

# **Analysis of Cybersecurity**

(Bro and OSSEC)

Student: Jin Tae Kim (1485971)

Course: ISCG8047, Computing, Unitec

Lecturer: Bahman Sarrafpour

Due Date: 15 April 2017

## Table of Contents

|  |           |
|--|-----------|
| <b>1. Introduction .....</b>                                 | <b>3</b>  |
| <b>2. Security Assessment Tools .....</b>                    | <b>3</b>  |
| <b>3. Network Intrusion Detection System (NIDS).....</b>     | <b>4</b>  |
| <b>3.1 BRO .....</b>   | <b>4</b>  |
| <b>3.2 Why choose BRO NIDS .....</b>                         | <b>4</b>  |
| <b>3.3 Installing BRO on Ubuntu [2].....</b>                 | <b>6</b>  |
| <b>3.4 Analysis of BRO NIDS using the STRIDE model .....</b> | <b>10</b> |
| <b>3.5 Bro requirements to people and processes .....</b>    | <b>11</b> |
| <b>4. Host Intrusion Detection System (HIDS) .....</b>       | <b>11</b> |
| <b>4.1 OSSEC .....</b>                                       | <b>11</b> |
| <b>4.2 Key Advantages .....</b>                              | <b>11</b> |
| <b>4.3 Key Elements .....</b>                                | <b>12</b> |
| <b>4.4 Installing OSSEC [4].....</b>                         | <b>13</b> |
| <b>4.5 Configuration of OSSEC .....</b>                      | <b>17</b> |
| <b>4.6 Investigating logs of OSSEC.....</b>                  | <b>19</b> |
| <b>4.7 Analysis of OSSEC using the STRIDE model.....</b>     | <b>19</b> |
| <b>4.8 OSSEC requirements to people and processes.....</b>   | <b>19</b> |
| <b>5. Conclusion .....</b>                                   | <b>19</b> |
| <b>6. References.....</b>                                    | <b>20</b> |

## **1. Introduction**

With the advent of the new age computing on the internet, there has been an emergence of various security risks. This has brought about the need to have security frameworks in place to give general guidelines in mitigating these risks. With these standard has come various cyber security tools classified according to the layer of the network they are deployed. They include the Network Intrusion Detection Tools that monitor security on a network or a host-based IDS which mostly inspects security threats that affect a particular host only.

## **2. Security Assessment Tools**

Currently, there are several tools to assess security threats in a network or in hosts. Open source software is preferred since they give more control to the user since their source code is free to manipulate as a user desires. On security assessment, there are several categories one can assess their security from. Malicious users use networks to gain unauthorized control of other host computers. This necessitates it to have a Network Intrusion Detection System (NIDS). An NIDS works by keeping an eye on network activity and tries to identify any malicious activity occurring over a network media. Several open source software are available for NIDS [1]. They include Bro NIDS, Snort, Suricata to mention just a few. Another category of security assessment is a host. A computer system can be a security risk as a stand-alone client hence the need to have a Host Intrusion Detection System. This category usually tracks single hosts on a network. At times it could analyze packets over a network for suspicious activity. An example of a Host Detection software is OSSEC. Since the invention of the internet, there have been several security threats that have affected devices connected to the network. Internet protocols like HTTP have been a major target for these threats. This necessitates the need for a Web Application Firewall to protect applications using the HTTP protocol. An example of a web application firewall is mod\_security. In trying to mitigate against some of these security threats, decoy servers and hosts have been developed. These systems work by means of gathering information about a potentially malicious activity for analysis to prevent the actual attack. An example of an open source honeypot is Kippo. In the event a network security threat is identified, software is necessary to report these threats to security experts to enable proper action to be taken. Applications known as Security Information and Event Management software are developed to perform this task. An example is Cyberoam. Once in a while, a user would like to know about the vulnerability status of their network. A network vulnerability scanner is an application that finds weak points in your network that a malicious user can exploit to get access to your network. In this report, two categories were chosen as the main focus of discussion. Host detection system and Network Intrusion detection system and the software discussed are Bro NIDS and OSSEC.

### **3. Network Intrusion Detection System (NIDS)**

NIDS are a software application that work by monitoring traffic activity over a network and identifies potentially malicious activity. In large-scale networks, the NIDS can be installed in the backbone of the network for it to monitor every bit of traffic. However, in small networks, the NIDS can be installed in a server, router or switch. Additionally, NIDS servers could also be used to inspect files located in a system and trying to identify potentially malicious activity and maintain integrity in system files and data. Another core function is identifying variations in server components. NIDS servers also work as scanners to server logs and identify patterns that match known malicious activity that is related to system hacking. Apart from protectively getting involved in monitoring traffic, an NIDS can also play a proactive role in scanning firewalls or servers in a network in a bid to scan live packets [1].

There are two types of NIDS classified by their mode of function.

- i. Anomaly detections
- ii. Signature detection

Signature based detection works by using existing patterns of malicious activity. If a network activity occurs that has a signature of a known security threat, an alert is generated and sent to the security analyst. These alerts usually turn out to be malware server attacks etc.

Anomaly based detection usually works by relying on baselines. With time a particular network's activity is captured and a normal traffic distribution is identified. Anomaly based detection uses these baselines to identify security threats. Anomalies like increased or decreased traffic usually send a signal of a malicious activity like a Denial of service attack.

#### **3.1 BRO**

Bro NIDS is an open source tool that passively analyzes network traffic. It monitors network packets for any suspicious activity. It was developed by Vern Paxson. It is, however, being developed by developers and researchers at ICSI in Berkeley California.

#### **3.2 Why choose BRO NIDS**

Bro NIDS is adaptable in nature. At the core of BRO's code is a domain-specific scripting language. This means that it can monitor specific sites at will. BRO NIDS is also efficient in that it is mainly used for monitoring large sites and operationally targets networks with a high performance.

BRO NIDS is also flexible in that its approach to detection is not restricted to any traditional signatures. Its level of logging activity is also high. The logs are inclusive of any connection that is

made to on the network inclusive of transcripts from the application layer like MIME types, key headers, and HTTP sessions. The logs are written to separate log files that are well structured and are tabbed to necessitate their processing by external software. Its flexibility also allows users to output the logs in several formats like saving it to external databases.

