={translate($4)}

src_ip={$8}

dst_ip={$9}

protocol={$12}

src_port={$13}

dst_port={$14}

```
userdata1=server: {$2}, sourcewpid: {$3}, in: {$5}, out: {$6}, len: {$10}, ttl: {$11}
```

Regular Expression for iptables (Pattern 2)

```
regexp=(\S+\s+\d+\s+\d\d:\d\d:\d\d)\s+(?P<sensor>\S*)\s+(\S*):.*IN=(\S*)\s+OUT=(\S*)\s+(?:MAC=(?P<mac>[^\s]*)\s+)?SRC=(?P<src_ip>\S+)\s+DST=(?P<dst_ip>\S+).*\s+PROTO=(?P<proto>\S*)\s+SPT=(?P<src_port>\S+)\s+DPT=(?P<dst_port>\S+)

date={normalize_date($1)}

device={resolv($sensor)}

protocol={$proto}

plugin_sid=6

src_ip={$src_ip}

dst_ip={$dst_ip}

src_port={$src_port}

dst_port={$dst_port}

userdata1={$mac}
```

Regular Expression for iptables (Pattern 3)

```
regexp=(\S+\s+\d+\s+\d\d:\d\d:\d\d)\s+(?P<sensor>\S*)\s+(\S*):.*?(?P<rule>\S+)\s+IN=(?P<intinf>\S*)\s+OUT=(?P<outif>\S*)\s+(?:MAC=(?P<mac>[^\s]*)\s+)?SRC=(?P<src_ip>\S+)\s+DST=(?P<dst_ip>\S+).*\s+PROTO=(?P<proto>\S*)\s+SPT=(?P<src_port>\S+)\s+DPT=(?P<dst_port>\S+)

date={normalize_date($1)}

device={resolv($sensor)}

protocol={$proto}

plugin_sid={translate($rule)}

src_ip={$src_ip}

dst_ip={$dst_ip}

src_port={$src_port}

dst_port={$dst_port}

userdata1={$mac}

userdata2={$intinf}

userdata3={$outif}
```

Regular Expression for iptables (Pattern 4)

```
regexp="(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+.*?Iptbl=(?P<iptabl>\S+)\s+IN=(?P<iniface>\S+).*?MAC=(?P<mac_address>\S+)\s+SRC=(?P<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+DST=(?P<dst_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}).*?PROTO=(?P<protocol>\S+).*?SPT=(?P<src_port>\d+)\s+DPT=(?P<dst_port>\d+)"

event_type=event

date={normalize_date($date)}

protocol={$protocol}

src_ip={$src_ip}

dst_ip={$dst_ip}

src_port={$src_port}

dst_port={$dst_port}

userdata1={$mac_address}

userdata2={$iniface}

userdata3={$iptabl}
```

5.3.2 Squid event logs received from gateway server to OSSIM server

```
Mar  5 23:24:25 172.20.1.5 (squid-1): 1488774265.552    660 172.20.240.8 TCP_MISS/200 193256 GET http://ucsc.cmb.ac.lk/wp-content/uploads/2016/03/BIT-convocation.jpg - ORIGINAL_DST/192.248.22.125 image/jpeg
```

Similar to iptables, the above squid event log entry received by the OSSIM server, can be normalised using regular expressions in to readable manner.

Regular Expression for squid (Pattern 1)

```
precheck='squid'

regexp='(?P<date>\SYSLOG_DATE)\s+(?P<sensor>\S+)\s+squid\S+\s+\S+\s+\S+\s+(?P<host>\S+)\s+(?P<sid>[^\/]+)/(?P<http_code>\d+)\s+\d+\s+(?P<http_method>\S+)\s+(?P<url>\S+)'

src_ip={resolv($host)}

dst_ip={resolv($sensor)}

plugin_sid={translate($sid)}

userdata1={$http_method}

userdata2={translate($http_code)}

userdata3={$http_code}

userdata4={$url}
```

Regular Expression for squid (Pattern 2)

```
regexp='\d+\.\d+\s+\d+\s+(?P<host>[^\s]+)\s+[^\/]+\/(?P<sid>(\d+))\s+\d+\s+\w+\s+(?P<url>[^\s]+)\s+\-\s+\S+\/(?P<dst_ip>[^\s]+).*'

src_ip={resolv($host)}

dst_ip={resolv($dst_ip)}

plugin_sid={$sid}

userdata1={$3}
```

Regular Expression for squid (Pattern 3)

```
regexp='(\IPV4) (\S+) (\S+) \[(?P<date>(\d\d)\/(\w\w\w)\/(\d\d\d\d):(\d\d):(\d\d):(\d\d)).+"(?P<info>.+)" (?P<sid>\d+) (\S+)'

src_ip={$1}

date={normalize_date($date)}

plugin_sid={$sid}

dst_ip=127.0.0.1

dst_port=80
```

5.3.3 Testing the event log messages using the regexp.py script

Iptables event log

```
Multiple regexp mode used, parsing iptables.cfg

Matched using 0002 - iptables

Mar 5 22:19:09 172.20.1.5 kernel: IN=red0 OUT= MAC=00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00 SRC=74.125.68.188 DST=172.20.1.253 LEN=52 TOS=0x00 PREC=0x00 TTL=48 ID=20071 PROTO=TCP SPT=5228 DPT=60484 WINDOW=360 RES=0x00 ACK URGP=0\n

[('Mar 5 22:19:09', '172.20.1.5', 'kernel', 'red0', '', '00:0c:29:1c:c9:cb:f8:d1:11:e3:3e:fa:08:00', '74.125.68.188', '172.20.1.253', 'TCP', '5228', '60484')]

-----------------------------------------------------------------------------

Rule:   0001 - iptables

                              Matched 0 times

Rule:   0002 - iptables

                              Matched 1 times

Rule:   0003 - iptables

                              Matched 0 times
```

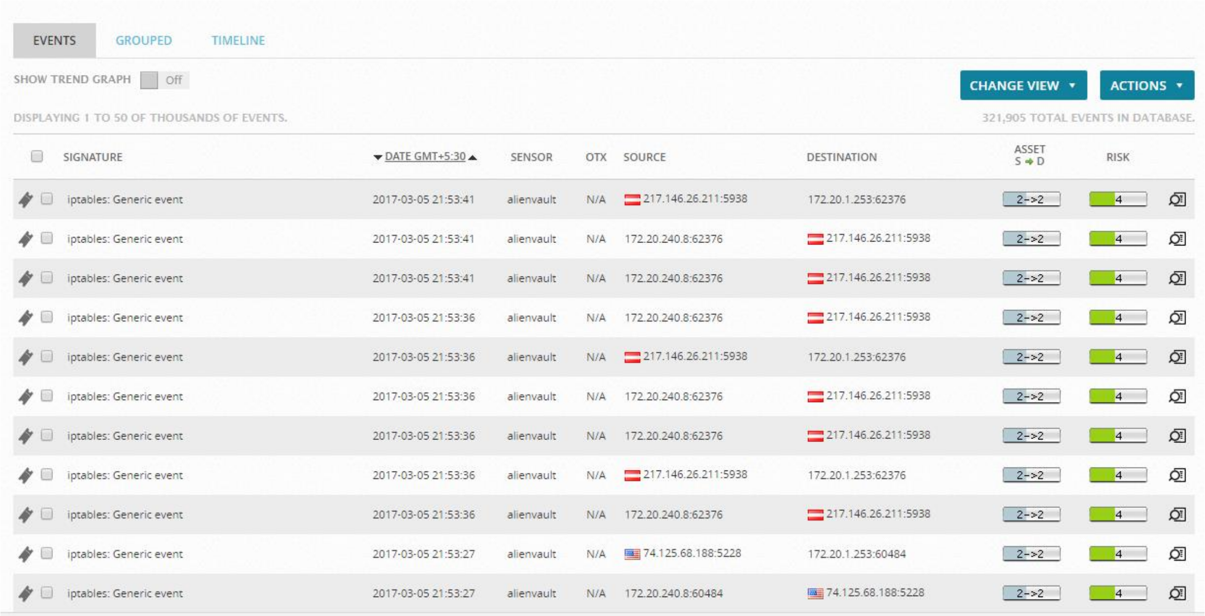
```
Rule:   0004 - iptables
                                        Matched 0 times
Counted 1 lines.
Matched 1 lines.
Ignored 0 blank lines.
```

