Computer System Security
ADE - Monitoring Documentation

Monitoring tool used - ***OSSEC***

OSSEC (Open Source HIDS SECurity) is a free and fully open-source host-based intrusion detection prevention system. It performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response. It allows a user to add custom alert rules and write scripts to take action when alerts occur.

OSSEC can be configured on several operating systems like Linux, FreeBSD, OpenBSD, Windows, Solaris etc. It can be used to monitor one server or many servers in server/agent mode and provides a real-time view into what's happening on a server. OSSEC has a cross-platform architecture that can be enabled to monitor and manage multiple systems from one, centralized location.

OSSEC consists of a main application, an agent, and a web interface.

- Manager – required for a distributed network or stand-alone installations.
- Agent – small program installed on the systems to be monitored
- Agentless – mode used to monitor firewalls, routers or Unix systems.

I chose OSSEC as my monitoring tool as it is one of the best open-source IDS. Most of the default configurations are enough to handle and provide alerts for different kinds of activities occurring on the agents or hosts it is monitoring.

OSSEC Monitoring Setup:

I set up OSSEC on a system running ubuntu using the following steps:

1. Open a terminal in the ubuntu system.
2. Type *sudo su* and provide the root password to become root.
3. First update the system to the latest stable version:

   apt-get update

4. OSSEC requires gcc, libc, apache and PHP. So, if these are not available on the box, it can be installed in one go using the following command:

   apt-get install build-essential gcc make apache2 libapache2-mod-php7.0 php7.0 php7.0-cli php7.0-common apache2-utils unzip wget sendmail inotify-tools -y

5. Execute the following commands to enable and start Apache2:

   sudo systemctl enable apache2
   $ sudo systemctl start apache2
   $ sudo a2enmod rewrite
   $ sudo systemctl restart apache2

6. Now, OSSEC can be downloaded:

   wget https://github.com/ossec/ossec-hids/archive/2.9.0.tar.gz

7. Once downloaded, the file can be extracted using:

   tar -xvzf 2.9.0.tar.gz

8. Change the directory and then run the installation script to install OSSEC:

   cd ossec-hids
   ./install.sh

   The installation begins and the following configuration questions are asked, I have provided the answers based on configuration set-up I used:

   Note: If you would like to proceed with the default configuration, you can directly hit ENTER.

   i. Select your installation language:

      (en/br/cn/de/el/es/fr/hu/it/jp/nl/pl/ru/sr/tr) [en]: ***ENTER***

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

   iii. 2- Setting up the installation environment.
      Choose where to install the OSSEC HIDS [/var/ossec]: **ENTER**
      Installation will be made at /var/ossec

   iv. 3- Configuring the OSSEC HIDS.

      3.1- What's the IP Address or hostname of the OSSEC HIDS server?: **<server IP address>**
      Adding Server IP <server IP address

      3.2- Do you want e-mail notification? (y/n) [y]: **n**
      Email notification disabled.

      3.3- Do you want to run the integrity check daemon? (y/n) [y]: **y**
      Running syscheck (integrity check daemon).

      3.4- Do you want to run the rootkit detection engine? (y/n) [y]: **y**
      Running rootcheck (rootkit detection).

      3.5 - Do you want to enable active response? (y/n) [y]: **n**
      Active response disabled.

      3.6- Do you want to enable remote syslog (port 514 udp) (y/n) [y]: **y**

   v. The installation process will now continue and complete.

9. To install the OSSEC web interface, enter the following commands:

   cd /tmp/
   sudo git clone https://github.com/ossec/ossec-wui.git
   sudo mv /tmp/ossec-wui /var/www/html
   cd /var/www/html/ossec-wui
   sudo ./setup.sh