The quickest advantage that a site picks up from conveying BRO NIDS is a broad arrangement of log documents that record a system's action in abnormal state terms. These logs incorporate not just a far-reaching record of each association seen on the wire, additionally application-layer transcripts, for example, e.g., all HTTP sessions with theirs asked for URIs, key headers, Emulate sorts, and server reactions; DNS asks for with answers; SSL endorsements; key substance of SMTP sessions; and substantially more. As a matter of course, BRO NIDS composes this data into very much organized tab-isolated log documents appropriate for post-preparing with outside programming. Clients can however additionally browsed an arrangement of option yield configurations and backends to interface specifically with, e.g., outside databases [2].

Notwithstanding the logs, BRO NIDS accompanies worked in usefulness for a scope of examination and identification errands, including extricating records from HTTP sessions, recognizing malware by interfacing to outer registries, announcing defenseless forms of programming seen on the system, distinguishing well known web applications, identifying SSH savage driving, approving SSL testament chains, and a great deal more.

In any case, the way to understanding BRO NIDS lies in understanding that despite the fact that the framework accompanies such intense usefulness out of the case, generally, it speaks to a stage for movement investigations that is completely adjustable and extensible: BRO NIDS furnishes clients with an area particular, Turing-finish scripting dialect for communicating discretionary examination undertakings. Theoretically, you can consider BRO NIDS an "area particular Python" (or Perl): simply like Python, the framework accompanies a vast arrangement of pre-fabricated usefulness (the "standard library"), yet you are not restricted to what the framework ships with but rather can put BRO NIDS to use in novel courses by composing your own particular code. To be sure, the majority of BRO NIDS's default investigations, including all the logging, is the aftereffect of such scripts; there's no particular examination hard-coded into the center of the framework.

BRO NIDS keeps running on ware equipment and consequently gives an ease contrasting option to costly exclusive arrangements. Regardless of the sticker price, in any case, BRO NIDS really goes long ways past the capacities of other system checking instruments, which commonly stay constrained to a little arrangement of hard-coded investigation assignments. We stress specifically that BRO NIDS is not a great mark based interruption discovery framework (IDS). While it backings such standard usefulness too, BRO NIDS's scripting dialect, in reality, encourages a considerably more

extensive range of altogether different ways to deal with finding vindictive movement, including semantic abuse discovery, irregularity identification, and behavioral investigation.

A substantial assortment of locales sends BRO NIDS operationally to protect their cyber infrastructure, including numerous colleges, inquire about labs, supercomputing focuses, open-science groups, and real organizations. BRO NIDS particularly targets rapid, high-volume arrange observing, and an expanding number of locales are presently utilizing the framework to screen their 10GE systems, with some effectively proceeding onward to 100GE connections. BRO NIDS obliges such elite settings by supporting adaptable load-adjusting: vast locales regularly run "BRO NIDS Groups" in which a fast frontend stack balancer circulates the movement over a fitting number of backend PCs, all running devoted BRO NIDS occurrences on their individual activity cuts. A focal supervisor framework arranges the procedure, synchronizing state over the backends and giving the administrators a focal administration interface for design and access to totaled logs. BRO NIDS's coordinated administration system, BroControl, backings such group setups out-of-the-case.

### **3.3 Installing BRO on Ubuntu [2]**

1. Bro requires the below libraries and tools.
  - i. Libpcap (<http://www.tcpdump.org>)
  - ii. OpenSSL libraries (<http://www.openssl.org>)
  - iii. BIND8 library
  - iv. Libz
  - v. Bash (for BroControl)
  - vi. Python 2.6 or greater (for BroControl)

Open terminal window and type below command to install all of dependencies.

```
$sudo apt-get install cmake make gcc g++ flex bison libcap-dev libgeoip-dev libssl-dev python-dev zlib1g-dev libmagic-dev libpcap0.8-dev swig2.0
```

```
Terminal File Edit View Search Terminal Help
jayk@ubuntu:~$ sudo apt-get install cmake make gcc g++ flex bison libcap-dev lib
geopip-dev libssl-dev python-dev zlib1g-dev libmagic-dev swig2.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version (4:5.3.1-1ubuntu1).
gcc is already the newest version (4:5.3.1-1ubuntu1).
make is already the newest version (4.1-6).
zlib1g-dev is already the newest version (1:1.2.8.dfsg-2ubuntu4).
The following packages were automatically installed and are no longer required:
docbook-xml docbook-xsl fonts-dejavu fonts-dejavu-extra icoutils kactivities
kate-data katepart kde-l10n-engb kde-runtime kde-runtime-data
kde-style-breeze kde-style-breeze-qt4 kdelibs-bin kdelibs5-data
kdelibs5-plugins kdoctools kpackagelauncherqml kpackagetool5 kwayland-data
kwayland-integration libattica0.4 libfam0 libgif7 libkactivities6
libkatepartinterfaces4 libkcmutils4 libkde3support4 libkdeclarative5
libkdesu5 libkdeui5 libkdewebkit5 libkdnssd4 libkemoticons4
libkf5activities5 libkf5archive5 libkf5attica5 libkf5auth-data libkf5auth5
libkf5calendarevents5 libkf5codecs-data libkf5codecs5 libkf5completion-data
libkf5completion5 libkf5config-bin libkf5config-data libkf5configcore5
libkf5configgui5 libkf5configwidgets-data libkf5configwidgets5
libkf5coreaddons-data libkf5coreaddons5 libkf5crash5 libkf5dbusaddons-bin
libkf5dbusaddons-data libkf5dbusaddons5 libkf5declarative-data
libkf5declarative5 libkf5globalaccel-bin libkf5globalaccel-data
```

