

Installation and use of SIEM, HIDS & NIDS

Configuration & Environment Information:

For this demonstration I am using 2 Physical devices. On my windows 10 there are 2 VM on the virtual box (kali & ubuntu). The 2 VM's are on the Bridge adapter and the total network sharing IP's from the same block 192.169.1.0/24. Config of the devices & use cases are given below.

Ubuntu(Separate PC):

Name: vboxuser@ubuntu

IP: 192.168.1.11

Purpose: install a SIEM (Elasticsearch, Kibana and Filebeat) , HIDS (Wazuh Manager) and NIDS (Suricata)

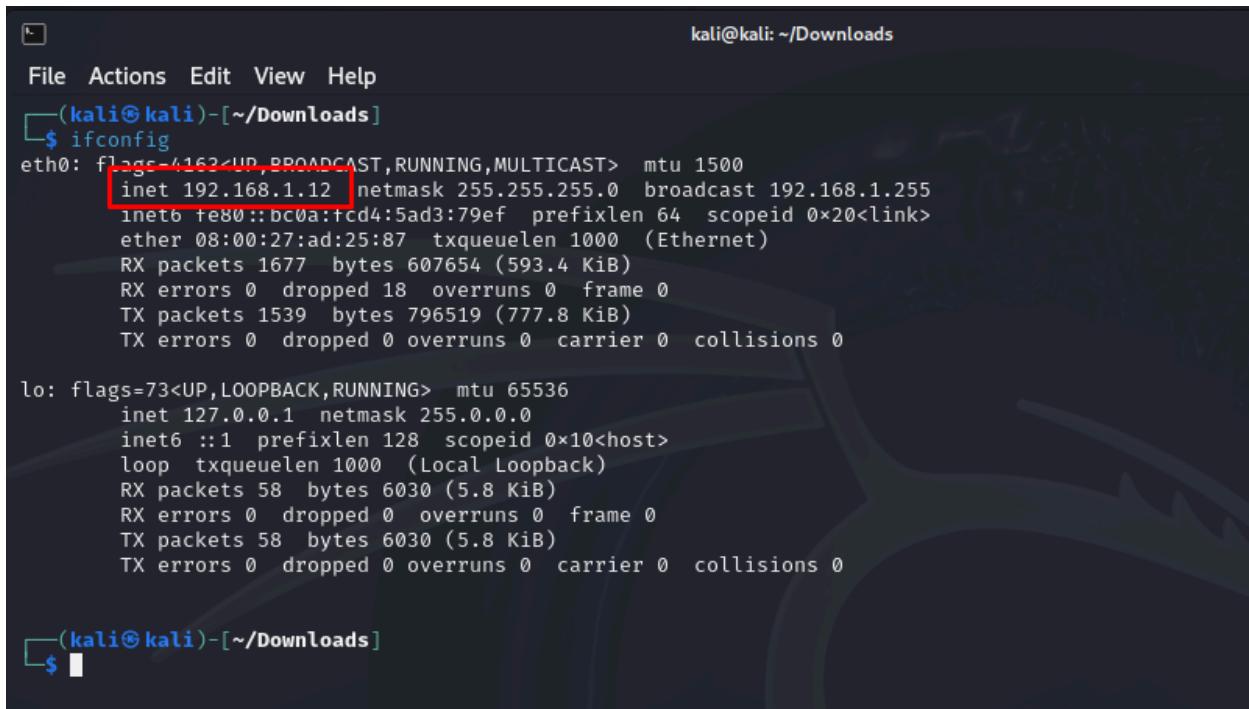
```
root@ubuntu:/home/vboxuser/soc_setup# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
→   inet 192.168.1.11 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::2b15:8870:8c76:2f9f prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:f9:1b:46 txqueuelen 1000 (Ethernet)
          RX packets 677144 bytes 539624488 (539.6 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 44382 bytes 3827781 (3.8 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

VM1 (Kali VM running on Virtual Box):

Name: Kali

IP: 192.168.1.12

Purpose: Attacking VM2



```
kali@kali: ~/Downloads
File Actions Edit View Help
[(kali㉿kali)-~/Downloads]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::bc0a:fcd4:5ad3:79ef prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
            RX packets 1677 bytes 607654 (593.4 KiB)
            RX errors 0 dropped 18 overruns 0 frame 0
            TX packets 1539 bytes 796519 (777.8 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 58 bytes 6030 (5.8 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 58 bytes 6030 (5.8 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[(kali㉿kali)-~/Downloads]
$ 
```

Windows 10 (PC):

Name: SIT

IP: 192.168.1.10

Purpose: Wazuh Agent (id: 001)

```
C:\Users\SIT>ipconfig  
Windows IP Configuration  
  
Ethernet adapter Ethernet:  
  
Connection-specific DNS Suffix . :  
Link local IPv6 Address . . . . . : fe80::b62e:c02a:9b67:48b9%3  
IPv4 Address. . . . . : 192.168.1.10  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : fe80::1%3  
192.168.1.1
```

Let's proceed with the installation

To set up a SIEM solution for comprehensive log analysis, the implementation involves the utilization of specific tools like, Elasticsearch, Kibana, and Filebeat. Furthermore, The SIEM setup includes Elasticsearch, Kibana and Filebeat version 7.17.13 as it is the compatible version to integrate with our HIDS Wazuh manager version 4.7.4 I am installing it in /otp The procedural steps for this deployment are encapsulated within a script accessible at the following repository:

Install Git & Curl first:

```
sudo apt install git  
sudo apt install curl
```

Navigate to otp:

```
cd /opt
```

Script execution:

1. Clone this repository to your local machine:

```
git clone https://github.com/samiul008ghub/source_setup/
```

2. Navigate to the repository's directory:

```
cd source_setup
```

3. Make the setup_script.sh executable:

```
chmod +x setup_script.sh
```

4. Execute the setup_script.sh”

./setup_script.sh

5. Follow the on-screen prompts to choose which components you want to install and continue with the setup. Post-Installation Steps

Related Screenshots for installation:

```
root@ubuntu: /home/vboxuser/soc_s... ×           vboxuser@ubuntu: ~ ×
vboxuser@ubuntu:~$ pwd
/home/vboxuser
vboxuser@ubuntu:~$ sudo su
[sudo] password for vboxuser:
root@ubuntu:/home/vboxuser# ls
Desktop  Downloads  Pictures  soc_setup  Videos
Documents  Music    Public    Templates
root@ubuntu:/home/vboxuser# cd soc_setup/
root@ubuntu:/home/vboxuser/soc_setup# chmod +x setup_script.sh
root@ubuntu:/home/vboxuser/soc_setup# ./setup_script.sh
All prerequisites are installed.



soc setup done here


This script will help you set up a security monitoring environment.



```
↳ [2025-01-07 17:37:41] Filebeat configured successfully.
↳ [2025-01-07 17:37:41] SIEM setup completed successfully!
```



Press Enter to continue...
Do you want to install Suricata (NIDS)? (y/n): y



↳ [2025-01-07 17:37:41] Filebeat configured successfully.
↳ [2025-01-07 17:37:41] SIEM setup completed successfully!