Squid event log messages

```
Multiple regexp mode used, parsing squid.cfg
atched using 0002 - squid-access-old
Mar  5 23:24:25 172.20.1.5 (squid-1): 1488774265.552    660 172.20.240.8 TCP_MISS/200 193256 GET http://ucsc.cmb.ac.lk/wp-content/uploads/2016/03/BIT-convocation.jpg - ORIGINAL_DST/192.248.22.125 image/jpeg\n
[('172.20.240.8', '200', '200', 'http://ucsc.cmb.ac.lk/wp-content/uploads/2016/03/BIT-convocation.jpg', '192.248.22.125')]
-----------------------------------------------------------------------
Rule:   0001 - squid-access
                                        Matched 0 times
Rule:   0002 - squid-access-old
                                        Matched 1 times
Rule:   0003 - squid-apache-access-old
                                        Matched 0 times
Counted 1 lines.
Matched 1 lines.
Ignored 0 blank lines.
```

5.4 Correlation of the Iptables and Squid event log data

In OSSIM SIEM plugin repository there are plugin which are able to analyse the nature of the iptables and squid event logs. iptables.cfg, squid.cfg, squidGuard.cfg can be found in */etc/ossim/agent/plugin/* directory. These tools will monitor the behaviour of the network traffic which is generated by the BYOD devices and will alert the suspicious activities. These alerts can be viewed in the OSSIM web interface.

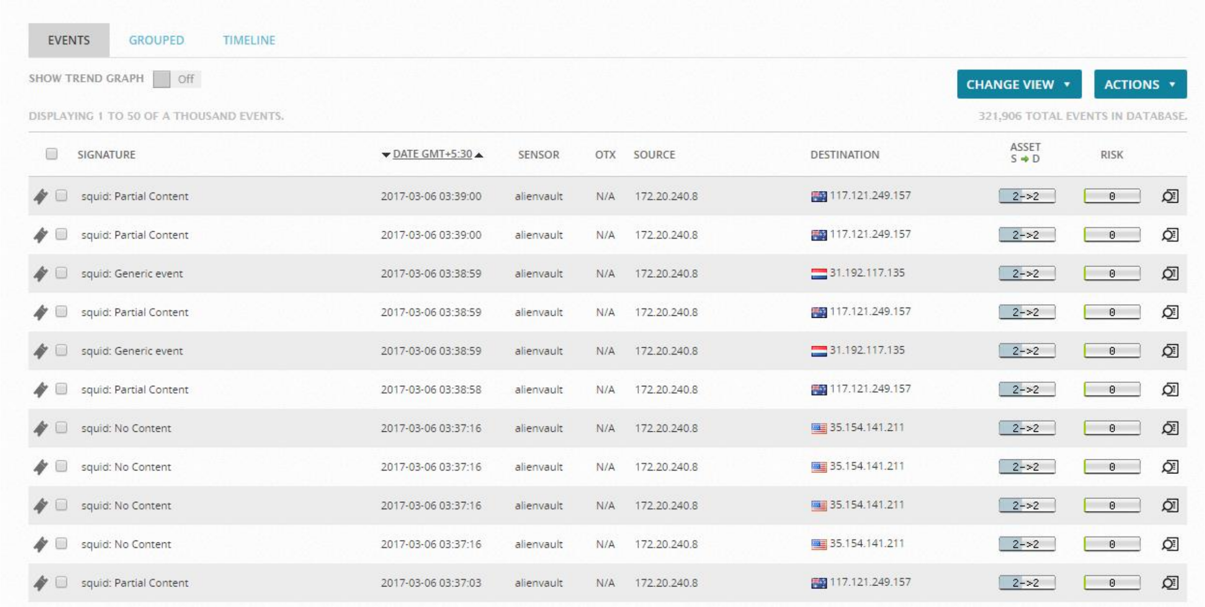


DISPLAYING 1 TO 50 OF THOUSANDS OF EVENTS.

321,905 TOTAL EVENTS IN DATABASE.

□	SIGNATURE	DATE GMT+5:30	SENSOR	OTX	SOURCE	DESTINATION	ASSET S → D	RISK
□	iptables: Generic event	2017-03-05 21:53:41	alienvault	N/A	217.146.26.211:5938	172.20.1.253:62376	2->2	4
□	iptables: Generic event	2017-03-05 21:53:41	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:41	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	217.146.26.211:5938	172.20.1.253:62376	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	217.146.26.211:5938	172.20.1.253:62376	2->2	4
□	iptables: Generic event	2017-03-05 21:53:36	alienvault	N/A	172.20.240.8:62376	217.146.26.211:5938	2->2	4
□	iptables: Generic event	2017-03-05 21:53:27	alienvault	N/A	74.125.68.188:5228	172.20.1.253:60484	2->2	4
□	iptables: Generic event	2017-03-05 21:53:27	alienvault	N/A	172.20.240.8:60484	74.125.68.188:5228	2->2	4

Figure 21 – Iptables log events in OSSIM Web Interface



DISPLAYING 1 TO 50 OF A THOUSAND EVENTS.

321,906 TOTAL EVENTS IN DATABASE.

□	SIGNATURE	DATE GMT+5:30	SENSOR	OTX	SOURCE	DESTINATION	ASSET S → D	RISK
□	squid: Partial Content	2017-03-06 03:39:00	alienvault	N/A	172.20.240.8	117.121.249.157	2->2	0
□	squid: Partial Content	2017-03-06 03:39:00	alienvault	N/A	172.20.240.8	117.121.249.157	2->2	0
□	squid: Generic event	2017-03-06 03:38:59	alienvault	N/A	172.20.240.8	31.192.117.135	2->2	0
□	squid: Partial Content	2017-03-06 03:38:59	alienvault	N/A	172.20.240.8	117.121.249.157	2->2	0
□	squid: Generic event	2017-03-06 03:38:59	alienvault	N/A	172.20.240.8	31.192.117.135	2->2	0
□	squid: Partial Content	2017-03-06 03:38:58	alienvault	N/A	172.20.240.8	117.121.249.157	2->2	0
□	squid: No Content	2017-03-06 03:37:16	alienvault	N/A	172.20.240.8	35.154.141.211	2->2	0
□	squid: No Content	2017-03-06 03:37:16	alienvault	N/A	172.20.240.8	35.154.141.211	2->2	0
□	squid: No Content	2017-03-06 03:37:16	alienvault	N/A	172.20.240.8	35.154.141.211	2->2	0
□	squid: No Content	2017-03-06 03:37:16	alienvault	N/A	172.20.240.8	35.154.141.211	2->2	0
□	squid: Partial Content	2017-03-06 03:37:03	alienvault	N/A	172.20.240.8	117.121.249.157	2->2	0

Figure 22 – Squid log events in OSSIM Web Interface

Once the sensors directives generate the above alerts against the event log messages, these alerts will be logged in */var/ossec/logs/alerts/alerts.log*

