Eksploatacija ranjivosti, detekcija, i Incident Response izveštaj

Ime studenata: Milica Đumić i Vasilije Zeković  
Datum: 17.11.2024.

Pregled Ranljivosti

* 1. **Informacije o ranljivosti**

ID ranljivosti (CVE): *CVE-2014-3704*  
Pogođen servis: *Drupal*  
CVSS ocena: *9.8*

**Opis ranljivosti:**

Slanjem *HTTP* *POST* zahteva, upotrebom *HTTP* protokola, korišćenjem porta 80 i *TCP* transportnog protokola, preko login stranice na *URI* *localhost/drupal/?q=node&destination=node,* sa parametrima koji su maliciozno definisani od strane korisnika, uspešno se izvršava napad *SQL* injekcijomnad *Drupal* servisom. Ovaj napad može dovesti do različitih zloupotreba nad bazom podataka, izvršavanjem proizvoljnih *PHP* skripti kao i instalacije *backdoors* programa. Direktno je pogođena metoda *protected function expandArguments(&$query, &$args)* apstrakne klase *abstract class DatabaseConnection extends PDO.* Pomenuta klasa i metoda pripadaju *Database Abstraction* sloju, koji omogućava podršku različitim bazama podataka.

1.2 **Opis eksploita**

**Metod eksploatacije:**

Potrebno je podesiti *URL* adresu sajta koju napadač cilja. Slanjem *HTTP POST* zahteva sa malicioznim parametrima na *login* stranicu se dodaje novi nalog u tabelu korisnika, a zatim se u tabeli uloga tom novododatom nalogu dodeljuje uloga administratora*.* U slučaju uspešne eksploatacije ranjivosti, napadač ima pristup novom administratorskom nalogu. Ovaj eksploit poseduje i proširenje u vidu pokretanja *.php* skripte kroz objavu članka od strane administratora i na taj način se ostvaruje pristup meterpreteru. Fokus izveštaja je na osnovnoj funkcionalnosti eksploita – primena *SQL* injekcije zarad ostvarivanja administratorskih privilegija na *Drupal* servisu.

Proces Eksploatacije

2.1 **Podešavanje eksploita  
  
Ranljiv cilj:**  
U pitanju je *metasploitable*3 ranjiva mašina, koja pokreće ranjiv *Drupal* servis sa verzijom 7.5 koji trči na portu 80.

**Alati za eksploataciju:**

Korišćen je *Metasploit* alat za eksploataciju ranjivosti. Odabran je eksploit pod nazivom: *“Drupal HTTP Parameter Key/Value SQL Injection”*. Sam eksploit poseduje dve metode primene. Odabrana metoda primene eksploita većim delom odgovara metodologiji eksploita opisanoj u izveštaju o proceni ranjivosti. Dakle, radi efikasnije aplikacije eksploita koristiće se *Metasploit* alat i izabran je [eksploit](https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/http/drupal_drupageddon.rb) koji predstavlja nadskup [prvobitnog eksploita](https://www.exploit-db.com/exploits/34993).

2.2 **Koraci eksploatacije**

Inicijalno se pokreće *metasploit* uz pomoć komande *msfconsole.*

C: \Windows\system32>msfconsole

Zatim se pronađe prethodno opisani eksploit i odabere.

msf6 > search drupal

18 \\_ target: Drupal 7.0 - 7.31 (user-post PHP injection method)

msf6 > use 18

Nakon toga se pomoću komande *info* mogu pronaći dodatne informacije u vezi obaveznih i opcionih parametara samog eksploita. U nastavku će biti inicijalizovani obavezni parametri.

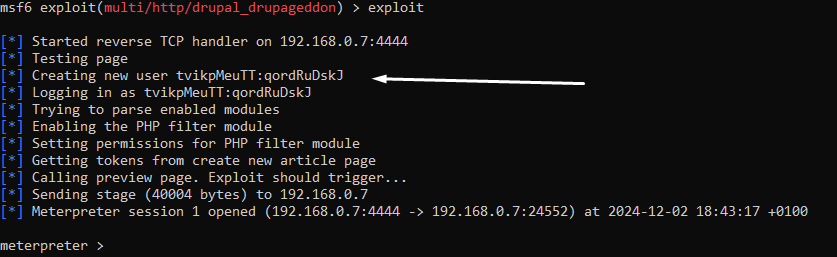
Neophodno je podesiti port, IPadresu i *URI* servisa ranjive mašine.

msf6 exploit(multi/http/drupal\_drupageddon) > set rhosts 172.29.200.11

msf6 exploit(multi/http/drupal\_drupageddon) > set rport 80

msf6 exploit(multi/http/drupal\_drupageddon) > set targeturi /drupal/

2.3 **Rezultat eksploatacije**

Naredbom *exploit* se pokreće eksploatacija.  