Press Enter to continue...



This script will help you set up a security monitoring environment.



```
↳ [2025-01-07 17:37:41] Filebeat configured successfully.
↳ [2025-01-07 17:37:41] SIEM setup completed successfully!
```


```

Filebeat is now configured and running.

The diagram consists of two main horizontal rows of nested brackets and boxes. The top row features large curly braces on the left and right sides, enclosing several pairs of square brackets. These square brackets are further nested within smaller curly braces. The bottom row follows a similar pattern but uses different bracket types: it starts with a large pair of square brackets, which are then enclosed by large curly braces, and so on, creating a layered, nested effect.

Press Enter to continue...
Do you want to install Wazuh (HIDS)? (y/n): y

A complex fractal pattern composed of nested brackets and parentheses, forming a self-similar structure. The pattern is highly detailed, with many levels of nesting that create a intricate, organic shape.

Setup Completion

SIEM

Elasticsearch installation proof

Provided screenshot from browser with ip address and port number.

The screenshot shows a browser window with the URL <https://192.168.1.8:9200/>. The page displays the Kibana configuration file in JSON format. Below the JSON, a terminal window shows the status of the Kibana service and network interface statistics.

```

{
  "name": "elasticsearch",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "wpeUJNTPoOx-MpRjub7A",
  "version": {
    "number": "7.17.13",
    "build_flavor": "default",
    "build_type": "tar",
    "build_hash": "282110800fbecaff7750443550a7f5d00f9c13",
    "build_date": "2023-08-31T17:30:19.958997072Z",
    "build_snapshot": false,
    "license_version": "8.15.1",
    "maximum_wire_compatibility_version": "8.8.0",
    "maximum_index_compatibility_version": "8.8.0-beta1"
  },
  "tagline": "You Know, For Search"
}