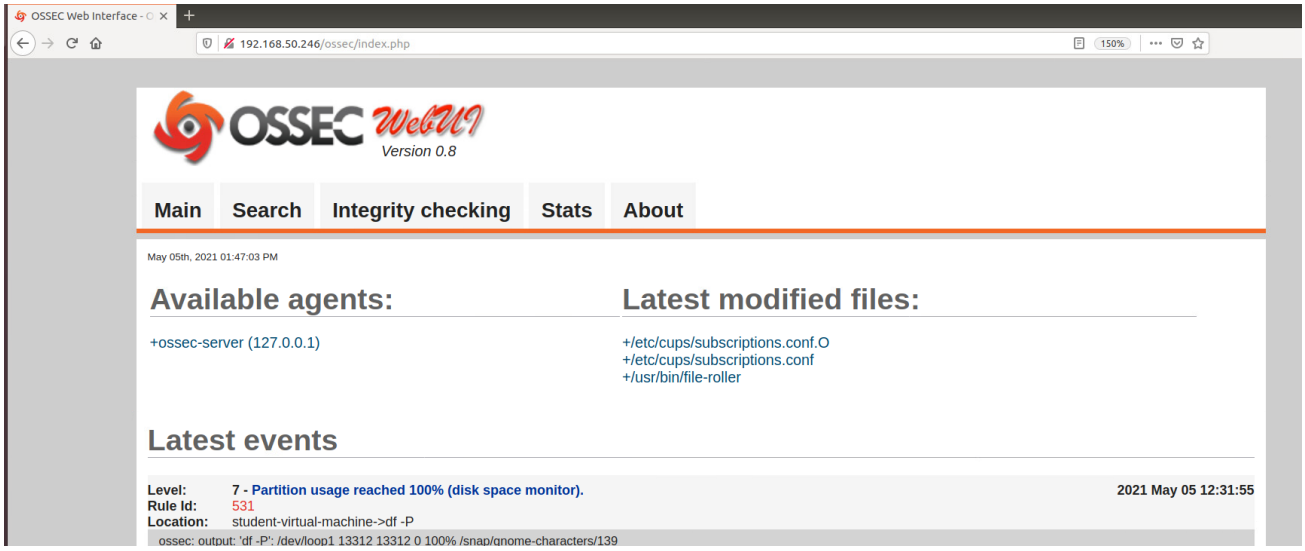
10. Enter your choice of username and password when prompted and set the web server username to *www-data.*
11. Set permissions using the following commands:

    sudo chown -R www-data:www-data /var/www/html/ossec-wui/
    sudo chmod -R 755 /var/www/html/ossec-wui/

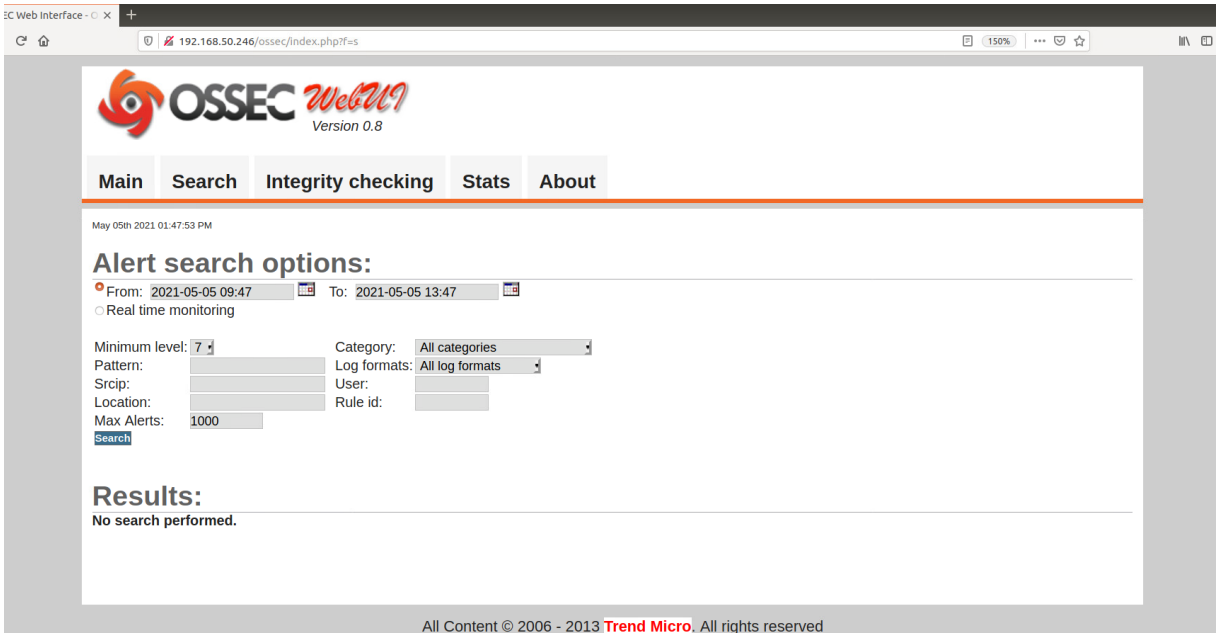
12. Restart Apache
    sudo systemctl restart apache2
13. Start OSSEC:
    sudo /var/ossec/bin/ossec-control start
14. Launch the web interface by navigating to http://<server-ip-address>/ossec-wui
15. OSSEC is now configured and ready to use.
16. Viewing Logs:
    a. The OSSEC logs can be viewed using the web interface.
    b. If you want to view older logs or the entire log file, execute the following commands in the terminal:

       sudo cd /var/ossec/logs/alerts
       leafpad alerts.log

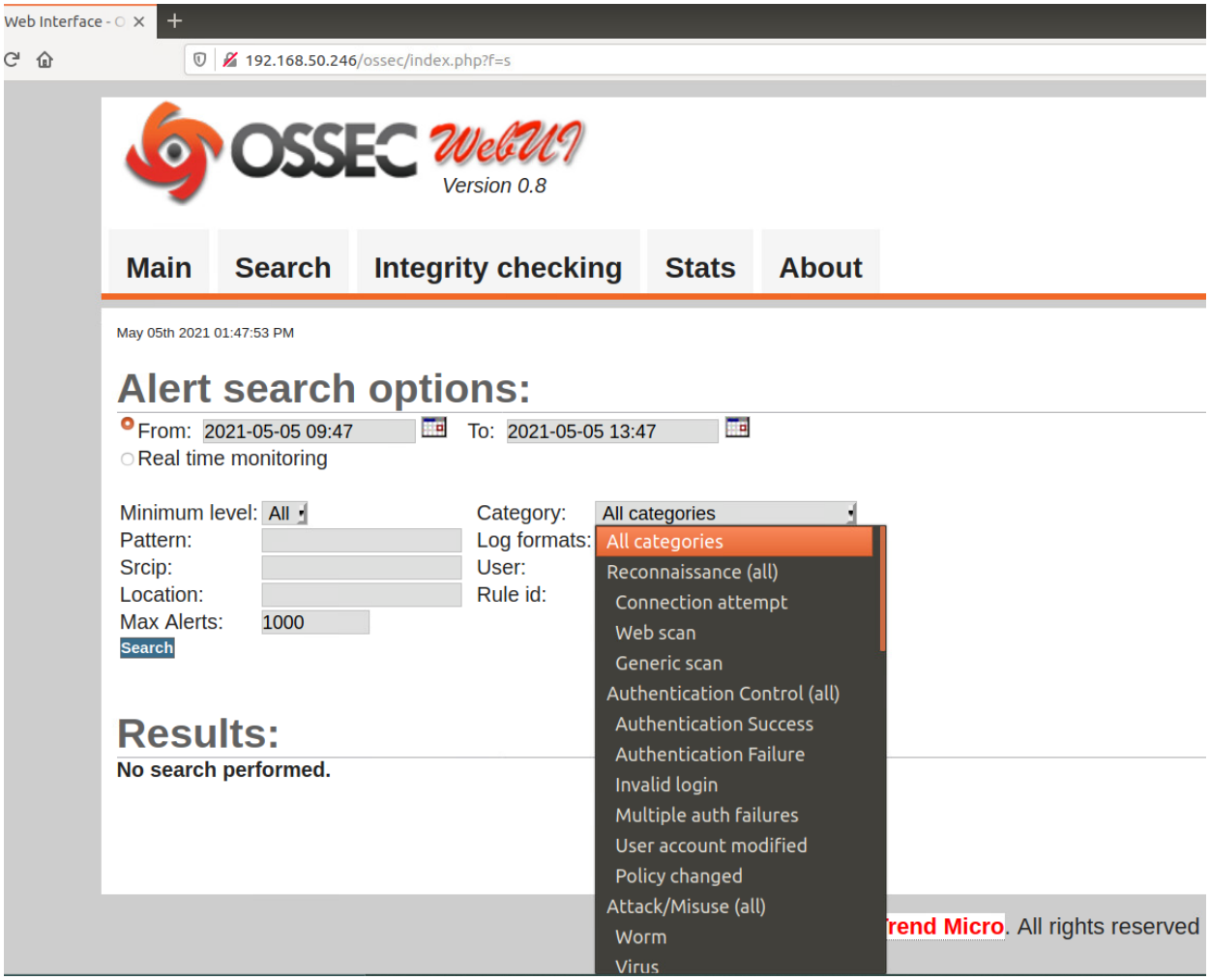
OSSEC Web UI:



You can search for different alerts and have various options to filter the alerts by:



The categories section shows different alert types that you can search by:



Interesting Activities and Attacks Observed using OSSEC:

The following were some of the interesting alerts that I observed as attempts were being made to exploit my vulnerable system. I have categorized them based on the type of alert:

1. *Reconnaissance:*

   a.