2. After completion, download the latest BRO source code from the official website [www.bro.org](http://www.bro.org) and build it using the code below. (User have to change directories to the one containing the source code). Follow below steps

```
$git clone --recursive git://git.bro.org/bro
```

```
$/ configure
```

```
$make
```

```
$make install
```

```
Terminal File Edit View Search Terminal Help
jayk@ubuntu:~$ clear

jayk@ubuntu:~$ git clone --recursive git://git.bro.org/bro
Cloning into 'bro'...
remote: Counting objects: 87469, done.
remote: Compressing objects: 100% (26186/26186), done.
remote: Total 87469 (delta 63470), reused 79829 (delta 57117)
Receiving objects: 100% (87469/87469), 42.77 MiB | 1.17 MiB/s, done.
Resolving deltas: 100% (63470/63470), done.
Checking connectivity... done.
Submodule 'aux/binpac' (git://git.bro.org/binpac) registered for path 'aux/binpac'
Submodule 'aux/bro-aux' (git://git.bro.org/bro-aux) registered for path 'aux/bro-aux'
Submodule 'aux/broccoli' (git://git.bro.org/broccoli) registered for path 'aux/broccoli'
Submodule 'aux/broctl' (git://git.bro.org/broctl) registered for path 'aux/broctl'
Submodule 'aux/broker' (git://git.bro.org/broker) registered for path 'aux/broker'
Submodule 'aux/btest' (git://git.bro.org/btest) registered for path 'aux/btest'
Submodule 'aux/netcontrol-connectors' (git://git.bro.org/bro-netcontrol) registered for path 'aux/netcontrol-connectors'
Submodule 'aux/plugins' (git://git.bro.org/bro-plugins) registered for path 'aux/plugins'
```

3. Or, user can download pre-build binary package from the official website.  
(Based on pre-build package) Follow below steps to install Bro

```
$sudo sh -c "echo 'deb
```

```
http://download.opensuse.org/repositories/network:/bro/xUbuntu_16.04/' >>
```

```
/etc/apt/sources.list.d/bro.list"
```

```
$sudo apt-get update
```

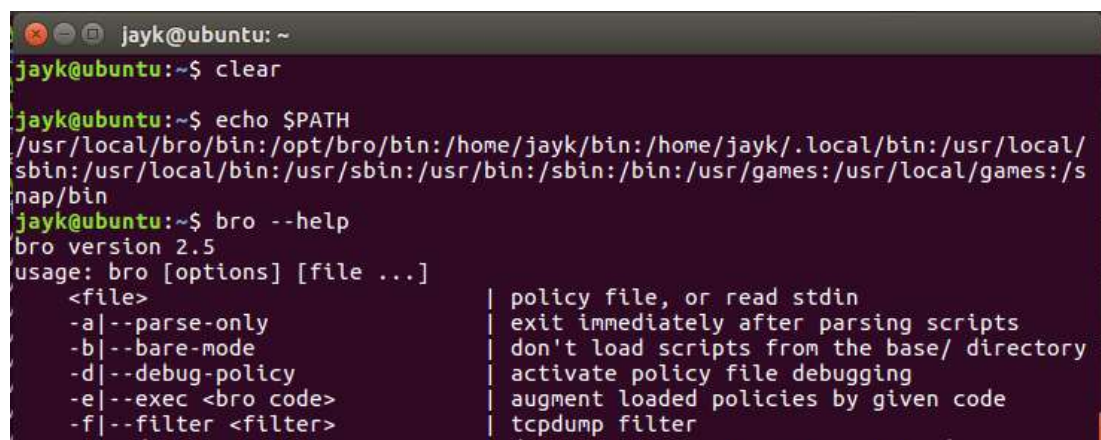
```
$sudo apt-get install bro
```

4. Update configuration and check the status of installation.

Update PATH for Bro.

Copy the line below at the end of .bashrc file which located in user home directory using text editor(vi or nano)

```
export PATH=opt/bro/bin:/usr/local/bro/bin:$PATH
```



```
jayk@ubuntu: ~  
jayk@ubuntu:~$ clear  
  
jayk@ubuntu:~$ echo $PATH  
/usr/local/bro/bin:/opt/bro/bin:/home/jayk/bin:/home/jayk/.local/bin:/usr/local/  
sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/s  
nap/bin  
jayk@ubuntu:~$ bro --help  
bro version 2.5  
usage: bro [options] [file ...]  
      <file>                                | policy file, or read stdin  
-a|--parse-only                             | exit immediately after parsing scripts  
-b|--bare-mode                             | don't load scripts from the base/ directory  
-d|--debug-policy                          | activate policy file debugging  
-e|--exec <bro code>                       | augment loaded policies by given code  
-f|--filter <filter>                       | tcpdump filter
```

5. Now, Bro is ready to run. Bro can be launched by command broctl. User can find required command with help option.

```
$broctl
```

```
[BroControl]help
```



```
jayk@ubuntu: ~  
jayk@ubuntu:~$ broctl  
Welcome to BroControl 1.5  
Type "help" for help.  
[BroControl] > help  
BroControl Version 1.5  
  
capstats [<nodes>] [<secs>] - Report interface statistics with capstats  
check [<nodes>] - Check configuration before installing it  
cleanup [--all] [<nodes>] - Delete working dirs (flush state) on nodes  
config - Print broctl configuration  
cron [--no-watch] - Perform jobs intended to run from cron  
cron enable|disable|? - Enable/disable "cron" jobs  
deploy - Check, install, and restart  
df [<nodes>] - Print nodes' current disk usage  
diag [<nodes>] - Output diagnostics for nodes  
exec <shell cmd> - Execute shell command on all hosts  
exit - Exit shell  
install - Update broctl installation/configuration  
netstats [<nodes>] - Print nodes' current packet counters  
nodes - Print node configuration  
peerstatus [<nodes>] - Print status of nodes' remote connections  
print <id> [<nodes>] - Print values of script variable at nodes  
process <trace> [<op>] [-- <sc>] - Run Bro (with options and scripts) on trace  
quit - Exit shell  
restart [--clean] [<nodes>] - Stop and then restart processing  
scripts [-c] [<nodes>] - List the Bro scripts the nodes will load  
start [<nodes>] - Start processing  
status [<nodes>] - Summarize node status  
stop [<nodes>] - Stop processing  
top [<nodes>] - Show Bro processes ala top  
update [<nodes>] - Update configuration of nodes on the fly  
  
Commands provided by plugins:  
  
ps.bro [<nodes>] - Show Bro processes on nodes' systems  
[BroControl] > █
```

```
jayk@ubuntu: /opt/bro/etc  
inet addr:127.0.0.1 Mask:255.0.0.0  
inet6 addr: ::1/128 Scope:Host  
UP LOOPBACK RUNNING MTU:65536 Metric:1  
RX packets:4207 errors:0 dropped:0 overruns:0 frame:0  
TX packets:4207 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1  
RX bytes:515964 (515.9 KB) TX bytes:515964 (515.9 KB)  
  
jayk@ubuntu:/opt/bro/etc$ vi node.cfg  
jayk@ubuntu:/opt/bro/etc$ broctl  
Warning: broctl node config has changed (run the broctl "deploy" command)  
  
Welcome to BroControl 1.5  
Type "help" for help.  
[BroControl] > start  
starting bro (was crashed) ...  
Error: error occurred while trying to send mail: send-mail: /usr/sbin/sendmail not found  
[BroControl] > status  
Name Type Host Status Pid Started  
bro standalone localhost running 17585 28 Mar 04:46:50  
[BroControl] > █
```