vboxuser@ubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.8 brd 192.168.1.255 bcast 192.168.1.255
  netmask 255.255.255.0 broadcast 192.168.1.255
  broadcast 192.168.1.255
  ether 00:00:27:f9:1b:46 txqueuelen 1000  (Ethernet)
    RX packets 5351874 bytes 7841565685 (7.0 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2418708 bytes 192938895 (192.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING mtu 65536
  inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
  loop  txqueuelen 1000  (Local Loopback)
    RX packets 1197842 bytes 495893798 (495.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1197842 bytes 495893798 (495.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vboxuser@ubuntu:~$
```

Kibana Installation Proof

Provided screenshot from browser with ip address and port number.

The screenshot shows a browser window with the URL <https://192.168.1.8:5601/app/home#/>. The page displays the Kibana home screen with four main cards: Welcome home, Security, and Analytics. Below the cards, a section titled "Get started by adding integrations" provides instructions for starting work with data. A terminal window at the bottom shows the status of the Kibana service and network interface statistics.

Welcome home

Security

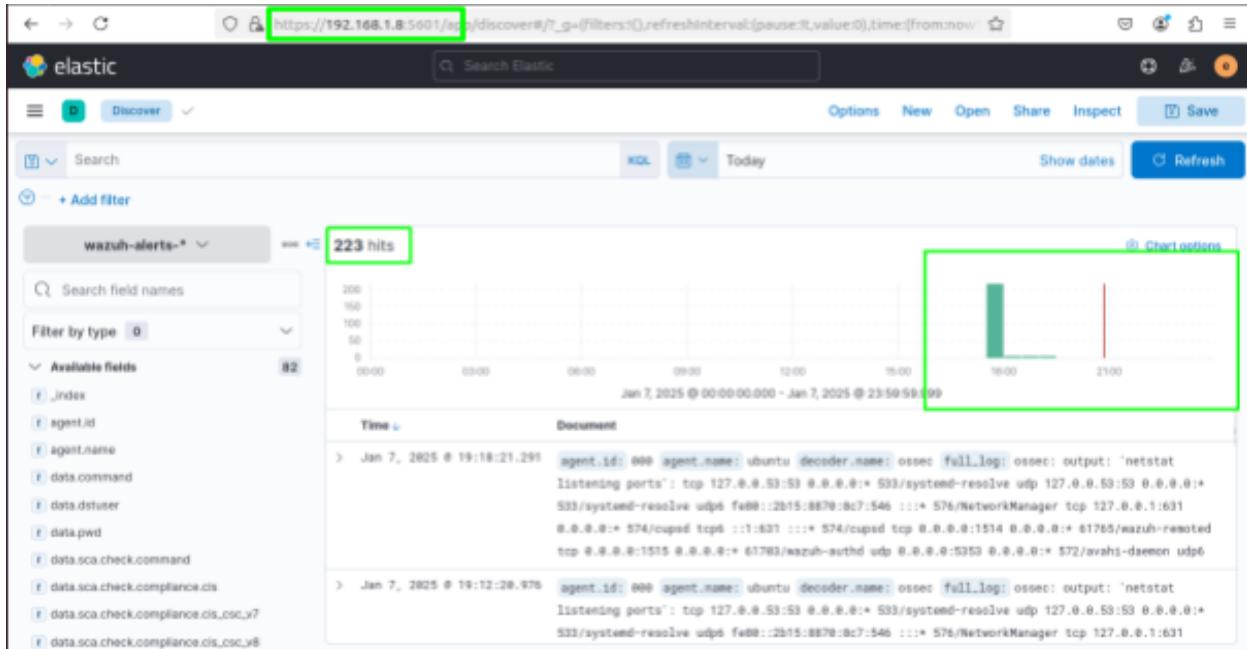
Analytics

Get started by adding integrations

vboxuser@ubuntu:~\$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.8 brd 192.168.1.255 bcast 192.168.1.255
 netmask 255.255.255.0 broadcast 192.168.1.255
 broadcast 192.168.1.255
 ether 00:00:27:f9:1b:46 txqueuelen 1000 (Ethernet)
 RX packets 5351874 bytes 7841565685 (7.0 GB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 2418708 bytes 192938895 (192.9 MB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Filebeat Installation Proof

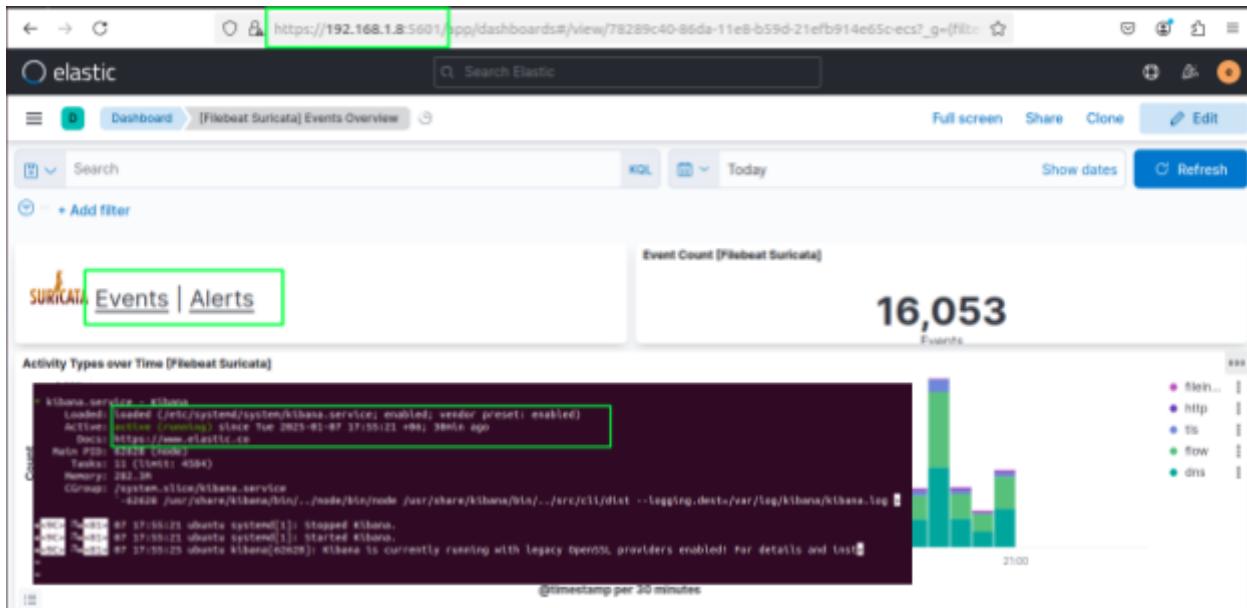
Provide a screenshot from the kibana dashboard which should have logs displayed.



NIDS

Suricata Installation Proof

Provide screenshot of Events/Alerts from kibana dashboard of suricata.



HIDS

Wazuh Manager Installation Proof

Provide a screen shot of the wazuh dashboard from the kibana console. It should have all components listed as zero as no agents were not integrated

The screenshot shows the Wazuh Manager interface at the URL <https://192.168.1.6:5443/wazuh#/management?tab=status>. The top navigation bar includes the Elastic logo and the Wazuh status indicator. The main content area is titled "Status" and displays a grid of service status icons. All services are shown as green (active), including wazuh-agentlessd, wazuh-moritord, wazuh-remoted, wazuh-db, wazuh-analysisd, wazuh-execd, wazuh-reportd, wazuh-apid, wazuh-authd, wazuh-integrationd, wazuh-syscheckd, wazuh-cayslogd, wazuh-logcollector, wazuh-clusterd, wazuh-dbd, wazuh-maild, and wazuh-modulesd. Below the status grid is a terminal window showing Kibana service logs. The logs indicate the service was started and is currently running. On the left sidebar, there is a "Manager" section with fields for Version (1.0.0), Compilation date (2023-05-17T17:55:21+00:00), Installation path (/opt/wazuh/), Installation type (server), and Agents (0).

DLL Hijacking

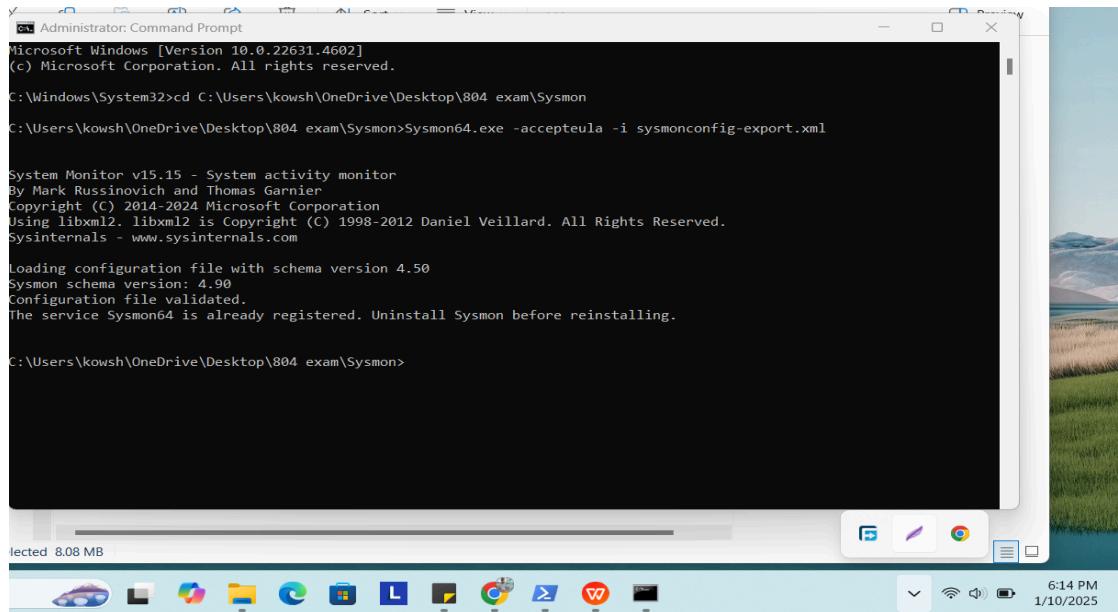
We done this on Machine [Windows-10] [10.42.0.238]

In essence, DLL injection is a common technique employed in malware and fileless adversary tradecraft to circumvent defenses. It entails the insertion of malicious code into running processes by coercing them to load external dynamic link libraries. These injected DLLs facilitate the execution of arbitrary code within the target process, granting attackers the ability to manipulate its behavior, extract sensitive data, or escalate privileges.

There is a tool called Sysmon which is a great tool that logs system activity to the Windows event log. It provides detailed information about process creation, network connections, file creation, registry changes, and more. Leveraging Sysmon, we can effectively detect DLL injection attempts and enhance our security posture.

Prerequisite: Need to install Visual studio community 2022

[Visual Studio 2022 Community Edition – Download Latest Free Version
\(microsoft.com\)](https://www.microsoft.com/en-us/p/visual-studio-community/9nblggh44kzr)



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

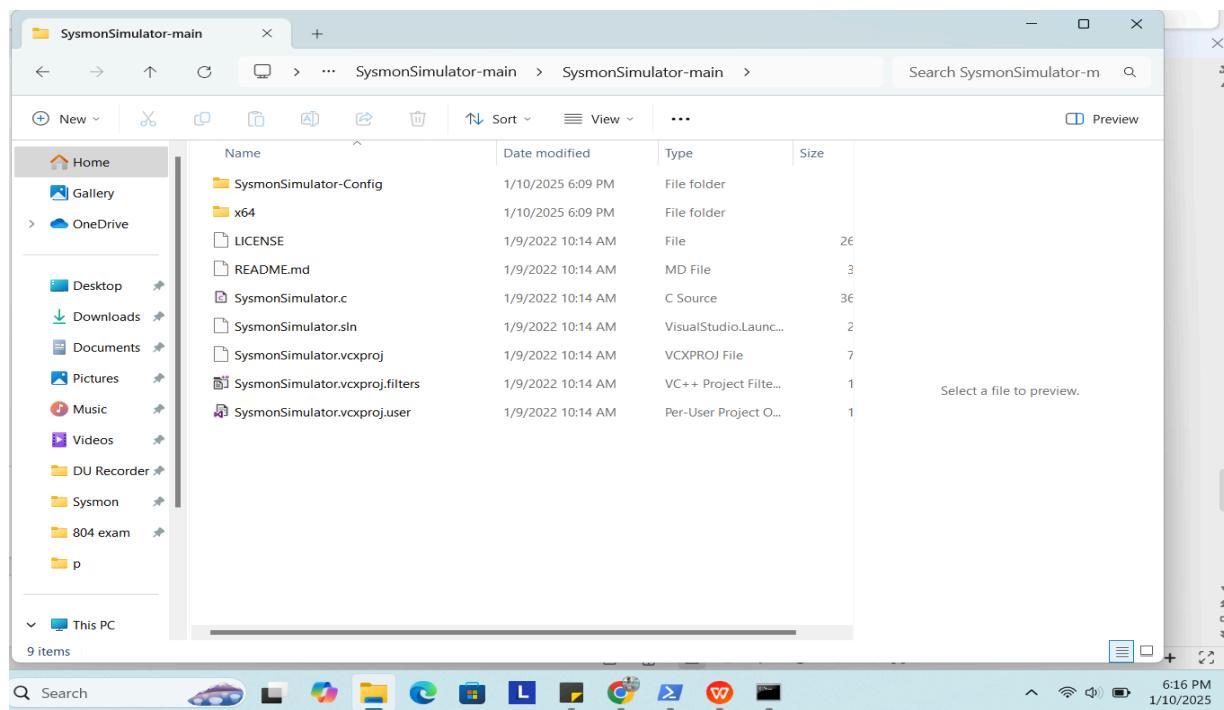
C:\Windows\System32>cd C:\Users\kowsh\OneDrive\Desktop\804 exam\Sysmon

C:\Users\kowsh\OneDrive\Desktop\804 exam\Sysmon>Sysmon64.exe -accepteula -i sysmonconfig-export.xml

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.90
Configuration file validated.
The service Sysmon64 is already registered. Uninstall Sysmon before reinstalling.

C:\Users\kowsh\OneDrive\Desktop\804 exam\Sysmon>
```