Below is how the iptables alert has been logged in alerts log

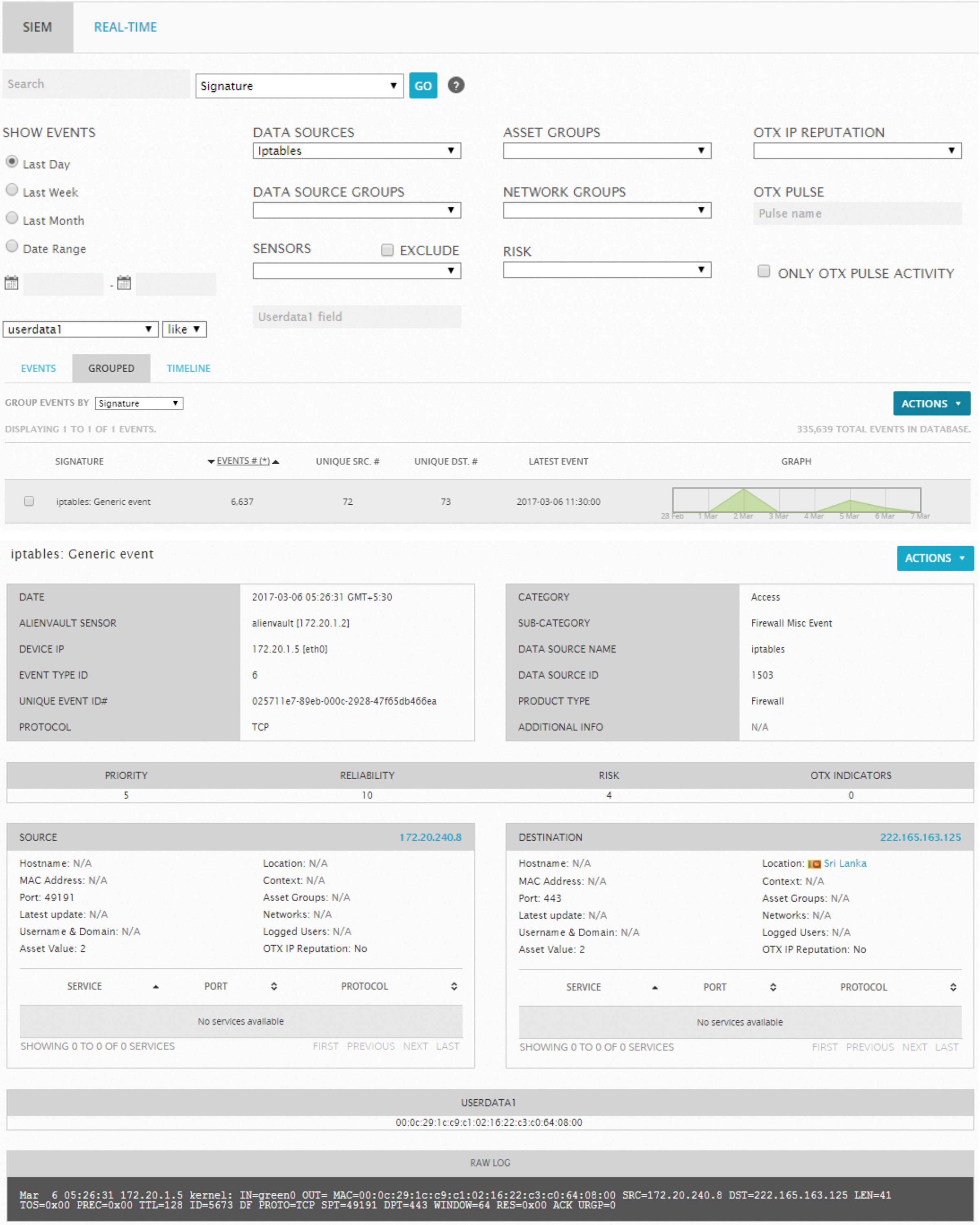
```
2017-03-06 15:39:13,588 Output [INFO]: event type="detector" date="1488794948"
device="172.20.1.5" interface="eth0" plugin_id="1503" plugin_sid="6" protocol="TCP"
src_ip="117.121.249.126" src_port="80" dst_ip="172.20.1.253" dst_port="42434"
userdata1="MDA6MGM6Mjk6MWM6Yzk6Y2I6Zjg6ZDE6MTE6ZTM6M2U6ZmE6MDg6M
DA="
log="TWFyICA2IDE1OjM5OjA4IDE3Mi4yMC4xLjUga2VybmVsOiBJTj1yZWQwIE9VVD0g
TUFDPTAwOjBjOjI5OjFjOmM5OmNiOmY4OmQxOjExOmUzOjNlOmZhOjA4OjAwIFNSQz
0xMTcuMTIxLjI0OS4xMjYgRFNUPTE3Mi4yMC4xLjI1MyBMRU49MTQ4MCBUT1M9MHg
wMCBQUkVDPTB4MDAgVFRMPTU5IElEPTU2NDYgREYgUFJPVE89VENQIFNQVD04
MCBEUFQ9NDI0MzQgV0lORE9XPTEyNTg0IFJFUz0weDAwIEFDSyBVUkdQPTAg"
fdate="2017-03-06 10:09:08" tzone="5.5" event_id="02ac11e7-b8f2-000c-2928-
47f6f594b1a4"
```

Below is how a squid alert has been logged in alerts log

```
2017-03-06 15:52:35,833 Output [INFO]: event type="detector" date="1488795755"
device="172.20.1.2" interface="eth0" plugin_id="1553" plugin_sid="304"
src_ip="172.20.240.8" dst_ip="104.75.84.18" userdata1="MzA0"
log="TWFyICA2IDE1OjUyOjM1IDE3Mi4yMC4xLjUgKHNxdWlkLTEpOiAxNDg4ODMzNT
U1LjgzNSAgICAgMTcgMTcyLjIwLjI0MC44IFRDUF9NSVNTLzMwNCAzMTUgR0VUIGh0d
HA6Ly9jcmwubWljcm9zb2Z0LmNvbS9wa2kvY3JsL3Byb2R1Y3RzL01pY3Jvc29mdFRpbWVT
dGFtcFBDQS5jcmwgLSBPUklHSU5BTF9EU1QvMTA0Ljc1Ljg0LjE4IGFwcGxpY2F0aW9u
L3BraXgtY3JsIA==" fdate="2017-03-06 10:22:35" tzone="5.5" event_id="02ae11e7-b378-
000c-2928-47f6d3c192d4"
```

5.5 Results shown in the OSSIM web interface

Confirmation of the receipt of event logs sent from the gateway server to OSSIM server, and if the configurations are absolutely correct, the results will be shown in the OSSIM web interface.



Signature ▾

DATA SOURCES: Iptables

userdata1 ▾ like ▾

GROUP EVENTS BY Signature ▾

DISPLAYING 1 TO 1 OF 1 EVENTS.

335,639 TOTAL EVENTS IN DATABASE.

	SIGNATURE	▾EVENTS # (*)▴	UNIQUE SRC. #	UNIQUE DST. #	LATEST EVENT	GRAPH
□	iptables: Generic event	6,637	72	73	2017-03-06 11:30:00	

iptables: Generic event

DATE	2017-03-06 05:26:31 GMT+5:30
ALIENVAULT SENSOR	alienvault [172.20.1.2]
DEVICE IP	172.20.1.5 [eth0]
EVENT TYPE ID	6
UNIQUE EVENT ID#	025711e7-89eb-000c-2928-47f65db466ea
PROTOCOL	TCP

CATEGORY	Access
SUB-CATEGORY	Firewall Misc Event
DATA SOURCE NAME	iptables
DATA SOURCE ID	1503
PRODUCT TYPE	Firewall
ADDITIONAL INFO	N/A

PRIORITY	RELIABILITY	RISK	OTX INDICATORS
5	10	4	0

SOURCE 172.20.240.8

Hostname: N/A
MAC Address: N/A
Port: 49191
Latest update: N/A
Username & Domain: N/A
Asset Value: 2

Location: N/A
Context: N/A
Asset Groups: N/A
Networks: N/A
Logged Users: N/A
OTX IP Reputation: No

SERVICE ▴	PORT ⇕	PROTOCOL ⇕
No services available		

SHOWING 0 TO 0 OF 0 SERVICES FIRST PREVIOUS NEXT LAST

DESTINATION 222.165.163.125

Hostname: N/A
MAC Address: N/A
Port: 443
Latest update: N/A
Username & Domain: N/A
Asset Value: 2

Location: Sri Lanka
Context: N/A
Asset Groups: N/A
Networks: N/A
Logged Users: N/A
OTX IP Reputation: No