Fokus ovog eksploita jeste na primeni *SQL* injekcije. Ona se ostvaruje na samom početku eksploita. Sve kasnije predstavlja eskalaciju eksploatacije ranjivosti i u konačnom ostvarivanja *meterpeter* sesije. Suština eksploita je u prosleđivanju problematičnih parametara *name* koji obično služe za dupli unos podataka, kao što je slučaj kod izmene lozinke pri ponovljenom unosu nove lozinke. Zahvaljujući tome se generišu parametri *name[0] = lozinka* i *name[1] = ponovljena\_lozinka*. Ovo se može zloupotrebiti usled neadekvatne validacije vrednosti 0 i 1 što su zapravo ključevi. Dakle, *Drupal* će prethodno opisane parametre automatski interpretirati u sledeći rečnik:

{

"name": {

"0": "lozinka",

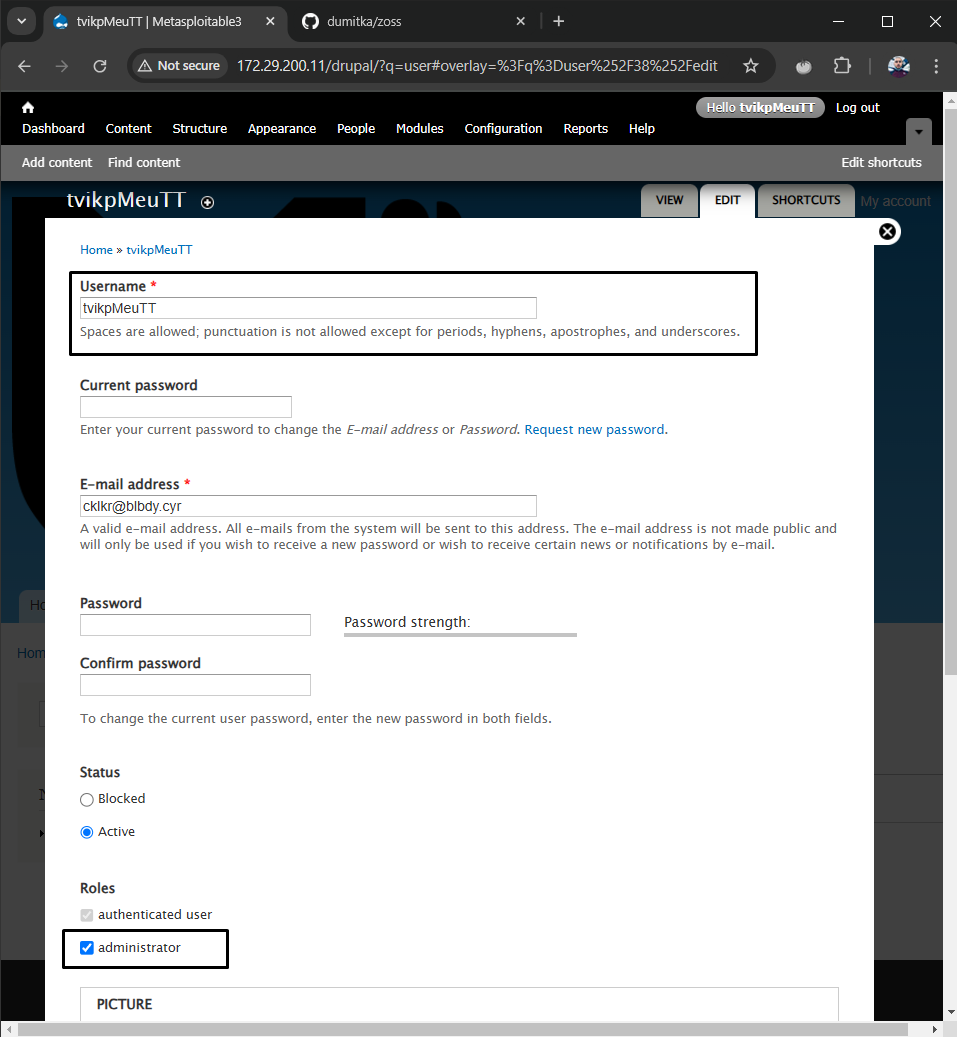
"1": "ponovljena\_lozinka"