Level: 10 - Multiple web server 400 error codes from same source ip. **2021 Apr 27 22:35:27**
Rule Id: 31151
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.229

```
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user4 HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~web HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user5 HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~uucp HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user3 HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user1 HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~user2 HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~testuser HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~toor HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~test HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
192.168.50.229 - - [27/Apr/2021:22:35:26 -0400] "GET /cgi-bin/~system HTTP/1.1" 404 437 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
```

   The description shows that IP address 192.168.50.229 has been trying to scan the contents of the /cgi-bin directory. The above alert is generated when someone is trying to perform a web scan. When someone is trying to get illegal access to a system, they will probably scan them looking for vulnerable applications. That will cause the web server to generate many 400 error messages.
   Since one of the first attempts to identify if a system is vulnerable to shellshock is to identify if the apache is running a cgi-bin, this alert appears while trying to search the directory.

2. *Attack/Misuse:*

   a.

Level: 6 - Common web attack. **2021 Apr 29 21:32:47**
Rule Id: 31104
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:32:45 -0400] "GET /cgi-bin/scripts/root.exe?/c+dir HTTP/1.1" 404 492 "-" "-"
```

   The “common web attack” alert implies that an attempt to perform an attack through apache was attempted but was unsuccessful as indicated by the 404 code.

b.

Level: 6 - **XSS (Cross Site Scripting) attempt.** **2021 Apr 29 21:30:46**
Rule Id: 31105
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:30:44 -0400] "GET /scripts/message/message_dialog.tml?how_many_back=\\\"><script>alert(1)</script> HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:006195)"
```

The above log indicates that a Cross Site Scripting attempt was made by 192.168.50.200, however the *404* indicates that the attempt was unsuccessful.

c.

Level: 6 - **Suspicious URL access.** **2021 Apr 29 21:32:49**
Rule Id: 31516
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/wp-config.php.swp HTTP/1.1" 404 492 "-" "-"
```

The above “suspicious URL access” alert was generated when the attacker ran a command to search the cgi-bin to access a file. But it was unsuccessful as indicated by the 404 error.

d.

Level: 10 - **Multiple common web attacks from same source ip.** **2021 Apr 29 21:30:52**
Rule Id: 31153
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /shop/magmi/web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magento/magmi-importer/web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magento/magmi/web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magmi-importer/web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magmi/web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /web/download_file.php?file=../../app/etc/local.xml HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007047)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /shop/magmi-importer/web/download_file.php?file=../../../../../../../../../../../../etc/passwd HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007046)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /shop/magmi/web/download_file.php?file=../../../../../../../../../../../../etc/passwd HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007046)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magento/magmi-importer/web/download_file.php?file=../../../../../../../../../../../../etc/passwd HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007046)"
192.168.50.200 - - [29/Apr/2021:21:30:51 -0400] "GET /magento/magmi/web/download_file.php?file=../../../../../../../../../../../../etc/passwd HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:007046)"
```

The “multiple common web attacks from the same source” indicates that someone using the IP address 192.168.50.200 made several common web attack attempts but they were unsuccessful as indicated by the *404* code.