SERVICE ▴	PORT ⇕	PROTOCOL ⇕
No services available		

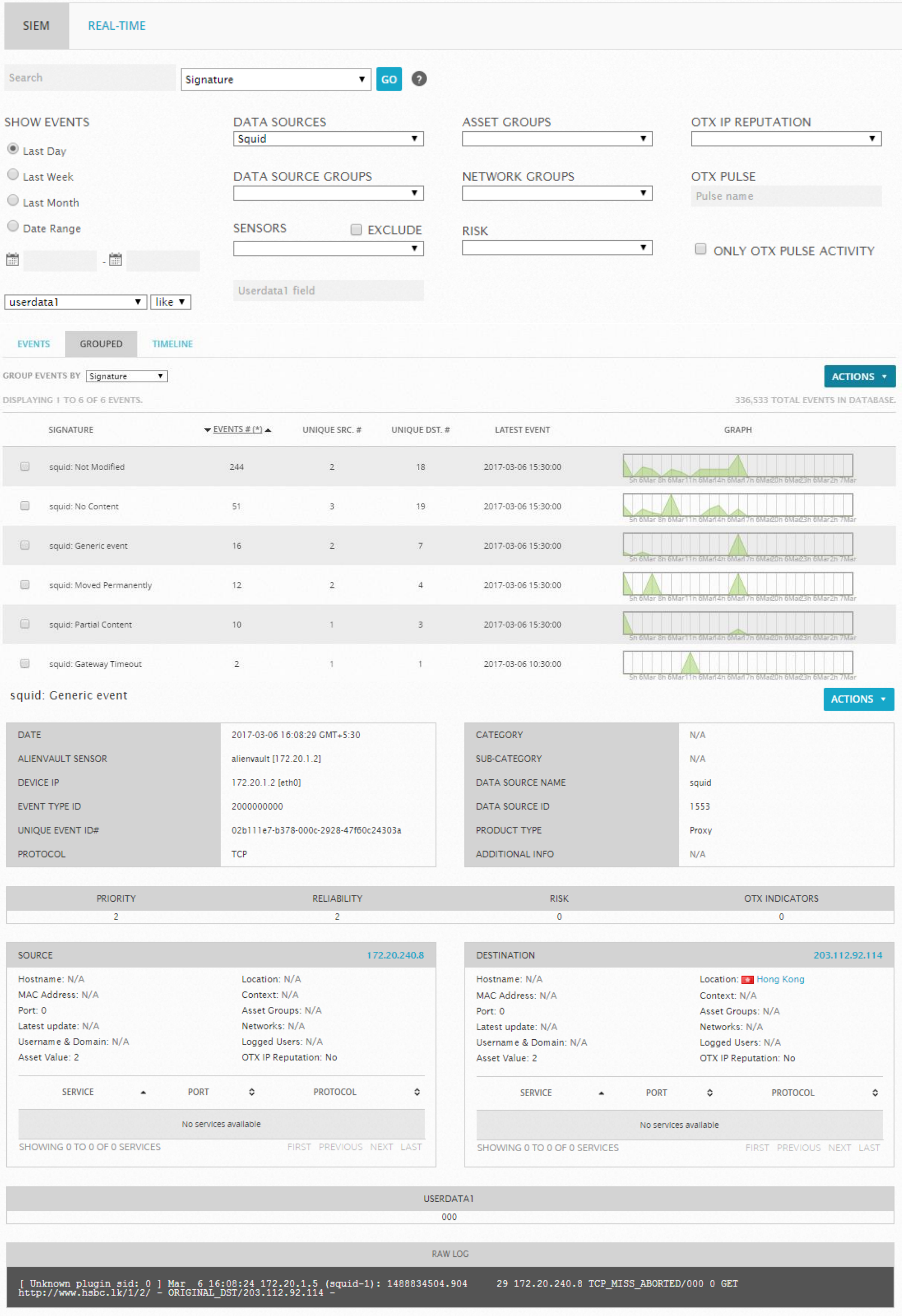
SHOWING 0 TO 0 OF 0 SERVICES FIRST PREVIOUS NEXT LAST

USERDATA1
00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00

RAW LOG

```
Mar  6 05:26:31 172.20.1.5 kernel: IN=green0 OUT= MAC=00:0c:29:1c:c9:c1:02:16:22:c3:c0:64:08:00 SRC=172.20.240.8 DST=222.165.163.125 LEN=41
TOS=0x00 PREC=0x00 TTL=128 ID=5673 DF PROTO=TCP SPT=49191 DPT=443 WINDOW=64 RES=0x00 ACK URGP=0
```

Figure 23 – Iptables log event captured by OSSIM



Signature

SHOW EVENTS

● Last Day
○ Last Week
○ Last Month
○ Date Range

DATA SOURCES: Squid

DATA SOURCE GROUPS:

SENSORS: □ EXCLUDE

ASSET GROUPS:

NETWORK GROUPS:

RISK:

OTX IP REPUTATION:

OTX PULSE: Pulse name

□ ONLY OTX PULSE ACTIVITY

userdata1 like Userdata1 field

GROUP EVENTS BY Signature

DISPLAYING 1 TO 6 OF 6 EVENTS.

336,533 TOTAL EVENTS IN DATABASE.

	SIGNATURE	EVENTS # (*)	UNIQUE SRC. #	UNIQUE DST. #	LATEST EVENT	GRAPH
□	squid: Not Modified	244	2	18	2017-03-06 15:30:00	
□	squid: No Content	51	3	19	2017-03-06 15:30:00	
□	squid: Generic event	16	2	7	2017-03-06 15:30:00	
□	squid: Moved Permanently	12	2	4	2017-03-06 15:30:00	
□	squid: Partial Content	10	1	3	2017-03-06 15:30:00	
□	squid: Gateway Timeout	2	1	1	2017-03-06 10:30:00	

5h 6Mar 8h 6Mar11h 6Mar14h 6Mar17h 6Mar20h 6Mar23h 6Mar2h 7Mar

squid: Generic event

DATE	2017-03-06 16:08:29 GMT+5:30
ALIENVAULT SENSOR	alienvault [172.20.1.2]
DEVICE IP	172.20.1.2 [eth0]
EVENT TYPE ID	2000000000
UNIQUE EVENT ID#	02b111e7-b378-000c-2928-47f60c24303a
PROTOCOL	TCP

CATEGORY	N/A
SUB-CATEGORY	N/A
DATA SOURCE NAME	squid
DATA SOURCE ID	1553
PRODUCT TYPE	Proxy
ADDITIONAL INFO	N/A

PRIORITY	RELIABILITY	RISK	OTX INDICATORS
2	2	0	0

SOURCE 172.20.240.8

Hostname: N/A	Location: N/A
MAC Address: N/A	Context: N/A
Port: 0	Asset Groups: N/A
Latest update: N/A	Networks: N/A
Username & Domain: N/A	Logged Users: N/A
Asset Value: 2	OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES FIRST PREVIOUS NEXT LAST

DESTINATION 203.112.92.114

Hostname: N/A	Location: Hong Kong
MAC Address: N/A	Context: N/A
Port: 0	Asset Groups: N/A
Latest update: N/A	Networks: N/A
Username & Domain: N/A	Logged Users: N/A
Asset Value: 2	OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES FIRST PREVIOUS NEXT LAST

USERDATA1
000

RAW LOG

```
[ Unknown plugin sid: 0 ] Mar  6 16:08:24 172.20.1.5 (squid-1): 1488834504.904     29 172.20.240.8 TCP_MISS_ABORTED/000 0 GET
http://www.hsbc.lk/1/2/ - ORIGINAL_DST/203.112.92.114 -
```

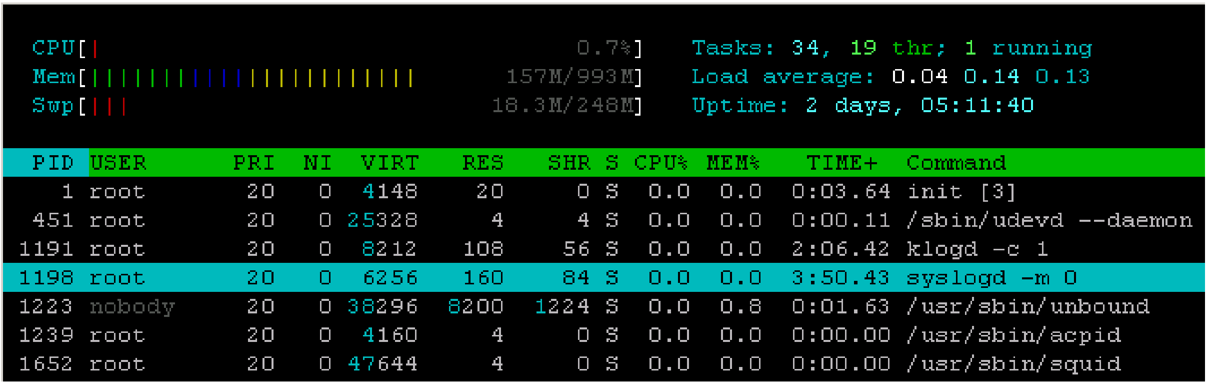
Figure 24 – Squid log event captured by OSSIM

5.6 Server resource utilisation at the gateway server

Table 3 – Syslog Resource Utilisation

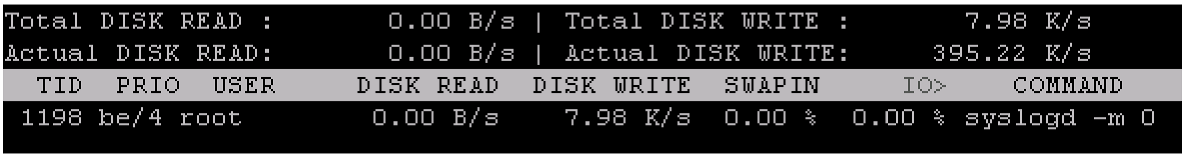
Resource	**Syslog Utilisation (Average)**
CPU Utilisation	0.10%
RAM Utilisation	0.01%
Disk Utilisation (Read)	0%
Disk Utilisation (Write)	5B/s
Network Utilisation	6.46Kb/s

Below are some of the screenshots which shows the resource utilisation by the syslog in the gateway server. This information was gathered with the help of using htop, iotop and iftop which are linux based monitoring tools.