}

}

Nakon toga će problematična funkcija izvršiti sređivanje unosa i obaviti kreranje dva nova ključa na osnovu ključa *name*. To su *name\_0* i *name\_1*. Zatim će ključevi biti inicijalizovani na sledeći način, *name\_0 = lozinka* i *name\_1 = ponovljena\_lozinka*. Važna napomena jeste da su *name\_0* i *name\_1* zapravo *placeholder-i* u upitu. Pošto ključevi 0 i 1 nisu validirani, omogućen je unos proizvoljnih vrednosti. Na primer, unos vrednosti “*0; maliciozni upit # zakomentarisanje svega ostalog”* umesto *“*0” omogućava izvršavanje proizvoljnog upita. U slučaju ovog eksploita “*maliciozni\_upit”* odgovara *DDL* naredbi za kreiranje administratorskog naloga.

U nastavku je dat dokaz da se dobijeni kredencijali uspešno koriste za logovanje i da je novom kroisniku dodeljena administratorska uloga.



Detekcija Korišćenjem *Wazuh SIEM-a*

3.1 **Konfiguracija *SIEM-a***

**Podešavanje *Wazuh* agenta:**

Podešavanje započinje u *Wazuh* menadžeru. Sam *Wazuh dashboard* je moguće otvoriti putem *browser-a* na adresi [*http://ip\_adresa\_menadžera:443*](http://ip_adresa_menadžera:443)ukoliko je server vidljiv iz lokalne mašine. Potrebno je ući u meni *Server Management > Endpoints Summary > Deploy new agent*, zatim izabrati opciju *Linux RPM amd64* i uneti IP adresu *Wazuh* menadžera. Ovim postupkom se izgenerišu potrebne komande za konfiguraciju ranjive mašine.

wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent\_4.9.2-1\_amd64.deb && sudo WAZUH\_MANAGER='172.29.200.157' dpkg -i ./wazuh-agent\_4.9.2-1\_amd64.deb

/var/ossec/bin/wazuh-control start

Pokretanjem ovih komandi *Wazuh* agent je uspešno podešen.

**Prikupljanje logova:**

Prikupljanje interesantnih logova *Apache* 2.4.7 webservera, koji je pokrenut u pozadini *Drupal* servisa, se obavlja u datoteci */var/log/apache2/access.log*.

Primer sloga:  *172.29.192.1- - [17/Nov/2024:11:02:45 +0000] "POST /drupal/?q=user/login HTTP/1.1" 302 534 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36"*

Nedostatak standardnog prikupljanja od strane *Apache* servera predstavlja nemogućnost adekvatne izmene sloga *access.log* datoteke. Eksploit prilikom slanja *HTTP POST* zahteva maliciozne parametre prosleđuje u telu zahteva. Nažalost, telo zahteva nije moguće zapisati u access.*log* datoteci. Kada bi eventualno postojala takva opcija, bilo bi dovoljno izmeniti postojeća pravila (*0245-web\_rules.xml*) i dekodere (*0025-apache\_decoders.xml*) koji su zaduženi za registrovanje napada *SQL* injekcijom.

Pošto je neophodno obezbediti detaljnije logovanje nad *Apache* serverom, odabran je modul *Dumpio.* Pokrenute su sledeće komande radi ispravne konfiguracije *Dumpio* modula.

Uključivanje *Dumpio* modula:

$ sudo a2enmod dump\_io

U *apache2.conf* fajlu na putanji /etc/apache2/ upisati sledeće:

LogLevel dumpio:trace7

DumpIOInput On

DumpIOOutput On

Restartovati server:

$ sudo service apache2 restart

Zatim je potrebno dodati deklaraciju, ukoliko prethodno ne postoji, u datoteci *ossec.conf* na mašini agenta na putanji */var/ossec/etc/*. Zahvaljujući toj deklaraciji će agent prikupljati *error* logove i otpremati ih *Wazuh* menadžeru.

<localfile>

<log\_format>apache</log\_format>

<location>/var/log/apache2/error.log</location>

</localfile>

Primer sloga u *error.log* datoteci:

*[Sun Dec 01 14:21:38.902128 2024] [dumpio:trace7] [pid 9324] mod\_dumpio.c(103): [client 172.29.192.1:10782] mod\_dumpio: dumpio\_in (data-HEAP): name%5b0%20%3binsert%20into%20users%20%28uid%$...*

Svaka linija HTTP zahteva ili odgovora se beleži od strane *Dumpio* modula i predstavlja jednu liniju (slog) u *error.log* datoteci. Ovime je omogućeno iščitati *name* parametar prosleđen u telu zahteva koji je karakterističan za *Drupal SQL injection* napad.