Executed the following command (as administrator) to install the Sysmon driver:

SysmonSimulator.c

```
#include <winsock.h>
#include <windows.h>
#include <tdio.h>
#include <winternl.h>
#include <tchar.h>
#include <wbeidl.h>

#pragma comment(lib,"WS2_32")
#pragma comment(lib,"dnsapi")
#pragma comment(lib, "ntdll")
#pragma comment(lib, "wbemuuid")

#define _WIN32_DCOM

EXTERN_C NTSTATUS NtTerminateProcess(HANDLE, NTSTATUS);
EXTERN_C NTSTATUS NTAPI NtReadVirtualMemory(HANDLE, PVOID, PVOID, ULONG, PULONG);
EXTERN_C NTSTATUS NTAPI NtWriteVirtualMemory(HANDLE, PVOID, PVOID, ULONG, PULONG);
EXTERN_C NTSTATUS NTAPI NtGetContextThread(HANDLE, PCONTEXT);
EXTERN_C NTSTATUS NTAPI NtSetContextThread(HANDLE, PCONTEXT);
EXTERN_C NTSTATUS NTAPI NtUnmapViewOfSection(HANDLE, PVOID);
EXTERN_C NTSTATUS NTAPI NtResumeThread(HANDLE, PULONG);

typedef NTSTATUS (WINAPI* _NtQueryInformationProcess) (
    HANDLE,
    PROCESSINFOCLASS,
    PVOID.
```

No issues found

Output

Show output from:

Ready

Air: Very Poor Now

6:17 PM 1/10/2025

We opened the sysmon simulator folder in **Visual studio community** and ran the **sysmonsimulator.c** file as “Open a project or solution” option.

Microsoft Visual Studio Debug

```
Usage: SysmonSimulator.exe -eid <event id>
C:\Users\kows\OneDrive\Desktop\b04 exam\SysmonSimulator-main\SysmonSimulator-main\Debug\SysmonSimulator.exe (process 13716) exited with code 0 (0x0).
Press any key to close this window . . .

10X
Ou
Sh
1> ----- Build started: Project: SysmonSimulator, Configuration: Debug|Win32 -----
1> SysmonSimulator.C
1> SysmonSimulator.vcxproj -> C:\Users\kows\OneDrive\Desktop\b04 exam\SysmonSimulator-main\SysmonSimulator-main\Debug\SysmonSimulator.exe
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****

Build succeeded.
```

22°C Haze

6:18 PM 1/10/2025

Here the build was successful and we got a **SysmonSimulator.exe** file in debug folder.

This PC > New Volume (D:) > Download > SysmonSimulator-main > Debug				
	Name	Date modified	Type	Size
is	SysmonSimulator	1/12/2025 3:58 PM	Application	72 KB
ls	SysmonSimulator.pdb	1/12/2025 3:58 PM	Program Debug D...	1,148 KB

This execution gave us a sysmonsimulator.exe file which we used to run the injection to identify whether the sysmon is working or not and also to generate a log for wazuh.

```
>SysmonSimulator.exe -h
```

```
C:\Users\kowsh\OneDrive\Desktop\804 exam\SysmonSimulator-main\SysmonSimulator-main\Debug>SysmonSimulator.exe -h

Sysmon Simulator v0.1 - Sysmon event simulation utility
  A Windows utility to simulate Sysmon event logs

Usage:
Run simulation : .\SysmonSimulator.exe -eid <event id>
Show help menu : .\SysmonSimulator.exe -help

Example:
SysmonSimulator.exe -eid 1

Parameters:
-eid 1 : Process creation
-eid 2 : A process changed a file creation time
-eid 3 : Network connection
-eid 5 : Process terminated
-eid 6 : Driver loaded
-eid 7 : Image loaded
-eid 8 : CreateRemoteThread
-eid 9 : RawAccessRead
-eid 10 : RawAccessWrite
-eid 11 : FileCreate
-eid 12 : RegistryEvent - Object create and delete
-eid 13 : RegistryEvent - Value Set
-eid 14 : RegistryEvent - Key and Value Rename
-eid 15 : FileCreateStreamHash
-eid 16 : ServiceConfigurationChange
-eid 17 : PipeEvent - Pipe Created
-eid 18 : PipeEvent - Pipe Connected
-eid 19 : WmiEvent - WmiEventFilter activity detected
-eid 20 : WmiEvent - WmiEventConsumer activity detected
-eid 21 : WmiEvent - WmiEventConsumerToFilter activity detected
-eid 22 : DNSEvent - DNS query
-eid 24 : ClipboardChange - New content in the clipboard
-eid 25 : ProcessTampering - Process Image change
-eid 26 : FileDeleteDetected - File Delete logged

Description:
Enter an event ID from the above parameters list and the related Windows API function is called
to simulate the attack and Sysmon event log will be generated which can be viewed in the Windows Event Viewer

Prerequisite:
Search
```

Once Sysmon is installed and configured, let's test it using SysmonSimulator. With SysmonSimulator, we can simulate various attacks using WINAPIs to generate the necessary logs for detection.