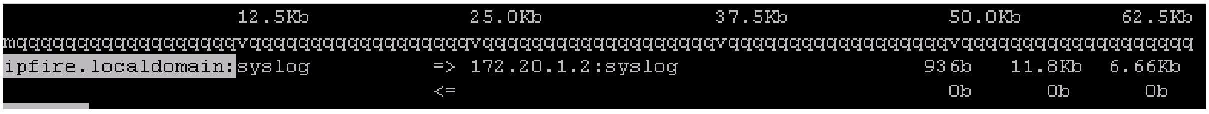
```
  CPU[|                                    0.7%]    Tasks: 34, 19 thr; 1 running
  Mem[|||||||||||||||||||||||         157M/993M]    Load average: 0.04 0.14 0.13
  Swp[|||                             18.3M/248M]    Uptime: 2 days, 05:11:40

  PID USER        PRI  NI  VIRT   RES    SHR S CPU% MEM%   TIME+  Command
    1 root         20   0  4148    20      0 S  0.0  0.0  0:03.64 init [3]
  451 root         20   0 25328     4      4 S  0.0  0.0  0:00.11 /sbin/udevd --daemon
 1191 root         20   0  8212   108     56 S  0.0  0.0  2:06.42 klogd -c 1
 1198 root         20   0  6256   160     84 S  0.0  0.0  3:50.43 syslogd -m 0
 1223 nobody       20   0 38296  8200   1224 S  0.0  0.8  0:01.63 /usr/sbin/unbound
 1239 root         20   0  4160     4      0 S  0.0  0.0  0:00.00 /usr/sbin/acpid
 1652 root         20   0 47644     4      0 S  0.0  0.0  0:00.00 /usr/sbin/squid
```

Figure 25 – CPU Usage by Syslog



```
Total DISK READ :        0.00 B/s | Total DISK WRITE :        7.98 K/s
Actual DISK READ:        0.00 B/s | Actual DISK WRITE:      395.22 K/s
  TID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO>    COMMAND
 1198 be/4 root         0.00 B/s    7.98 K/s  0.00 %  0.00 % syslogd -m 0
```

Figure 26 – Disk Read/Write Usage by Syslog



```
                 12.5Kb              25.0Kb              37.5Kb              50.0Kb         62.5Kb
mqqqqqqqqqqqqqqqqqvqqqqqqqqqqqqqqqqqvqqqqqqqqqqqqqqqqqvqqqqqqqqqqqqqqqqqvqqqqqqqqqqqqqqqqq
ipfire.localdomain:syslog          => 172.20.1.2:syslog                     936b   11.8Kb  6.66Kb
                                   <=                                         0b       0b      0b
```

Figure 27 – Network Usage by Syslog

5.7 Tools used in the evaluation phase

During the evaluation phase, open source linux based tools have been used. These tools are publicly available.

Table 4 – Tools used in the Evaluation

Tool name	**Operation**
regexp.py	Using this python script, the regular expressions in the plugin files can be validated
htop	htop is a altervative tool to the unix top tool. Using htop, the system processes can be monitored and this tool is interactive
iotop	Using iotop, the hard disk Input/Output usage can be monitored
iftop	iftop is monitoring tool which can be used to monitor the network traffic generated through the network interface
reputation.data	This is a database of malicious hosts which OSSIM will correlate against the event logs received from remote hosts

6 Conclusion and Future Work

6.1 Summery

In this thesis main objective was to find out the current problems in BYOD in terms of security threats and potential risks within the corporate network and corporate perimeter and to find out a method to prevent security loop wholes which are identified performing a mutual comparison and relate BYOD network traffic and log records.

In the present era of information technology, the technology rapidly changes and improves daily basis. The public is majorly relying on more technological systems in order perform their daily activities (which are good in my opinion) and complete their tasks in an easier manner. However, knowing the fact that the information technology is improving at a fast track, there are more vulnerable activities taking place. Compared to the past, now we quite frequently hear about cyber attacks through the media such hacking, cracking, stealing intellectual property. In parallel to the development of the information technology the security loopholes should also be addressed.

The BYOD concept is approved and implemented in many organisations at present. The reason behind for the corporate environment to adopt BYOD is mainly to reduce the infrastructure cost. Also, many organisations are in the agreement of the opinion that the employee effectiveness and efficiency is incremented when they are using the devices which they are familiar with. Therefore, we can see that employees in any organisation will connect their devices to the corporate network for various reasons.

The biggest asset that any organisation will have is their intellectual property, trade secrets and other classified information such as employee details and project details. If this information is compromised with unwanted parties the damage to the organisation is unpredictable. This can even led a company to cease their operations.

The attacker can find a path to get into the organisation perimeter to perform a destructive action when there are plenty of vulnerabilities available in the infrastructure. Especially when an organisation is practising BYOD concept, these attacks can be much easier to the attacker if correct security measurements has not been taken place. Therefore the concept of SIEM has been introduced to be alerted about suspicious activities in the corporate network.

However, even if a SIEM has been implemented in an organisation, the SIEM monitoring staff or the network and system administrators can be misled by false positive alarms generated by the SIEM solutions. In order to overcome this overhead, the security professionals later introduced the event log correlation as a preventive measure.

After the introduction of event log correlation solution providers came up with many effective products to monitor the network behaviours and correlate the event and alert to the administrators. In terms of the cost of these products it will be massive amount of investment. This is not pragmatic for small and medium scale organisation. Furthermore, in some organisations the administrators struggle when they have to install 3rd party applications, modify network and firewall rules when the organisation management does not allow to perform with a prior approval.

Therefore, my implementation is constructed with commonly available tools for operating systems and with the help of OSSIM SIEM solution provided by Alien Vault.

6.2 Findings

During the implementation phase it was understood that the squid logs cannot be sent to a remote server by squid daemon. Therefore squid event logs were captured locally by the syslog agent running on the gateway server. And then these logs were transmitted to the OSSIM SIEM server through the syslog daemon. There can be a cost involved in the method since the squid event log messages are being sent to the OSSIM server through one or more hops.

6.3 Future work

Securing the syslog messages

The syslog transmission from the local host to the remote host is using UDP traffic. This is not a secure connection. Hence, it is vulnerable for attacks such as man-in-the-middle attack Therefore, syslog transmission can be improved to a key encryption and decryption mechanism.

Automated event response

Once OSSIM SIEM recognises a suspicious connection being made to a malicious host, the OSSIM SIEM can trigger an action to drop or reject the suspicious connection from the gateway server by adjusting the configurations on the go. This can be possibly done using shell script being executed at the OSSIM server to change the configuration in the gateway server. However, the connection made to between the servers should be secure since it holds the configuration changes.

7 References

[1] Trend Micro. 2012. *Consumerization Survey Report*. [ONLINE] Available at: https://www.trendmicro.de/cloud-content/us/pdfs/rpt_consumerization-survey-report.pdf. [Accessed August 2016].

[2] Meisam Eslahi, Maryam Var Naseri, H. Hashim, N.M. Tahir, Ezril Hisham , Mat Saad R. 2014. *BYOD:Current State and Security Challenges*. [ONLINE] Available at: https://www.researchgate.net/publication/261871646_BYODCurrent_State_and_Security_Challenges. [Accessed 10 August 2016].