6. Log files of Bro - conn.log



Elevation of privileges is an unlikely scenario with BRO. This is due to the fact that an intruder will be detected by BRO as they try to break into the network. This will notify the security team and they will deploy countermeasure to try and block the malicious use. This will be essential since the user won't have the time to elevate their privileges

### **3.5 Bro requirements to people and processes**

Since BRO uses a policy language tailored to work with the application only, the system it is installed on should be able to handle the resource requirements of the application. This will include a large disk space to store the log files and a large RAM space to hold the runtime application data of the application.

The application is designed for use by experts in UNIX like operating systems. It is therefore a requirement that a user intending to use bro be conversant with the operating systems. A lack of understanding in the UNIX like operating systems will lead to poor operation of the program and users will not experience the full functionality of the application [1].

## **4. Host Intrusion Detection System (HIDS)**

HIDS are intrusion detection systems that run on a single host and detect security intrusions on these hosts. HIDS are concerned with running the application on only one computer system while giving a report on its status.

Most HIDS bundles now can effectively anticipate pernicious or irregular action on the host framework. Because of the potential effect, this can have on the end client, HIDS is as often as possible sent in "screen just" mode at first. This empowers the executive to make a standard of the framework arrangement and movement. Dynamic hindering of utilizations, framework changes, and system action is restricted to just the most terrible exercises. Directors can then tune the framework strategy in light of what is viewed as "ordinary action".

### **4.1 OSSEC**

OSSEC is a stage to screen and control your frameworks. It combines every one of the parts of HIDS (host-based intrusion identification), log checking, and Security Occurrence Administration (SIM)/Security Data and Occasion Administration (SIEM) together in a straightforward, capable, and open source arrangement [4].

### **4.2 Key Advantages**

|                            |   |
|----------------------------|---|
| Consistence<br>Necessities | OSSEC helps clients meet particular consistency necessities, for example, PCI and HIPAA. It gives clients a chance to identify and caution on unapproved document framework alterations and malevolent conduct inserted in the log records of |
|----------------------------|---|

|                                      |   |
|--------------------------------------|---|
|                                      | business items and additionally custom applications. For PCI, it covers the areas of document honesty observing (PCI 11.5, 10.5), log examination and checking (segment 10), and strategy implementation/checking   |
| Multi-stage                          | OSSEC gives clients a chance to actualize a complete host based interruption location framework with fine-grained application/server particular strategies over numerous stages, for example, Linux, Solaris, Windows, and Macintosh OS X   |
| Continuous and Configurable Cautions | OSSEC gives clients a chance to design episodes they need to be alarmed on and gives them a chance to concentrate on raising the need of basic occurrences over the consistent clamor on any framework. The combination with SMTP, SMS, and Syslog permits clients to be on top of cautions by sending them to email empowered gadgets. Dynamic reaction choices to hinder an assault instantly are additionally accessible |
| Coordination with current framework  | OSSEC will coordinate with current ventures from clients, for example, SIM/SEM (Security Episode Administration/Security Occasions Administration) items for concentrated revealing and relationship of occasions.  |
| Concentrated administration          | OSSEC gives an improved unified administration server to oversee approaches over various working frameworks. Furthermore, it likewise gives clients a chance to characterize server particular supersedes for better-grained strategies   |
| Operator and agentless observing     | OSSEC offers the adaptability of the operator based and agentless observing of frameworks and systems administration segments, for example, switches and firewalls. Agentless checking gives clients who a chance to have limitations on programming being introduced on frameworks, (for example, FDA endorsed frameworks or machines) meet security and consistency needs   |

### 4.3 Key Elements

|                                  |   |
|----------------------------------|---|
| Document Respectability checking | There is one thing in like manner to any assault to your systems and PCs: they change your frameworks somehow. The objective of document honesty checking (or FIM - record uprightness observing) is to distinguish these progressions and caution you when they happen. It can be an assault, or an abuse by a representative or even a grammatical error by an administrator, any record, catalog or registry change will be alarmed to you   |
| Log Observing                    | You're working framework needs to address you, however, do you know how to tune in? Each working framework, application, and gadget on your system create logs (occasions) to tell you what is going on. OSSEC gathers, investigates and associates these logs to fill you in as to whether something suspicious is going on (assault, abuse, blunders, and so forth). Would you like to know when an application is introduced on your customer box? Or, on the other hand when somebody changes a control in your firewall? By observing your logs, OSSEC will inform you |

|                  |  |
|------------------|--|
| Rootkit location | Criminal programmers need to conceal their activities, however utilizing rootkit identification you can be told when the framework is altered in a path normal to rootkits |
| Dynamic reaction | Dynamic reaction permits OSSEC to make a prompt move when determined alarms are activated. This may keep an episode from spreading before a director can make a move       |