>*SysmonSimulator.exe -eid 8*

```
C:\Users\kowsh\OneDrive\Desktop\804 exam\SysmonSimulator-main\SysmonSimulator-main\Debug>SysmonSimulator.exe -eid 8

[+] Simulation : Started successfully
[+] Event ID : 8
[+] Event Name : Create Remote Thread Event
[+] Inject into : PID 10968
[+] Opened process's handle
[+] Created Remote Thread
[+] Closed Handle to the process
[+] Event Viewer : Check Sysmon Event ID 8 for detection
[+] Event Time : Event 8 simulation is performed on 10/01/2025 at 18:22:02

C:\Users\kowsh\OneDrive\Desktop\804 exam\SysmonSimulator-main\SysmonSimulator-main\Debug>
```

Event Viewer

File Action View Help

Operational Number of events: 20,202

Level	Date and Time	Source	Event ID	Task Category
Information	1/10/2025 6:22:45 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:45 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:40 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:06 PM	Sysmon	3	Network connection detected (...)
Information	1/10/2025 6:22:05 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:05 PM	Sysmon	11	File created (rule:FileCreate)
Information	1/10/2025 6:22:02 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:02 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:02 PM	Sysmon	5	Process terminated (rule: Proce...
Information	1/10/2025 6:22:02 PM	Sysmon	8	CreateRemoteThread detected ...
Information	1/10/2025 6:22:02 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:02 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:22:02 PM	Sysmon	5	Process terminated (rule: Proce...
Information	1/10/2025 6:20:29 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:20:12 PM	Sysmon	11	File created (rule:FileCreate)
Information	1/10/2025 6:20:05 PM	Sysmon	3	Network connection detected (...)
Information	1/10/2025 6:19:47 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:19:47 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:19:47 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:19:47 PM	Sysmon	1	Process Create (rule:ProcessCre...
Information	1/10/2025 6:19:47 PM	Sysmon	1	Process Create (rule:ProcessCre...

Ubuntu

Change local rules.xml

Activities Google Chrome

Not secure https://10.33.3.10:9200/_app/wazuh#/manager/?tab=rules

wazuh elastic

Management Rules

< local_rules.xml

```

1 <!-- Local rules -->
2 <!-- Modify it at your will. -->
3 <!-- Copyright (C) 2015, Wazuh Inc. -->
4
5 <!-- Example-->
6 <group name="local,syslog,sshd">
7
8 <!--
9   Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
10  -->
11  <rule id="100001" level="5">
12    <if _id=="$!f_id">
13      <if _id=="$!f_ip">
14        <srcip>1.1.1.1</srcip>
15        <description>sshd: authentication failed from IP 1.1.1.1.</description>
16        <group>authentication_failed,pc1_dss_10.2.4,pc1_dss_10.2.5,</group>
17      </if>
18    </rule>
19  </group>
20
21
22
23 <group name="windows,synon">
24  <rule id="100002" level="12">
25    <if _id=="$!f_id">
26      <description>Possible process injection activity detected from "$(_win.eventdata.sourceImage)" on "$(_win.eventdata.targetImage)"</description>
27      <ntrule>
28        <if _id=="$!f_ip">
29          <description>Ignore Windows binaries and Chrome</description>
30        </if>
31      </ntrule>
32    <rule id="100100" level="0">
33      <if _id=="$!f_id">
34        <field name="$(_win.eventdata.sourceImage)" type="pcre2">>(C:\Windows\system32)\chrome.exe</field>
35      </if>
36    </rule>
37  </group>

```

Ruleset Test Save

Alert Generated!

The screenshot shows the Wazuh Security Events interface. At the top, there are three pie charts: one for Application Compatibility U..., one for Windows detections, and one for Symon - Suspicious Process anomalies. Below these are sections for Security Alerts and Rule Details.

Security Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 10, 2025 @ 20:27:33.696	T1055.001	Defense Evasion, Privilege Escalation	Possible process injection activity detected from "C:\Users\likowsh\OneDrive\Desktop\804 exam\SysmonSimulator-main\Debug\SysmonSimulator.exe" on "<unknown process>"	12	100200
Jan 10, 2025 @ 19:55:22.007	T1546.011	Privilege Escalation, Persistence	Application Compatibility Database launched	12	92058
Jan 10, 2025 @ 19:42:35.013	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
Jan 10, 2025 @ 19:42:35.001	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
Jan 10, 2025 @ 19:42:08.630	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
Jan 10, 2025 @ 18:55:21.889	T1546.011	Privilege Escalation, Persistence	Application Compatibility Database launched	12	92058
Jan 10, 2025 @ 18:46:47.045	T1055	Defense Evasion, Privilege Escalation	Syomon - Suspicious Process - explorer.exe	12	61640

Rows per page: 10

This screenshot shows the detailed view for the alert with Rule ID 100200. The alert details are as follows:

id: 1736519253.3816178
input.type: log
location: EventChannel
manager.name: lab707-10
rule.description: Possible process injection activity detected from "C:\Users\likowsh\OneDrive\Desktop\804 exam\SysmonSimulator-main\Debug\SysmonSimulator.exe" on "<unknown process>"
rule.firetimes: 1
rule.groups: windows, sysmon
rule.id: 100200
rule.level: 12
rule.mail: true
rule.mitre.id: T1055.001
rule.mitre.tactic: Defense Evasion, Privilege Escalation
rule.mitre.technique: Dynamic-link Library Injection
timestamp: 2025-01-10T20:27:33.696+0600

Associated Alerts

Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jan 10, 2025 @ 19:55:22.007	T1546.011	Privilege Escalation, Persistence	Application Compatibility Database launched	12	92058
Jan 10, 2025 @ 19:42:35.013	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213

END

SHELL SHOCK ATTACK

What is a Shellshock attack?

Shellshock is a vulnerability that affects Unix-based operating systems, including Linux distributions like Ubuntu, and web servers like Apache. The vulnerability resides in the Bash shell, which is a command-line interpreter widely used on Unix-like operating systems. In a Shellshock attack targeting an Apache2 server on Ubuntu, the attacker exploits this vulnerability to execute arbitrary commands on the server. This can lead to various security breaches, such as data theft, unauthorized access, or the installation of malware. In our shellshock attack we used **HTTP requests** for sending malicious code embedded into the headers.