[3] Zoran Mitrovic, Ivan Veljkovic, Grafton Whyte, Kevin Thompson. 2014. *Introducing BYOD in an organisation: the risk and customer services viewpoints*. [ONLINE] Available at: http://ir.polytechnic.edu.na/bitstream/handle/10628/522/. [Accessed August 2016].

[4] Robert Ogie. 2016. *Bring your own device: an overview of risk assessment.* [ONLINE] Available at: http://ro.uow.edu.au/eispapers/5418/. [Accessed August 2016].

[5] Francis Nwebonyi. 2013. *An Access Control System to Improve Security Amongst Randomly Associated Nodes in BYOD Network* [ONLINE] Available at: https://core.ac.uk/download/pdf/29821692.pdf. [Accessed September 2016].

[6] Misun Song, Kyungho Lee. 2014. *Proposal of MDM Management Framework for BYOD use of Large Companies* [ONLINE] Available at: http://www.sersc.org/journals/IJSH/vol8_no1_2014/13.pdf. [Accessed September 2016].

[7] Abubakar Bello Garba, Jocelyn Armarego, David Murray. 2015. *A Policy-Based Framework for Managing Information Security and Privacy Risks in BYOD Environments* [ONLINE] Available at: http://www.ijettcs.org/pabstract.php?vol=Volume4Issue2&pid=IJETTCS-2015-04-23-122. [Accessed September 2016].

[8] Daniel Arregui. 2016. *Mitigating BYOD information security risks* [ONLINE] Available at: https://business.uow.edu.au/content/groups/public/@web/@bus/documents/doc/uow223871.pdf. [Accessed September 2016].

[9] Cloudessa RADIUS Service | Cloudessa. 2016. *Cloudessa RADIUS Service | Cloudessa*. [ONLINE] Available at: http://cloudessa.com/products/cloudessa-radius-service/. [Accessed 10 August 2016].

[10] eBrahma 2016. *Configure System Logging in SRX Device - eBrahma*. [ONLINE] Available at: http://www.ebrahma.com/2015/09/configure-system-logging-in-srx-device/. [Accessed 10 August 2016].

[11] OSSIM 2016. *OSSIM a Careful, Free and Always Available Guardian for Your Network* . [ONLINE] Available at: http://nnc3.com/mags/LJ_1994-2014/LJ/242/11676.html. [Accessed 10 August 2016].

[12] ISACA 2013. *Advanced Persistent Threat Awareness* [ONLINE] Available at: http://www.trendmicro.ie/media/misc/apt-survey-report-en.pdf [Accessed 10 August 2016]

[13] InfoSec Resources. 2016. *AlienVault OSSIM Review - Open Source SIEM*. [ONLINE] Available at: http://resources.infosecinstitute.com/alienvault-ossim-review-open-source-siem/#gref. [Accessed 10 August 2016].

[14] IETF 2009. *RFC 5424 - The Syslog Protocol*. [ONLINE] Available at: https://tools.ietf.org/html/rfc5424. [Accessed 10 August 2016].

8 APPENDIX A

Regex.py (Regular Expression Script)

```
#!/usr/bin/python
#   - Match rules from .cfg in the same order as the Agent does
#   - Count and ignore null lines
#   - Fixed aliases translation, reading AV definitions
#   - Fixed "y" modifier
#   - Plugin file can have any extension
#   - Deleted "number" modifier, no clue what it does
#       (ok, is for mono-regex)
# TODO
#   - Make multi-line
#   - Fix error with null lines when mixing EOLs (win/*nix/osx);
#       maybe look for NEWLINE parameter on OPEN method
#   - Fix multiline regex: it does not match "newlines" with "."
import sys,re
import ConfigParser
from os.path import isfile
##############################            Function            definitions
###########################

def hitems(config, section):
    hash = {}
    for item in config.items(section):
        hash[item[0]] = _strip_value(item[1])
    return hash
def _strip_value(value):
    from string import strip
    return strip(strip(value, '"'), "'")
def get_entry(config, section, option):
    value = config.get(section, option)
    value = _strip_value(value)
    return value
def translate_aliases(regex):
    for alias in aliases:
        tmp_al = ""
        tmp_al = "\\" + alias;
        regex = regex.replace(tmp_al,aliases[alias])
    return regex
############################## End definitions ###########################

##############################             Aliases             definitions
###########################
aliases = {}
if isfile('/etc/ossim/agent/aliases.cfg'):
    try:
        aliases_file = open('/etc/ossim/agent/aliases.cfg', mode='rU')
    except Exception:
        print "[W] Aliases file can not be opened."
    else:
        for line in aliases_file.readlines():
            if line[0] in ('\s', '#', '[', '\n', ';'):
                    continue
            else:
                    (alias_name, alias_value) = line.split('=',1)
                    alias_value = alias_value.strip()
                    aliases[alias_name]=alias_value
else:
    print "[W] Aliases file does not exist, using defaults"
    aliases['IPV4']="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"
    aliases['IPV6_MAP']="::ffff:\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"
    aliases['MAC']="\w{1,2}:\w{1,2}:\w{1,2}:\w{1,2}:\w{1,2}:\w{1,2}"
    aliases['PORT']="\d{1,5}"
    aliases['HOSTNAME']="((([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-]*[a-zA-Z0-
9])\.)([a-zA-Z])+)"
    aliases['TIME']="\d\d:\d\d:\d\d"
    aliases['SYSLOG_DATE']="\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d"
    aliases['SYSLOG_WY_DATE']="\w+\s+\d{1,2}\s\d{4}\s\d\d:\d\d:\d\d"
############################## End definitions ###########################

try:
   tmp = sys.argv[3]
except:
    print "\n\t%s log_filename regexp modifier" % sys.argv[0]
    print  "\n\tmodifier  can  be  V/v/y/n/a  number  indicating  the  offset  to
show"
    print "\ty --> show not matched lines"
    print "\tn --> do not show not matched lines"
    print "\tnumber --> Show $number"
    print "\tv --> verbose, show matching line"
    print "\tV --> vverbose, show matching line and regexp"
    print "\tq --> quiet, just show a summary"
    print  "\tIf  regexp  is  a  plugin  file  all  regexps  in  that  file  will  be
checked\n"
    sys.exit()
try:
    f = open(sys.argv[1], mode='rU')
except Exception:
    print "[E] Log file cannot be opened."
    sys.exit(-1)
if sys.argv[3] not in ('y','n','v','V','q'):
    print "[E] Modifier not found."
    sys.exit(-1)
data = f.readlines()
cfg_file=exp=sys.argv[2]
single_regexp=True
if isfile(cfg_file):
    single_regexp=False
    print "Multiple regexp mode used, parsing %s " % exp
else:
    if exp.endswith(".cfg") or exp.endswith(".cfg.local"):
        print "[E] Plugin file does not exist."
        sys.exit(-1)

line_match = 0
```

```
matched = 0
nulls = 0

if single_regexp == True:
    # single regexp mode
    multiline = False
    for line_index in range(0, len(data)):
        line = data[line_index]
        if multiline:
            if line_index != new_line_index:
                continue
            else:
                multiline = False
        if line == '\n':
            nulls += 1
            continue
        if exp.find('\\n') != -1 and re.search( "^"+exp.split('\\n')[0],
line, re.S):
            multiline = True
            exp = exp.rstrip('\\n')
            multiline_index = exp.count('\\n')
            for a in range(1, multiline_index+1):
                line += data[line_index+a]
            line = line.rstrip('\n')
            exp = exp.replace('\\n', '\n')
            new_line_index = line_index + multiline_index + 1
        exp = translate_aliases(exp)
        result = re.findall(exp,line)
        try:
            tmp = result[0]
        except IndexError:
            if sys.argv[3] is "y":
                print "Not matched:", line
            continue
        # Matched
        if sys.argv[3] is "v":
            print line.replace('\n', '\\n')
        if sys.argv[3] is "V":
            print "Regexp: ", exp.replace('\n', '\\n')
            print "Line: ", line.replace('\n', '\\n')
        try:
            if int(sys.argv[3]) > 0:
                print "Match $%d: %s" %
(int(sys.argv[3]),tmp[int(sys.argv[3])-1])
                #print "Match %d: %s" %
(int(sys.argv[3]),result[int(sys.argv[3])])
            else:
                if sys.argv[3] is not "q":
                    print "Result: ", result
        except ValueError:
            if sys.argv[3] is not "q":
                print "Result: ", result
        matched += 1
    print "Counted", len(data), "lines."
    print "Matched", matched, "lines."
else:
    SECTIONS_NOT_RULES = ["config", "info", "translation"]
    rules = {}
    sorted_rules = {}
    rule_stats = []
    # .cfg file mode
    config = ConfigParser.RawConfigParser()
    config.read(cfg_file)
    for section in config.sections():
        if section.lower() not in SECTIONS_NOT_RULES :
            rules[section] = hitems(config,section)
    keys = rules.keys()
    keys.sort()
    multiline = False
    for line_index in range(0,len(data)):
        line = data[line_index]
        if multiline:
            if line_index != new_line_index:
                continue
            else:
                multiline = False
        if line == '\n':
            nulls += 1
            continue
        line_errors = 0
        for rule in sorted(rules.iterkeys()):
            rulename = rule
            regexp = get_entry(config, rule, 'regexp')
            if regexp is "":
                continue
            #elif regexp.find('\\n') != -1 and line.startswith(
regexp.split('\\n')[0] ):
            elif regexp.find('\\n') != -1 and re.search(
"^"+exp.split('\\n')[0], line, re.S):
                multiline = True
                regexp = regexp.rstrip('\\n')
                multiline_index = regexp.count('\\n')
                for a in range(1, multiline_index+1):
                    line += data[line_index+a]
                line = line.rstrip('\n')
                regexp = regexp.replace('\\n', '\n')
                new_line_index = line_index + multiline_index + 1

            # Replace vars
            regexp = translate_aliases(regexp)
            result = re.findall(regexp,line)
            try:
                tmp = result[0]
            except IndexError:
                line_errors += 1
                continue
            # Matched
            if sys.argv[3] is not 'y':
                if sys.argv[3] is not "q":
                    print
                    print "Matched using %s" % (rulename)
                if sys.argv[3] is "v":
                    print line.replace('\n', '\\n')
                if sys.argv[3] is "V":
                    print regexp.replace('\n', '\\n')
                    print line.replace('\n', '\\n')
                try:
                    if int(sys.argv[3]) > 0:
                        print "Match $%d: %s" %
(int(sys.argv[3]),tmp[int(sys.argv[3])-1])
                    else:
                        if sys.argv[3] is not "q":
                            print result
                except ValueError:
                    if sys.argv[3] is not "q":
                        print result
            # Do not match more rules for this line
            rule_stats.append(str(rulename))
            matched += 1
            break
        if sys.argv[3] is 'y' and line_errors is len(rules.keys()):
            print line

    print "-----------------------------------------------------------------
--------------"

    for key in keys:
        print "Rule: \t%s\n\t\t\t\t\t\tMatched %d times" % (str(key),
rule_stats.count(str(key)))

    print "Counted", len(data), "lines."
    print "Matched", matched, "lines."
    print "Ignored", nulls, "blank lines."


# vim: tabstop=8 expandtab shiftwidth=4 softtabstop=4:
```