Level: 10 - **Multiple XSS (Cross Site Scripting) attempts from same source ip.** **2021 Apr 29 21:30:41**
Rule Id: 31154
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /cgi-bin/cvsblame.cgi?file=<script>alert('Vulnerable')</script> HTTP/1.1" 404 491 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003285)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /webtools/bonsai/cvsblame.cgi?file=<script>alert('Vulnerable')</script> HTTP/1.1" 404 491 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003284)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /cgi-bin/cvslog.cgi?file=<script>alert('Vulnerable')</script> HTTP/1.1" 404 491 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003283)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /cgi-bin/cvslog.cgi?file=*&rev=&root=<script>alert('Vulnerable')</script> HTTP/1.1" 404 491 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003282)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /webtools/bonsai/cvslog.cgi?file=<script>alert('Vulnerable')</script> HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003281)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /webtools/bonsai/cvslog.cgi?file=*&rev=&root=<script>alert('Vulnerable')</script> HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003280)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /cgi-bin/cvsquery.cgi?module=<script>alert('Vulnerable')</script>&branch=&dir=&file=&who=<script>alert(document.domain)</script>&
sortby=Date&hours=2&date=week HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003279)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /cgi-bin/cvsquery.cgi?branch=<script>alert('Vulnerable')</script>&file=<script>alert(document.domain)</script>&
date=<script>alert(document.domain)</script> HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003278)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /webtools/bonsai/cvsquery.cgi?module=<script>alert('Vulnerable')</script>&branch=&dir=&file=&who=<script>alert(document.domain)</script>&
sortby=Date&hours=2&date=week HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003277)"
192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /webtools/bonsai/cvsquery.cgi?branch=<script>alert('Vulnerable')</script>&file=<script>alert(document.domain)</script>&
date=<script>alert(document.domain)</script> HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003276)"
```

Just like the previous web attack alert, this “multiple XSS (Cross Site Scripting) attempts from the same source ip” indicates that someone using the IP address 192.168.50.200 made several Cross Site Scripting attack attempts but they were unsuccessful as indicated by the *404* code.

Namita Madhira

e.

Level: 6 - A web attack returned code 200 (success). **2021 Apr 29 21:30:46**
Rule Id: 31106
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

192.168.50.200 - - [29/Apr/2021:21:30:45 -0400] "GET /?xmlcontrol=body%20onload=alert(123) HTTP/1.1" 200 11228 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:006785)"

The "**web attack return code 200 (success)**" indicates that a web attack was successful. The location of this was at /var/log/apache2/access.log. As the vulnerability uses apache cgi-bin as the attack vector, this alert indicates that there was a successful attack by 192.168.50.200.

f.

Level: 6 - PHPMyAdmin scans (looking for setup.php). **2021 Apr 29 21:32:49**
Rule Id: 31515
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/xampp/phpmyadmin/scripts/setup.php HTTP/1.1" 404 492 "-" "-"

The "PHPMyAdmin scans (looking for setup.php)" alert indicates that indicates that a web scan was performed on the cgi-bin to view/access a file, but it was unsuccessful as indicated by the *404* code.

g.

Level: 6 - SQL injection attempt. **2021 Apr 29 21:30:41**
Rule Id: 31103
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

192.168.50.200 - - [29/Apr/2021:21:30:40 -0400] "GET /pls/portal/owa_util.cellsprint?p_theQuery=select+*+from+sys.dba_users HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:003456)"

The above log shows that an SQL injection attempt was made by 192.168.50.200 to view the contents of sys.dba_users, but it was unsuccessful as indicated by the *404* code.

h.

Level: 6 - PHP CGI-bin vulnerability attempt. **2021 Apr 29 21:30:46**
Rule Id: 31110
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

192.168.50.200 - - [29/Apr/2021:21:30:45 -0400] "GET /login.php?-s HTTP/1.1" 404 492 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:006524)"

The "PHP CGI-bin vulnerability attempt" alert log shows that an attempt to exploit a vulnerability in PHP CGI-bin was made but it was not successful as indicated by the 404 code.

3. *Access Control:*

a.

Level: 5 - Web server 400 error code. **2021 Apr 29 21:32:49**
Rule Id: 31101
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/wp-includes/rss-functions.php HTTP/1.1" 404 492 "-" "-"

Level: 10 - **Multiple web server 400 error codes from same source ip.** 2021 Apr 29 21:32:49
Rule Id: 31151
Location: student-virtual-machine->/var/log/apache2/access.log
Src IP: 192.168.50.200

```
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~user HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~user3 HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~toor HTTP/1.1" 404 493 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~user1 HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~user2 HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~sync HTTP/1.1" 404 491 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~test HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~system HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~staff HTTP/1.1" 404 456 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~testuser HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~sql HTTP/1.1" 404 492 "-" "-"
192.168.50.200 - - [29/Apr/2021:21:32:48 -0400] "GET /cgi-bin/~shutdown HTTP/1.1" 404 492 "-" "-"
```

The above two alerts “Web server 400 code” and “Multiple web server 400 error codes from the same source ip” indicate access attempts using cgi-bin. These alerts occur when a brute force tool (like dirsearch) has been used to search the cgi-bin directory. The files, however, cannot be accessed and hence a 404-error code has been generated.

4. *System Monitor:*

a.

Level: 5 - **Process segfaulted.** 2021 Apr 29 21:40:01
Rule Id: 1010
Location: student-virtual-machine->/var/log/syslog