Let's start, Install Apache2 and start the server:

```
sudo apt update
```

```
Sudo apt install apache2
```

```
sudo ufw app list
```

```
sudo
```

```
ufw allow 'Apache'
```

```
sudo ufw status
```

```
sudo systemctl status apache2
```

```
root@ubuntu:/home/vboxuser# sudo apt update
Hit:1 https://packages.wazuh.com/4.x/apt stable InRelease
Hit:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://ppa.launchpad.net/oisf/suricata-stable/ubuntu focal InRelease
Hit:5 http://bd.archive.ubuntu.com/ubuntu focal InRelease
Get:6 http://bd.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:7 http://bd.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 128 kB in 2s (62.8 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu:/home/vboxuser# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-l
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libap
0 upgraded, 8 newly installed, 0 to remove and 6 not upgraded.
Need to get 1,723 kB of archives.
After this operation, 7,535 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
root@ubuntu:/home/vboxuser# sudo systemctl status apache2
* apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2025-01-12 16:10:49 +06; 53s ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 65503 (apache2)
   Tasks: 55 (limit: 6995)
   Memory: 5.2M
  CGroup: /system.slice/apache2.service
          |-65503 /usr/sbin/apache2 -k start
          |-65504 /usr/sbin/apache2 -k start
          `--65505 /usr/sbin/apache2 -k start
```

```
root@ubuntu:/home/vboxuser# sudo ufw app list
Available applications:
 Apache
 Apache Full
 Apache Secure
 CUPS
```

```
root@ubuntu:/home/vboxuser# sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
```

```
vboxuser@ubuntu:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
```

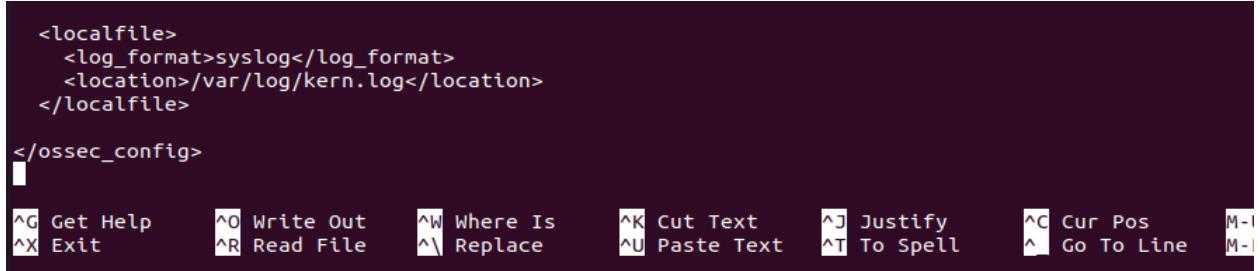
Add some rules in Wazuh Agent

As we want to collect data of the shellShock attack on apache2 server we need to update some rules in “ossec.conf” file so that our wazuh agent sends us the log.

```
root@ubuntu:/home/vboxuser# cd /var/ossec/
root@ubuntu:/var/ossec# ls
active-response agentless api backup bin etc framework integrations lib logs queue ruleset stats templates tmp var wodles
root@ubuntu:/var/ossec# cd etc
root@ubuntu:/var/ossec/etc# ls
client.keys internal_options.conf local_internal_options.conf ossec.conf rules sslmanager.cert
decoders lists localtime rootcheck shared sslmanager.key
root@ubuntu:/var/ossec/etc# nano ossec.conf
```

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/kern.log</location>
</localfile>

</ossec_config>
```



```
root@ubuntu:/home/vboxuser/soc_setup# systemctl status wazuh-agent
* wazuh-agent.service - Wazuh agent
  Loaded: loaded (/etc/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
  Active: active (exited) since Sun 2025-01-12 18:47:48 +06; 18s ago
    Process: 57773 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)

<9C> 12 18:47:45 ubuntu env[57773]: wazuh-db already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-execd already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-analysisd already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-syscheckd already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-remoted already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-logcollector already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-monitord already running...
<9C> 12 18:47:46 ubuntu env[57773]: wazuh-modulesd already running...
<9C> 12 18:47:48 ubuntu env[57773]: Completed.
<9C> 12 18:47:48 ubuntu systemd[1]: Started Wazuh agent.
root@ubuntu:/home/vboxuser/soc_setup#
```

Launch The Attack

```
sudo curl -H "User-Agent: () { :; }; /bin/cat /etc/passwd" 192.168.1.11
```

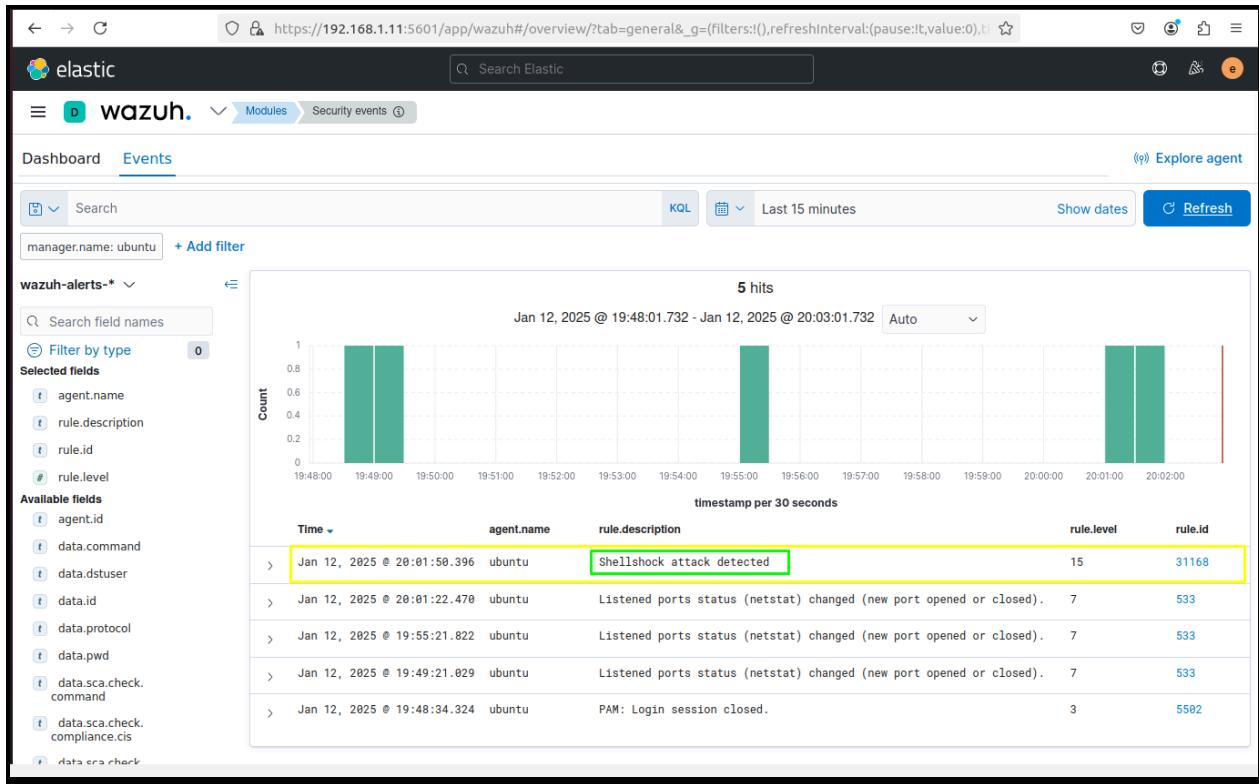
```
└─(kali㉿kali)-[~/Downloads]
$ sudo curl -H "User-Agent: () { :; }; /bin/cat /etc/passwd" 192.168.1.11
[sudo] password for kali:
```

```
Reporting Problems
</div>
<div class="content_section_text">
<p>
    Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
    Apache2 package with Ubuntu. However, check <a
    href="https://bugs.launchpad.net/ubuntu/+source/apache2"
    rel="nofollow">existing bug reports</a> before reporting a new bug.
</p>
<p>
    Please report bugs specific to modules (such as PHP and others)
    to respective packages, not to the web server itself.
</p>
</div>