#### 4.4 Installing OSSEC [4]

1. Before installing OSSEC, need to change to root user.  
\$sudo su
2. Installation package can be downloaded from [www.ossec.net](http://www.ossec.net). In order to download the latest OSSEC, type below command in terminal window.  
\$ wget -U ossec <http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz>  
\$ wget -U ossec http://www.ossec.net/files/ossec-hids-2.8.1-checksum.txt
3. Extract the contents of the archive into a folder and open a terminal then change directory into the one with the files.  
\$ tar -zxf ossec-hids-2.8.1.tar.gz  
\$ cd ossec-hids-2.8.1
4. In the directory, there is OSSEC installation script. To start install, type below command in terminal  
\$ ./install.sh

Then, user will be prompted to answer several installation questions.

```

root@ubuntu: /home/jayk/ossec-hids-2.8.1
OSSEC HIDS v2.8 Installation Script - http://www.ossec.net

You are about to start the installation process of the OSSEC HIDS.
You must have a C compiler pre-installed in your system.
If you have any questions or comments, please send an e-mail
to dcid@ossec.net (or daniel.cid@gmail.com).

- System: Linux ubuntu 4.8.0-36-generic
- User: root
- Host: ubuntu

-- Press ENTER to continue or Ctrl-C to abort. --

```



```
root@ubuntu: /home/jayk/ossec-hids-2.8.1
OSSEC HIDS v2.8 Installation Script - http://www.ossec.net

You are about to start the installation process of the OSSEC HIDS.
You must have a C compiler pre-installed in your system.
If you have any questions or comments, please send an e-mail
to dcid@ossec.net (or daniel.cid@gmail.com).

- System: Linux ubuntu 4.8.0-36-generic
- User: root
- Host: ubuntu

-- Press ENTER to continue or Ctrl-C to abort. --

1- What kind of installation do you want (server, agent, local, hybrid or help)?
local

- Local installation chosen.

2- Setting up the installation environment.

- Choose where to install the OSSEC HIDS [/var/ossec]:

- Installation will be made at /var/ossec .

3- Configuring the OSSEC HIDS.

3.1- Do you want e-mail notification? (y/n) [y]:
- What's your e-mail address? jaykkr@gmail.com

- We found your SMTP server as: alt3.gmail-smtp-in.l.google.com.
- Do you want to use it? (y/n) [y]: y

--- Using SMTP server: alt3.gmail-smtp-in.l.google.com.

3.2- Do you want to run the integrity check daemon? (y/n) [y]: y

- Running syscheck (integrity check daemon).

3.3- Do you want to run the rootkit detection engine? (y/n) [y]:
```

```
root@ubuntu: /home/jayk/ossec-hids-2.8.1
you can block an IP address or disable access for
a specific user.
More information at:
http://www.ossec.net/en/manual.html#active-response

- Do you want to enable active response? (y/n) [y]: y

- Active response enabled.

- By default, we can enable the host-deny and the
firewall-drop responses. The first one will add
a host to the /etc/hosts.deny and the second one
will block the host on iptables (if linux) or on
ipfilter (if Solaris, FreeBSD or NetBSD).
- They can be used to stop SSHD brute force scans,
portscans and some other forms of attacks. You can
also add them to block on snort events, for example.

- Do you want to enable the firewall-drop response? (y/n) [y]: y

- firewall-drop enabled (local) for levels >= 6

- Default white list for the active response:
- 127.0.1.1

- Do you want to add more IPs to the white list? (y/n)? [n]: n

3.6- Setting the configuration to analyze the following logs:
-- /var/log/auth.log
-- /var/log/syslog
-- /var/log/dpkg.log

- If you want to monitor any other file, just change
the ossec.conf and add a new localfile entry.
Any questions about the configuration can be answered
by visiting us online at http://www.ossec.net .

--- Press ENTER to continue ---
```

5. After finishing installation, check status and then start OSSEC with below command. OSSEC Command location is /var/ossec/bin/ which should be added to environment variables(PATH).  
\$ ossec-control status  
\$ ossec-control start

```
root@ubuntu: /home/jayk/ossec-hids-2.8.1
- To stop OSSEC HIDS:
    /var/ossec/bin/ossec-control stop

- The configuration can be viewed or modified at /var/ossec/etc/ossec.conf

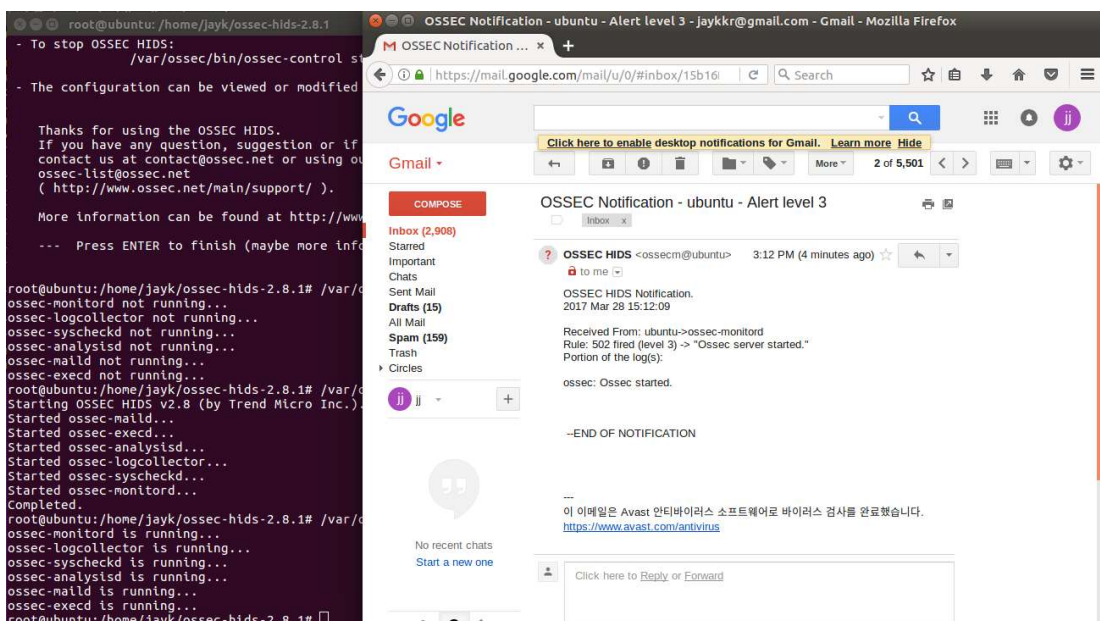
Thanks for using the OSSEC HIDS.
If you have any question, suggestion or if you find any bug,
contact us at contact@ossec.net or using our public maillist at
ossec-list@ossec.net
( http://www.ossec.net/main/support/ ).

More information can be found at http://www.ossec.net

--- Press ENTER to finish (maybe more information below). ---

root@ubuntu: /home/jayk/ossec-hids-2.8.1# /var/ossec/bin/ossec-control status
ossec-monitord not running...
ossec-logcollector not running...
ossec-syscheckd not running...
ossec-analysisd not running...
ossec-maild not running...
ossec-execd not running...
root@ubuntu: /home/jayk/ossec-hids-2.8.1# /var/ossec/bin/ossec-control start
Starting OSSEC HIDS v2.8 (by Trend Micro Inc.)...
Started ossec-maild...
Started ossec-execd...
Started ossec-analysisd...
Started ossec-logcollector...
Started ossec-syscheckd...
Started ossec-monitord...
Completed.
root@ubuntu: /home/jayk/ossec-hids-2.8.1# /var/ossec/bin/ossec-control status
ossec-monitord is running...
ossec-logcollector is running...
ossec-syscheckd is running...
ossec-analysisd is running...
ossec-maild is running...
ossec-execd is running...
root@ubuntu: /home/jayk/ossec-hids-2.8.1#
```