```
Apr 29 21:40:00 student-virtual-machine kernel: [1375108.770503] shell.sh[30540]: segfault at 0 ip 000055b99f5f94a3 sp 00007ffdd67af8b0 error 4 in bash[55b99f5a6000+100000]
```

This “Process segfaulted” alert occurred after the exploit was successful and the attacker was trying to access a memory location that they were not allowed to access or trying to perform a write/overwrite to a read-only location. This is inferred as the description displays “shell.sh”, which is the vulnerable bash script that is used to exploit the system. The shellshock vulnerability using apache cgi-bin as the attack vector only allows you to write files in the www-data directory of apache. An attacker cannot write a file outside that directory.

b.

Level: 5 - **Invalid URI (bad client request).** 2021 Apr 27 21:51:44
Rule Id: 30315
Location: student-virtual-machine->/var/log/apache2/error.log
Src IP: 192.168.50.229

```
Src Port: 48664
[Tue Apr 27 21:51:43.540249 2021] [core:error] [pid 22474] [client 192.168.50.229:48664] AH00126: Invalid URI in request GET /help/../../../../../../../../../../../../../../../../../../../etc/shadow HTTP/1.1
```

Level: 10 - **Multiple Invalid URI requests from same source.** 2021 Apr 29 01:25:52
Rule Id: 30316
Location: student-virtual-machine->/var/log/apache2/error.log
Src IP: 192.168.50.95

```
Src Port: 53384
[Thu Apr 29 01:25:51.673013 2021] [core:error] [pid 19490] [client 192.168.50.95:53384] AH00126: Invalid URI in request GET /cgi-bin/PRN/../../../../../../../../../WINNT/system32/ipconfig.exe HTTP/1.1
[Thu Apr 29 01:25:51.671185 2021] [core:error] [pid 19495] [client 192.168.50.95:53382] AH00126: Invalid URI in request GET /cgi-bin/NUL/../../../../../../../../../WINNT/system32/ipconfig.exe HTTP/1.1
[Thu Apr 29 01:25:51.669341 2021] [core:error] [pid 19492] [client 192.168.50.95:53380] AH00126: Invalid URI in request GET /cgi-bin/../../../../../../../../../../WINNT/system32/ipconfig.exe HTTP/1.1
[Thu Apr 29 01:25:50.912377 2021] [core:error] [pid 20058] [client 192.168.50.95:53366] AH00126: Invalid URI in request GET ////////../../../../../../etc/passwd HTTP/1.1
[Thu Apr 29 01:25:50.910785 2021] [core:error] [pid 19491] [client 192.168.50.95:53364] AH00126: Invalid URI in request GET ////../../data/config/microsrv.cfg HTTP/1.1
[Thu Apr 29 01:25:49.324601 2021] [core:error] [pid 20058] [client 192.168.50.95:53234] AH00126: Invalid URI in request GET /../config.dat HTTP/1.1
[Thu Apr 29 01:25:49.304017 2021] [core:error] [pid 19491] [client 192.168.50.95:53232] AH00126: Invalid URI in request GET ../../../../../../../../../etc/passw* HTTP/1.1
[Thu Apr 29 01:25:49.302300 2021] [core:error] [pid 19493] [client 192.168.50.95:53230] AH00126: Invalid URI in request GET ../../../../../../../../../etc/* HTTP/1.1
[Thu Apr 29 01:25:49.219112 2021] [core:error] [pid 19490] [client 192.168.50.95:53228] AH00126: Invalid URI in request GET /../webserver.ini HTTP/1.1
[Thu Apr 29 01:25:48.921744 2021] [core:error] [pid 20058] [client 192.168.50.95:53222] AH00126: Invalid URI in request GET /file/../../../../../../../etc/ HTTP/1.1
```

The “Invalid URI (bad client request)” and “Multiple Invalid URI requests from the same source” alert with the “Invalid URI in request GET” indicate that the attacker (192.168.50.95) is trying to view files or directory through apache or www-data that it does not have read access to. Hence, the URI request cannot be resolved.

References:

1. https://www.ossec.net/
2. https://en.wikipedia.org/wiki/OSSEC
3. https://www.rapid7.com/blog/post/2017/06/30/how-to-install-and-configure-ossec-on-ubuntu-linux/
4. https://www.ossec.net/docs/