</div>
</div>
<div class="validator">
</div>
</body>
</html>

└─(kali㉿kali)-[~/Downloads]
$ █
```

Shell Shock Attack Got Detected!



Let's view and analyze the attack with wazuh manager:

Wazuh employs rulesets that define specific conditions indicative of a Shellshock attack. These rules analyze system logs, network traffic, or other relevant data sources for patterns or behaviors associated with the exploit.

This is steps the wazuh HTTP decoder took

Packet Capture: Wazuh Can Capture network traffic using tools like tcpdump. When HTTP traffic passes through the network interface being monitored, Wazuh captures the packets containing HTTP requests and responses.

Packet Analysis: Once captured, Wazuh analyzes the captured packets to identify HTTP traffic. The HTTP decoder examines the packet headers to determine if the packet contains an HTTP request or response.

Header Parsing: When an HTTP packet is identified, the HTTP decoder parses the headers of the request or response. It extracts various components from the HTTP header, such as the request method, URL, headers (like User-Agent), and payload.

Payload Inspection: After parsing the headers, the HTTP decoder inspects the payload of the HTTP request or response. It looks for any anomalies, suspicious patterns, or known attack signatures within the payload.

Rule Matching: Based on the parsed information and payload inspection, Wazuh compares the HTTP traffic against a set of predefined rules designed to detect suspicious or malicious activities. These rules can cover a wide range of threats, including injection attacks, command injection, directory traversal, and more.

Alert Generation: Upon detecting the Shellshock attack payload in the HTTP request, Wazuh generates an alert for each time I launched the attack, indicating a potential Shellshock attack attempt originating from the source IP address of the HTTP request.

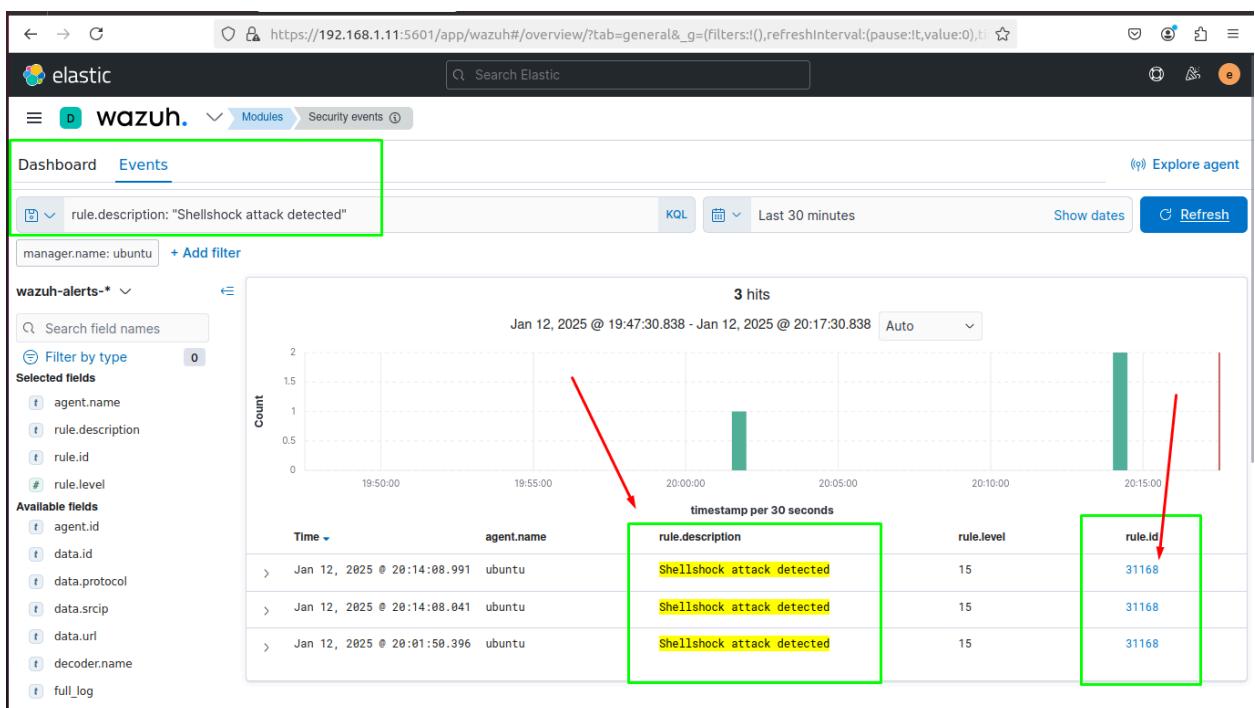
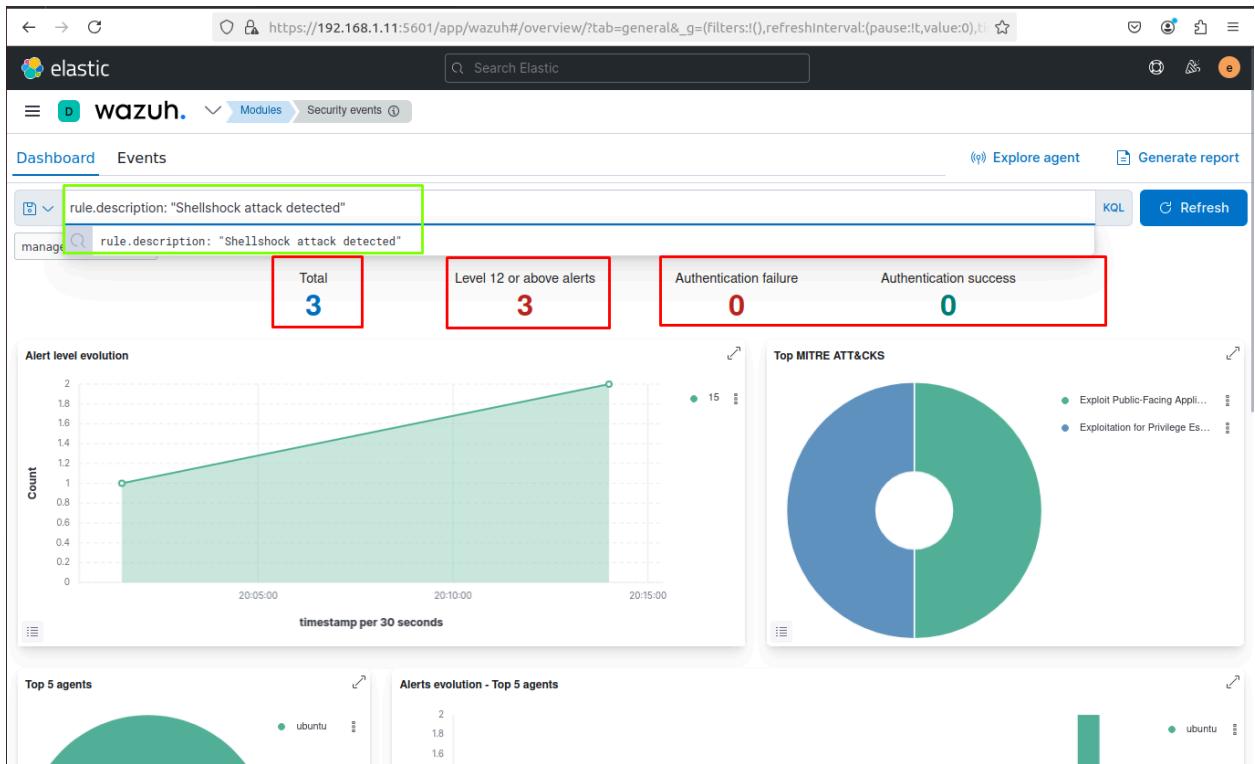
In our attack scenario wazuh used HTTP decoder. It examines HTTP request headers for evidence of the Shellshock payload. So I believe, Wazuh would use an HTTP decoder to parse the web server logs and extract details about the incoming HTTP request, including the User-Agent header containing the potential Shellshock exploit payload

I have launched this attack 3 times thus the number is 3 when we apply rule

rule.description:"Shellshock attack detected"

Here we are looking for a shellshock attack detected match in the log description. As I launched this attack on Agent 002 this is also added in the filters. We could also filter this by rule.id. In case of shellshock the ID will be 31168. We also customize the columns for a better view.

We also customize the columns for a better view.



Wazuh rule and Decoder:

This is the process I followed for rule & decoder detection.

The screenshot shows the Wazuh Events page in the Elastic Stack interface. A modal window titled "Edit filter" is open, allowing the creation of a custom label for a specific rule description. The "Field" is set to "rule.description" and the "Value" is "Shellshock attack detected". The "Custom label" field contains "Shell Shock". The "Save" button is highlighted in blue.

rule.level	rule.id
15	31168
15	31168
7	533

The screenshot shows the Wazuh Events page after applying the custom label "Shell Shock" to the search filter. The search bar now includes "manager.name: ubuntu" and "Shell Shock". The results table shows three hits for the "Shellshock attack detected" rule, all associated with rule.id 31168 and rule.level 15.

Time	agent.name	rule.description	rule.level	rule.id
> Jan 12, 2025 @ 20:14:08.991	ubuntu	Shellshock attack detected	15	31168
> Jan 12, 2025 @ 20:14:08.041	ubuntu	Shellshock attack detected	15	31168
> Jan 12, 2025 @ 20:01:50.396	ubuntu	Shellshock attack detected	15	31168

← → 🔍 https://192.168.1.11:5601/app/wazuh#/manager/rules?tab=rules&redirectRule=31168

elastic Search Elastic

wazuh. Management Rules

Rules (4487)

From here you can manage your rules.

Filter or search

ID ↑	Description
1	Generic template for all syslog rules.
2	Generic template for all firewall rules.
3	Generic template for all ids rules.
4	Generic template for all web rules.
5	Generic template for all web proxy rules.
6	Generic template for all windows rules.
7	Generic template for all wazuh rules.
200	Grouping of wazuh rules.
201	Agent event queue rule
202	Agent event queue is level full.

Rows per page: 10 ~

Shellshock attack detected

View alerts of this Rule

Information

ID 31168	Level 15	File 0245-web_rules.xml	Path ruleset/rules
-------------	-------------	----------------------------	-----------------------

Groups
attack, web, accesslog

Details

If_sid 31108	Regex pattern: "(\() s*{s*\w*;: s*} s*; "(\\) s*{s*\w*;s*} s*;	Info CVE-2014-6271: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
-----------------	--	--

Compliance

GDPR IV_35.7.d	TSC CC6.1, CC6.8, CC7.2, CC7.3
-------------------	-----------------------------------

Related rules

ID ↑	Description	Groups	Compliance	Le... File
311	Access log messages grouped.	web,		0 0245-.....

Rows per page: 10 ~

Shellshock attack detected

View alerts of this Rule

Information

ID: 31168, Level: 15, File: 0245-web_rules.xml, Path: ruleset/rules

Groups: attack, web, accesslog

Details

If_sid: 31108, Regex: pattern: "(\()|s*{s*\w*;:
s*}|s*;|"(\\)|s*{s*\w*;s*}|s*;, Info: CVE-2014-6271: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271>

Compliance

GDPR: IV_35.7.d, TSC: CC6.1, CC6.8, CC7.2, CC7.3

Related rules

Related rule: Access log messages grouped. (File: 0245-.....)

← → 🔍 https://192.168.1.11:5601/app/wazuh#/manager/rules?tab=rules&redirectRule=31168

elastic Search Elastic

wazuh. Management Rules

Rules (1)

From here you can manage your rules.

rule_ids: 31168 × Filter or search

Manage rules files Add new rules file Refresh Export formatted

Custom rules

ID ↑	Description	Groups	Regulatory compliance	Level	File	Path
31168	Shellshock attack detected	attack, web, accesslog	PCI_DSS, GDPR, NIST_800_53, TSC, MITRE	15	0245-web_rules.xml	ruleset/rules

Rows per page: 10 ~

rule_ids: 31168 × Filter or search

Manage rules files Add new rules file Refresh Export formatted

Custom rules

ID: 31168, Description: Shellshock attack detected, Groups: attack, web, accesslog, Regulatory compliance: PCI_DSS, GDPR, NIST_800_53, TSC, MITRE, Level: 15, File: 0245-web_rules.xml, Path: ruleset/rules

< 0245-web_rules.xml