6. Right after starting OSSEC, user will receive an email about starting of OSSEC.

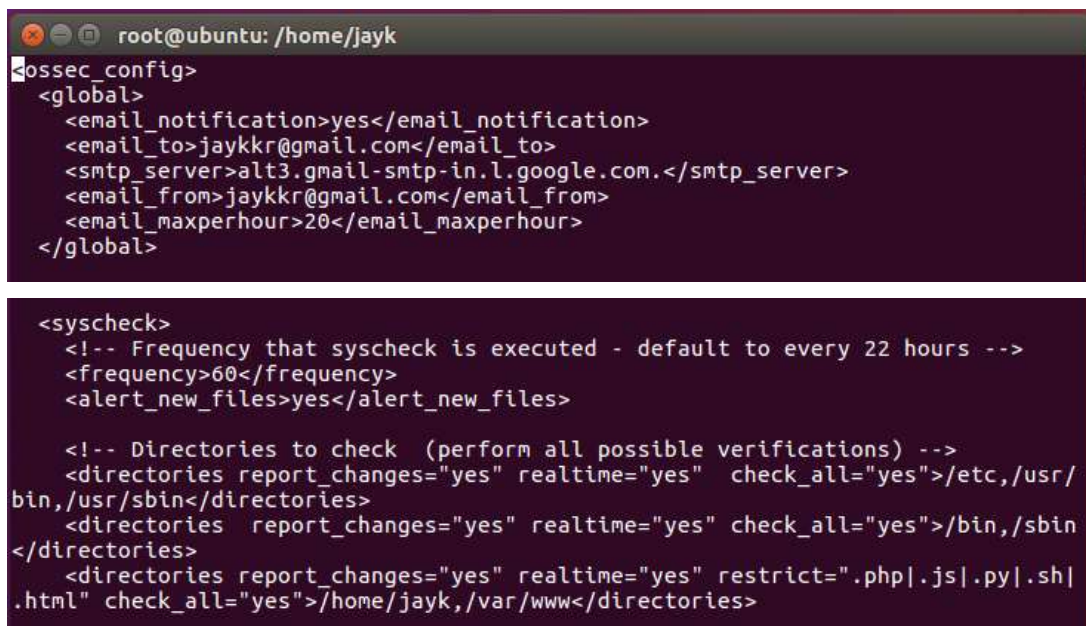




## 4.5 Configuration of OSSEC

Main configuration files and predefined rules are located in `/var/ossec/etc` and `/var/ossec/rules` directory respectively and logs of OSSEC is stored in `/var/ossec/logs` directory. With `ossec.conf` files in `/var/ossec/etc` directory, user could change email setting like an email address, a cadence of email notification etc., and also may control the frequency of system check. In case some occasion meet to the criteria which is defined by user with configuration files, OSSEC will send an email notification to user with specification of the case. To the next, even though there are many rule files in the `/var/ossec/rules` directory, user could utilize `local_rules.xml` to customize their system. For instance, with this file, user could have an email notification if file is add to specific directory.

1. In `ossec.conf` file, user may need to  
change email related settings like sender and the max number of email per hour etc.  
enable realtime monitoring  
specify directory which user may want to monitor