9 APPENDIX B

Squid Configuration

```
# Do not modify '/var/ipfire/proxy/squid.conf' directly since any changes
# you make will be overwritten whenever you resave proxy settings using
the
# web interface!
#
# Instead, modify the file '/var/ipfire/proxy/advanced/acls/include.acl'
and
# then restart the proxy service using the web interface. Changes made to
the
# 'include.acl' file will propagate to the 'squid.conf' file at that time.
# Yehan Gunaratne yehan_gunaratne@hotmail.com

shutdown_lifetime 5 seconds
icp_port 0

http_port 172.20.1.5:800
http_port 172.20.1.5:3128 intercept


cache_effective_user squid
umask 022

pid_filename /var/run/squid.pid

cache_mem 2 MB
error_directory /usr/lib/squid/errors/en

digest_generation off

acl SSL_ports port 443 # https
acl SSL_ports port 563 # snews
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 563 # snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl Safe_ports port 800 # Squids port (for icons)

acl IPFire_http  port 81
acl IPFire_https port 444
acl IPFire_ips              dst 172.20.1.5
acl     IPFire_networks                                            src
"/var/ipfire/proxy/advanced/acls/src_subnets.acl"
```

```
acl     IPFire_servers                                              dst
"/var/ipfire/proxy/advanced/acls/src_subnets.acl"
acl IPFire_green_network    src 172.20.0.0/16
acl IPFire_green_servers    dst 172.20.0.0/16
acl CONNECT method CONNECT
maximum_object_size 4096 KB
minimum_object_size 0 KB

cache_dir aufs /var/log/cache 50 16 256
request_body_max_size 0 KB
access_log stdio:/var/log/squid/access.log
#access_log stdio:/var/log/messages
#access_log stdio:/var/log/messages
cache_log /var/log/squid/cache.log
cache_store_log none
access_log stdio:/var/log/squid/user_agent.log useragent
#access_log syslog squid
access_log syslog:LOG_LOCAL4

strip_query_terms off

log_mime_hdrs off
forwarded_for off
via off

acl within_timeframe time MTWHFAS 00:00-24:00


#Access to squid:
#local machine, no restriction
http_access allow         localhost

#GUI admin if local machine connects
http_access allow         IPFire_ips IPFire_networks IPFire_http
http_access allow CONNECT IPFire_ips IPFire_networks IPFire_https

#Deny not web services
http_access deny          !Safe_ports
http_access deny  CONNECT !SSL_ports
#Set custom configured ACLs
http_access allow IPFire_networks within_timeframe
http_access deny  all

#Strip HTTP Header
request_header_access X-Forwarded-For deny all
reply_header_access X-Forwarded-For deny all
request_header_access Via deny all
reply_header_access Via deny all

visible_hostname ipfire.localdomain


max_filedescriptors 16384
```

10 APPENDIX C

Iptables Configuration

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
 185K  152M BADTCP     tcp  --  any    any     anywhere
anywhere
 195K  155M CUSTOMINPUT  all  --  any    any     anywhere
anywhere
 195K  155M P2PBLOCK   all  --  any    any     anywhere
anywhere
 195K  155M GUARDIAN   all  --  any    any     anywhere
anywhere
    0     0 OVPNBLOCK  all  --  tun+   any     anywhere
anywhere
 195K  155M IPTVINPUT  all  --  any    any     anywhere
anywhere
 195K  155M ICMPINPUT  all  --  any    any     anywhere
anywhere
 195K  155M LOOPBACK   all  --  any    any     anywhere
anywhere
 191K  154M CONNTRACK  all  --  any    any     anywhere
anywhere
 2062  169K DHCPGREENINPUT  all  --  green0 any     anywhere
anywhere
 2474  251K GEOIPBLOCK  all  --  any    any     anywhere
anywhere
 2474  251K IPSECINPUT  all  --  any    any     anywhere
anywhere
 2474  251K GUIINPUT   all  --  any    any     anywhere
anywhere
 2474  251K WIRELESSINPUT  all  --  any    any     anywhere
anywhere             ctstate NEW
 2474  251K OVPNINPUT  all  --  any    any     anywhere
anywhere
 2474  251K TOR_INPUT  all  --  any    any     anywhere
anywhere
 2474  251K INPUTFW    all  --  any    any     anywhere
anywhere
 2474  251K REDINPUT   all  --  any    any     anywhere
anywhere
 2474  251K POLICYIN   all  --  any    any     anywhere
anywhere

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
 658K  592M BADTCP     tcp  --  any    any     anywhere
anywhere
 5019  279K TCPMSS     tcp  --  any    any     anywhere
anywhere             tcp flags:SYN,RST/SYN TCPMSS clamp to PMTU
```