```
313    Shellshock detected
314    Code: 2xx, 3xx
315    -->
316    <rule id="31168" level="15">
317        <if_sid>31108</if_sid>
318        <regex>"\(\)\s*\{\s*\w*:\;\s*\}\s*;|\"\(\)\s*\{\s*\w*;\s*\}\s*;"</regex>
319        <description>Shellshock attack detected</description>
320    </rule>
321    <rule id="31169" level="15">
322        <if_sid>31108</if_sid>
323        <regex>"\(\)\s*\{\s*_;\.*\}\s*[_\$\(\$\(\))]\s*;"</regex>
324        <description>Shellshock attack detected</description>
325    </rule>
326    <rule id="31170" level="6">
327        <if_sid>31108</if_sid>
328        <url>%2csleep|sysdate()|nslookup%20dns.sql</url>
329        <description>SQL injection attempt.</description>
330    </rule>
331    <rule id="31171" level="6">
332        <if_sid>31108</if_sid>
333        <url>%2cselect|sysdate()|nslookup%20dns.sql</url>
334        <description>SQL injection attempt.</description>
335    </rule>
336    <rule id="31172" level="6">
337        <if_sid>31108</if_sid>
338        <url>%2cselect|sysdate()|nslookup%20dns.sql</url>
339        <description>SQL injection attempt.</description>
340    </rule>
341    <rule id="31173" level="6">
342        <if_sid>31108</if_sid>
343        <url>%2cselect|sysdate()|nslookup%20dns.sql</url>
344        <description>SQL injection attempt.</description>
345    </rule>
346    <rule id="31174" level="6">
347        <if_sid>31108</if_sid>
348        <url>%2cselect|sysdate()|nslookup%20dns.sql</url>
349        <description>SQL injection attempt.</description>
350    </rule>
```

Rules for Shell Shock Attack

END