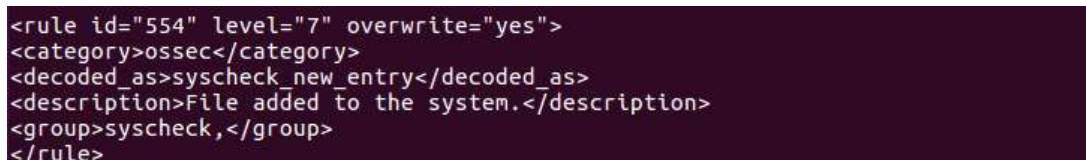
A terminal window with a dark background and light text. The prompt is 'root@ubuntu: /home/jayk'. The user has run 'ossec\_config' and the output shows XML configuration for email notifications and system checks. The email settings include 'email\_notification' set to 'yes', 'email\_to' as 'jaykkr@gmail.com', 'smtp\_server' as 'alt3.gmail-smtp-in.l.google.com.', 'email\_from' as 'jaykkr@gmail.com', and 'email\_maxperhour' as '20'. The syscheck settings include 'frequency' of '60', 'alert\_new\_files' set to 'yes', and a list of directories to monitor: '/etc,/usr/bin,/usr/sbin' and '/home/jayk,/var/www'. The second directory list has 'report\_changes' set to 'yes', 'realtime' set to 'yes', 'check\_all' set to 'yes', and a 'restrict' list containing '.php|.js|.py|.sh|.html'.

```
root@ubuntu: /home/jayk
ossec_config
<global>
  <email_notification>yes</email_notification>
  <email_to>jaykkr@gmail.com</email_to>
  <smtp_server>alt3.gmail-smtp-in.l.google.com.</smtp_server>
  <email_from>jaykkr@gmail.com</email_from>
  <email_maxperhour>20</email_maxperhour>
</global>

<syscheck>
  <!-- Frequency that syscheck is executed - default to every 22 hours -->
  <frequency>60</frequency>
  <alert_new_files>yes</alert_new_files>

  <!-- Directories to check (perform all possible verifications) -->
  <directories report_changes="yes" realtime="yes" check_all="yes">/etc,/usr/
bin,/usr/sbin</directories>
  <directories report_changes="yes" realtime="yes" check_all="yes">/bin,/sbin
</directories>
  <directories report_changes="yes" realtime="yes" restrict=".php|.js|.py|.sh|
.html" check_all="yes">/home/jayk,/var/www</directories>
```

2. Add new rule at the end of `local_rules.xml` file like below (rule 554 is about adding a file to directory). New rule will notify any changes on the specified directory to user.

A terminal window showing the XML configuration for a new rule. The rule has an id of '554', level of '7', and 'overwrite' set to 'yes'. The category is 'ossec'. The decoded message is 'syscheck\_new\_entry'. The description is 'File added to the system.'. The group is 'syscheck'.

```
<rule id="554" level="7" overwrite="yes">
  <category>ossec</category>
  <decoded_as>syscheck_new_entry</decoded_as>
  <description>File added to the system.</description>
  <group>syscheck,</group>
</rule>
```

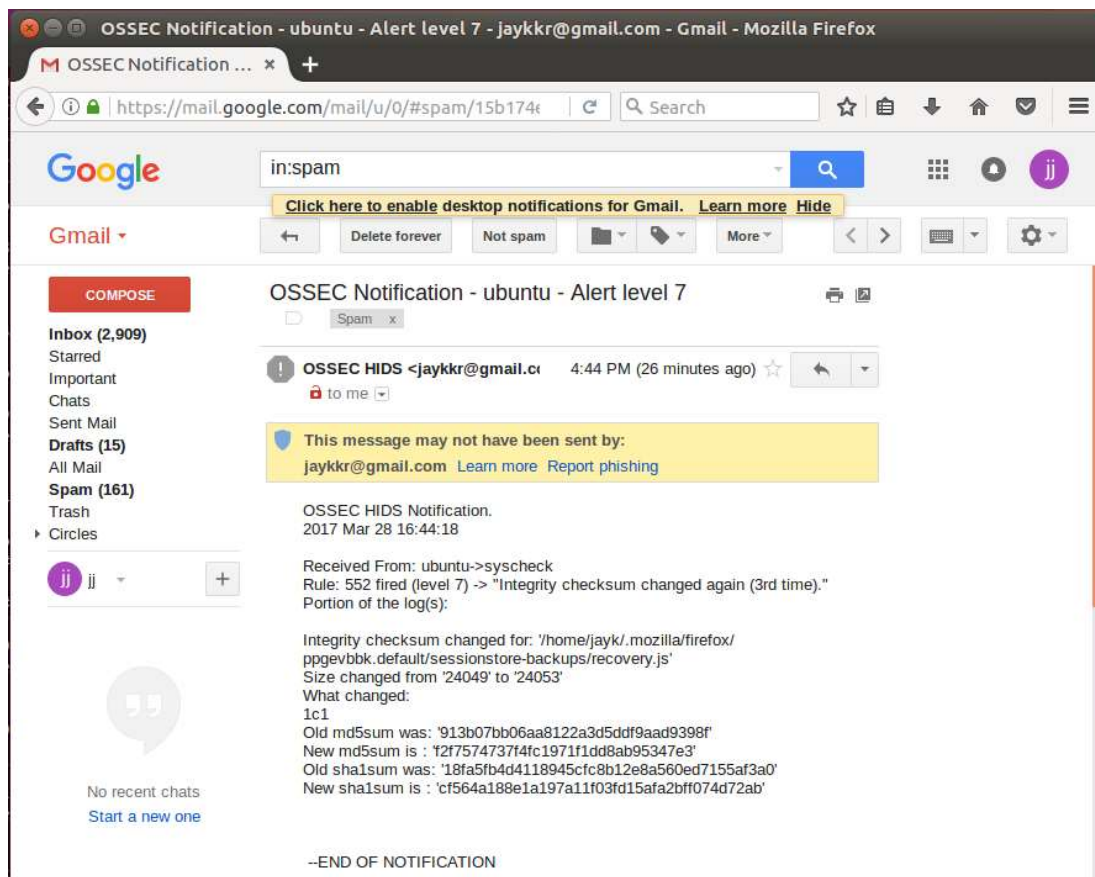
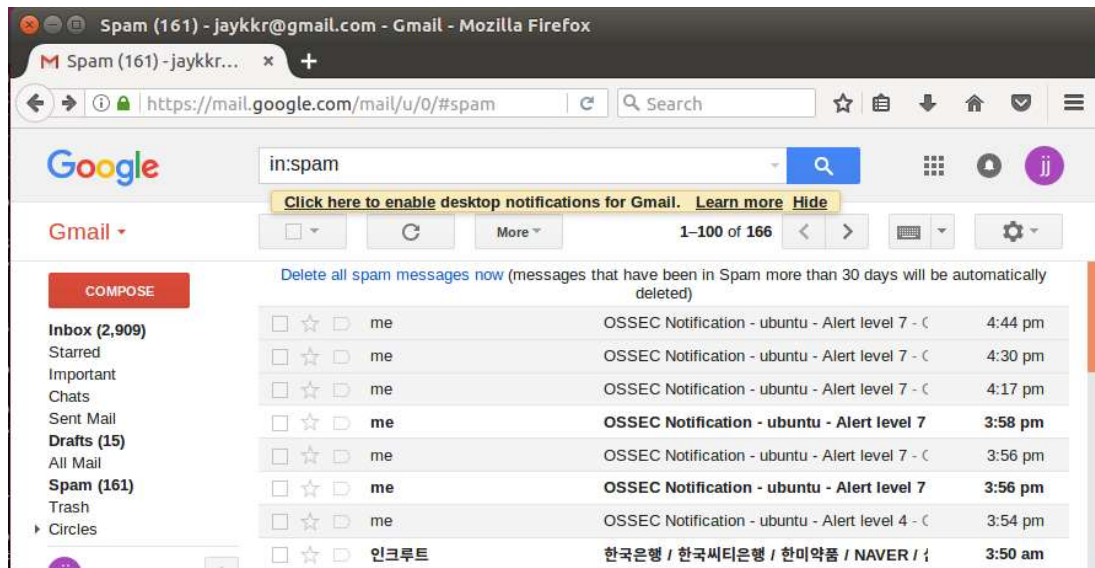
3. Restart OSSEC to apply new rules.  
\$ `/var/ossec/bin/ossec-control restart`
4. Conduct series of commands below. As you see in `ossec.conf` file, `/home/jayk/` directory was specified for realtime alerting.