```
1449K 1273M CUSTOMFORWARD  all  --  any    any     anywhere
anywhere
1449K 1273M P2PBLOCK   all  --  any    any     anywhere
anywhere
1449K 1273M GUARDIAN   all  --  any    any     anywhere
anywhere
1449K 1273M IPSECBLOCK  all  --  any    any     anywhere
anywhere             policy match dir out pol none
    0     0 OVPNBLOCK  all  --  tun+   any     anywhere
anywhere
    0     0 OVPNBLOCK  all  --  any    tun+    anywhere
anywhere
1449K 1273M IPTVFORWARD  all  --  any    any     anywhere
anywhere
1449K 1273M LOOPBACK   all  --  any    any     anywhere
anywhere
1449K 1273M CONNTRACK  all  --  any    any     anywhere
anywhere
 4792 2024K GEOIPBLOCK  all  --  any    any     anywhere
anywhere
 4792 2024K IPSECFORWARD  all  --  any    any     anywhere
anywhere
 4792 2024K WIRELESSFORWARD  all  --  any    any     anywhere
anywhere             ctstate NEW
 4792 2024K FORWARDFW  all  --  any    any     anywhere
anywhere
 4792 2024K UPNPFW     all  --  any    any     anywhere
anywhere             ctstate NEW
 4792 2024K REDFORWARD  all  --  any    any     anywhere
anywhere
 4792 2024K POLICYFWD  all  --  any    any     anywhere
anywhere

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
 192K  149M CUSTOMOUTPUT  all  --  any    any     anywhere
anywhere
 192K  149M P2PBLOCK   all  --  any    any     anywhere
anywhere
 192K  149M IPSECBLOCK  all  --  any    any     anywhere
anywhere             policy match dir out pol none
 192K  149M LOOPBACK   all  --  any    any     anywhere
anywhere
 188K  149M CONNTRACK  all  --  any    any     anywhere
anywhere
 6910 1877K DHCPGREENOUTPUT  all  --  any    green0  anywhere
anywhere
11364 2193K IPSECOUTPUT  all  --  any    any     anywhere
anywhere
11364 2193K OUTGOINGFW  all  --  any    any     anywhere
anywhere
11364 2193K POLICYOUT  all  --  any    any     anywhere
anywhere
```

```
Chain BADTCP (2 references)
 pkts bytes target     prot opt in     out     source
destination
   76 27254 RETURN     all  --  lo     any     anywhere
anywhere
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,PSH,URG
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,ACK,URG
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp
flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,PSH,ACK,URG
    1    52 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:SYN,RST/SYN,RST
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:FIN,SYN/FIN,SYN
    0     0 PSCAN      tcp  --  any    any     anywhere
anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
  297 88731 NEWNOTSYN  tcp  --  any    any     anywhere
anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN ctstate NEW

Chain CONNTRACK (3 references)
 pkts bytes target     prot opt in     out     source
destination
1808K 1571M ACCEPT     all  --  any    any     anywhere
anywhere             ctstate ESTABLISHED
  818 62545 DROP       all  --  any    any     anywhere
anywhere             ctstate INVALID
  357 32807 ACCEPT     icmp --  any    any     anywhere
anywhere             ctstate RELATED
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             ctstate RELATED helper match "sip"
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             ctstate RELATED helper match "h323"
    0     0 ACCEPT     tcp  --  any    any     anywhere
anywhere             ctstate RELATED helper match "ftp" tcp
dpts:1024:65535
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             ctstate RELATED helper match "tftp"
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             ctstate RELATED helper match "irc"

Chain CUSTOMFORWARD (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain CUSTOMINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain CUSTOMOUTPUT (1 references)
```

```
 pkts bytes target     prot opt in     out     source
destination

Chain DHCPBLUEINPUT (0 references)
 pkts bytes target     prot opt in     out     source
destination

Chain DHCPBLUEOUTPUT (0 references)
 pkts bytes target     prot opt in     out     source
destination

Chain DHCPGREENINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain DHCPGREENOUTPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain DHCPINPUT (0 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     udp  --  any    any     anywhere
anywhere             udp spt:bootpc dpt:bootps
    0     0 ACCEPT     tcp  --  any    any     anywhere
anywhere             tcp spt:bootpc dpt:bootps

Chain DHCPOUTPUT (0 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     udp  --  any    any     anywhere
anywhere             udp spt:bootps dpt:bootpc
    0     0 ACCEPT     tcp  --  any    any     anywhere
anywhere             tcp spt:bootps dpt:bootpc

Chain FORWARDFW (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain GEOIPBLOCK (2 references)
 pkts bytes target     prot opt in     out     source
destination

Chain GUARDIAN (2 references)
 pkts bytes target     prot opt in     out     source
destination

Chain GUIINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     tcp  --  green0 any     anywhere
anywhere             tcp dpt:snpp

Chain ICMPINPUT (1 references)
```

```
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     icmp --  any    any     anywhere
anywhere             icmp echo-request

Chain INPUTFW (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPSECBLOCK (2 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPSECFORWARD (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPSECINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPSECOUTPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPTVFORWARD (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain IPTVINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain LOG_DROP (0 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 LOG        all  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 LOG level warning
    0     0 DROP       all  --  any    any     anywhere
anywhere

Chain LOG_REJECT (0 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 LOG        all  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 LOG level warning
    0     0 REJECT     all  --  any    any     anywhere
anywhere             reject-with icmp-port-unreachable

Chain LOOPBACK (3 references)
 pkts bytes target     prot opt in     out     source
destination
 4501  557K ACCEPT     all  --  lo     any     anywhere
anywhere
```

```
 4501  557K ACCEPT     all  --  any    lo      anywhere
anywhere
    0     0 DROP       all  --  any    any     127.0.0.0/8
anywhere
    0     0 DROP       all  --  any    any     anywhere
127.0.0.0/8

Chain NEWNOTSYN (1 references)
 pkts bytes target     prot opt in     out     source
destination
  271 82724 LOG        all  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 LOG level warning prefix
"DROP_NEWNOTSYN "
  297 88731 DROP       all  --  any    any     anywhere
anywhere             /* DROP_NEWNOTSYN */

Chain OUTGOINGFW (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain OVPNBLOCK (3 references)
 pkts bytes target     prot opt in     out     source
destination
    0     0 RETURN     icmp --  any    any     anywhere
anywhere             ctstate RELATED

Chain OVPNINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain P2PBLOCK (3 references)
 pkts bytes target     prot opt in     out     source
destination

Chain POLICYFWD (1 references)
 pkts bytes target     prot opt in     out     source
destination
 4792 2024K ACCEPT     all  --  green0 any     172.20.0.0/16
anywhere
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             policy match dir in pol ipsec
    0     0 ACCEPT     all  --  tun+   any     anywhere
anywhere
    0     0 LOG        all  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 LOG level warning prefix
"DROP_FORWARD "
    0     0 DROP       all  --  any    any     anywhere
anywhere             /* DROP_FORWARD */

Chain POLICYIN (1 references)
 pkts bytes target     prot opt in     out     source
destination
 2062  169K ACCEPT     all  --  green0 any     anywhere
anywhere
```

```
    0     0 ACCEPT     all  --  any    any     anywhere
anywhere             policy match dir in pol ipsec
    0     0 ACCEPT     all  --  tun+   any     anywhere
anywhere
  192 43286 LOG        all  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 LOG level warning prefix
"DROP_INPUT "
  412 82115 DROP       all  --  any    any     anywhere
anywhere             /* DROP_INPUT */

Chain POLICYOUT (1 references)
 pkts bytes target     prot opt in     out     source
destination
11354 2192K ACCEPT     all  --  any    any     anywhere
anywhere
    0     0 DROP       all  --  any    any     anywhere
anywhere             /* DROP_OUTPUT */

Chain PSCAN (7 references)
 pkts bytes target     prot opt in     out     source
destination
    1    52 LOG        tcp  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 /* DROP_TCP PScan */ LOG
level warning prefix "DROP_TCP Scan "
    0     0 LOG        udp  --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 /* DROP_UDP PScan */ LOG
level warning prefix "DROP_UDP Scan "
    0     0 LOG        icmp --  any    any     anywhere
anywhere             limit: avg 10/min burst 5 /* DROP_ICMP PScan */ LOG
level warning prefix "DROP_ICMP Scan "
    0     0 LOG        all  -f  any    any     anywhere
anywhere             limit: avg 10/min burst 5 /* DROP_FRAG PScan */ LOG
level warning prefix "DROP_FRAG Scan "
    1    52 DROP       all  --  any    any     anywhere
anywhere             /* DROP_PScan */

Chain REDFORWARD (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain REDINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain TOR_INPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain UPNPFW (1 references)
 pkts bytes target     prot opt in     out     source
destination

Chain WIRELESSFORWARD (1 references)
```

```
 pkts bytes target     prot opt in     out     source
destination

Chain WIRELESSINPUT (1 references)
 pkts bytes target     prot opt in     out     source
destination
```