\$ touch /home/jayk/index.html : renew time information of the file

\$ nano /home/jayk/index.html : editing(update file)

\$ rm /home/jayk/index.html : remove file from the directory

5. User should notified the activities based on the definition in local\_rules.xml



#### **4.6 Investigating logs of OSSEC**

OSSEC records all activities that user performed in /var/ossec/logs/ directory since starting the application. Later on, these logs can be used to analyze and spot any suspicious activity.

#### **4.7 Analysis of OSSEC using the STRIDE model**

The first threat category in the STRIDE standard is Spoofing. OSSEC is vulnerable to spoofing in that if a malicious user gains control of a computer host, the can change the login details for OSSEC and use a false identity to log into OSSEC's interface. However, this security threat will be logged into the application's log files (Microsoft,2005).

The second threat category is data tampering. In this threat category, a malicious user will intentionally change data values held in either a data file or in a log file. With strong text based editors in Linux like vim, a user can open and edit any log files available to cover their tracks.

Repudiation is not possible with OSSEC. This is due to the fact that every activity a user does is logged by OSSEC and the users can't deny it, especially after the log files are analyzed. Information disclosure is not a threat while using OSSEC since only authenticated users can only access the log files. Any data of sensitive nature is not easily accessible since the application has authentication and authorization structures. Information disclosure is a threat category under the STRIDE threat model. OSSEC will capture any attempts to stop any application used by users. If analyzed in time, the threat can be averted b taking necessary mitigation measures. The last category is privilege elevation. In this category, the user usually gains complete access to the system and changes their credentials to that of administrators who can perform any task in the system.

#### **4.8 OSSEC requirements to people and processes**

To prevent the intentional deletion of log files b a malicious user or a malware, the default installation directory of OSSEC should be caned to a different one. A daemon or a cron job should be set up to copy the log files to a remote host or to an email address. Some applications like ipfilters can be setup in routers or access points to block connections from blacklisted IP addresses.

### **5. Conclusion**

It is imperative for network or system administrators to understand the use of security assessment tools. Understanding the various security standards out there is essential for them to apply them in securing their systems. From this report, it can be deduced that application of only one tool is not enough to secure a system. A combination of different tools can be used to properly secure a system to standards acceptable by the various frameworks in place.

## 6. References

- [1] M. Handley, V. Paxson 그리고 C. Kreibich, "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics," ICSI, 2001. [온라인]. Available: <http://www.icir.org/vern/papers/norm-usenix-sec-01-html/index.html>.
- [2] B. Org, "Bro Documentation and Training," Bro Project, [온라인]. Available: <https://www.bro.org/documentation/index.html>.
- [3] T. S. P. R. M. S. B. N. B.Santos Kumar, "Intrusion Detection System- Types and Prevention," (IJCSIT) International Journal of Computer Science and Information Technologies, 2013. [온라인]. Available: <http://ijcsit.com/docs/Volume%204/Vol4Issue1/ijcsit2013040119.pdf>.
- [4] O. p. team, "Welcome to OSSEC's documentation," OSSEC org, [온라인]. Available: <http://ossec.github.io/docs/>.
- [5] etutorials.org, "Intrusion Detection Systems (IDS)," etutorials.org, [온라인]. Available: <http://etutorials.org/Networking/Cisco+Certified+Security+Professional+Certification/Part+V+Intrusion+Detection+Systems+IDS/>.
- [6] Microsoft, "The STRIDE Threat Model," Microsoft, 2002. [온라인]. Available: <https://msdn.microsoft.com/en-us/library/ee823878>.
- [7] H. Debar, "An Introduction to Intrusion-Detection Systems," IBM Research, [온라인]. Available: <http://sharif.edu/~kharrazi/courses/40817-941/reading/Debar00a.pdf>.
- [8] I. labs, "Intrusion Detection Systems," ICSA, [온라인]. Available: <https://www.ipa.go.jp/security/fy11/report/contents/intrusion/ids-meeting/idsbg.pdf>.
- [9] J. M. J. S. Mark Dowd, "The Art of Software Security Assessment - Identifying and Preventing Software Vulnerabilities," Addison Wesley Professional, 2007. [온라인]. Available: <https://leaksource.files.wordpress.com/2014/08/the-art-of-software-security-assessment.pdf>.
- [10] N. O. S. & S. A. Assistant Secretary of the Navy Chief System Engineer, "Software Security Assessment Tools Review," Booz Allen Hamilton, 2009. [온라인]. Available: <https://samate.nist.gov/docs/NAVSEA-Tools-Paper-2009-03-02.pdf>.
- [11] S. Campbell, "How To Install and Configure OSSEC Security Notifications on Ubuntu 14.04," digitalocean, 2015. [온라인]. Available: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ossec-security-notifications-on-ubuntu-14-04>.