Sledi konfiguracija dekodera kako bi se adekvatno iščitavao deo sloga svakog zahteva u kojem se posebno prepoznaje deo sa slanjem parametara u telu zahteva. Poenta je uhvatiti sve unutar *name* parametra.

<localfile>

<log\_format>apache</log\_format>

<location>/var/log/apache2/error.log</location>

</localfile>

* 1. ***Wazuh SIEM* pravila**

**Pravila korišćena za detekciju:**

U *Wazuh*-u ne postoji dekoder koji adekvatno prepoznaje slogove *error.log*. Prvenstveno je bilo potrebno napisati dekoder kojime će se obaviti adekvatan *prematch* kako bi se maliciozna vrednost parametra *name* namapirala na promenljivu koja se može koristiti u pravilu koje je kasnije definisano. Korišćenjem sledeće vrednosti *prematch* taga će se nedvosmisleno izdvojiti svi parametri zahteva sa problematičnim nazivom *name*.

**Korisnički definisan dekoder:**

<decoder name="drupal\_decoder\_v2">

<prematch>mod\_dumpio: dumpio\_in \(data-HEAP\): name</prematch>

</decoder>

<decoder name="drupal\_decoder\_v2\_child">

<parent>drupal\_decoder\_v2</parent>

<regex offset="after\_parent">(\S+)</regex>

<order>illegal\_parameter\_value</order>

</decoder>

**Korisnički definisano pravilo:**

Ovo pravilo se okida na osnovu dekodovanih slogova od strane dekodera sa nazivom *drupal\_decoder\_v2.* Prepoznata je promenljiva *illegal\_parameter\_value* u kojoj se nalazi potencijalno maliciozni sadržaj. Zatim se dinamički proverava da li ta vrednost sadrži neku od zabranjenih reči. *ID* je dodeljenja vrednost 100002 iz razloga što su vrednosti u opsegu od 100000 do 120000 rezervisane za korisnička pravila. Dodeljujemo *level* 13, što znači da aktivacija ovog pravila pripada kategoriji ozbiljnih bezbednosnih pretnji. Pravila numerisana nivoom 13 predstavljaju jasno utvrđene obrasce napada. Pravilo koje je definisano je vrlo specifično i bilo koji korisnik koji svojim delovanjem aktivira ovo pravilo defintivno ima maliciozne namere. Njegov pokušaj najverovatnije rezultuje uspešnom eksploatacijom ranjivosti koja ima ozbiljne posledice po sistem.

<group name="drupal">

<rule id="100002" level="13">

<decoded\_as>drupal\_decoder\_v2</decoded\_as>

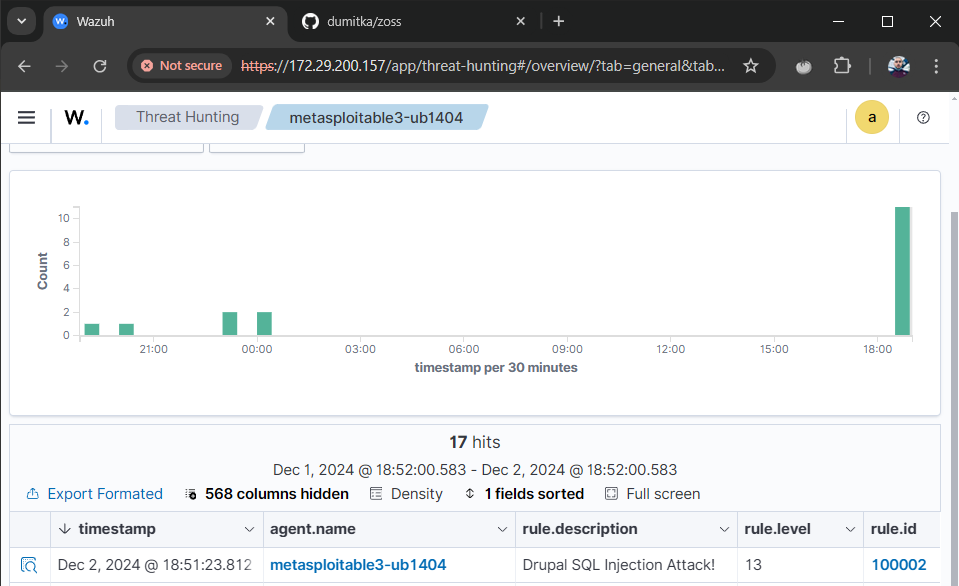
<field name="illegal\_parameter\_value">insert|union|delete|update|drop|select</field>

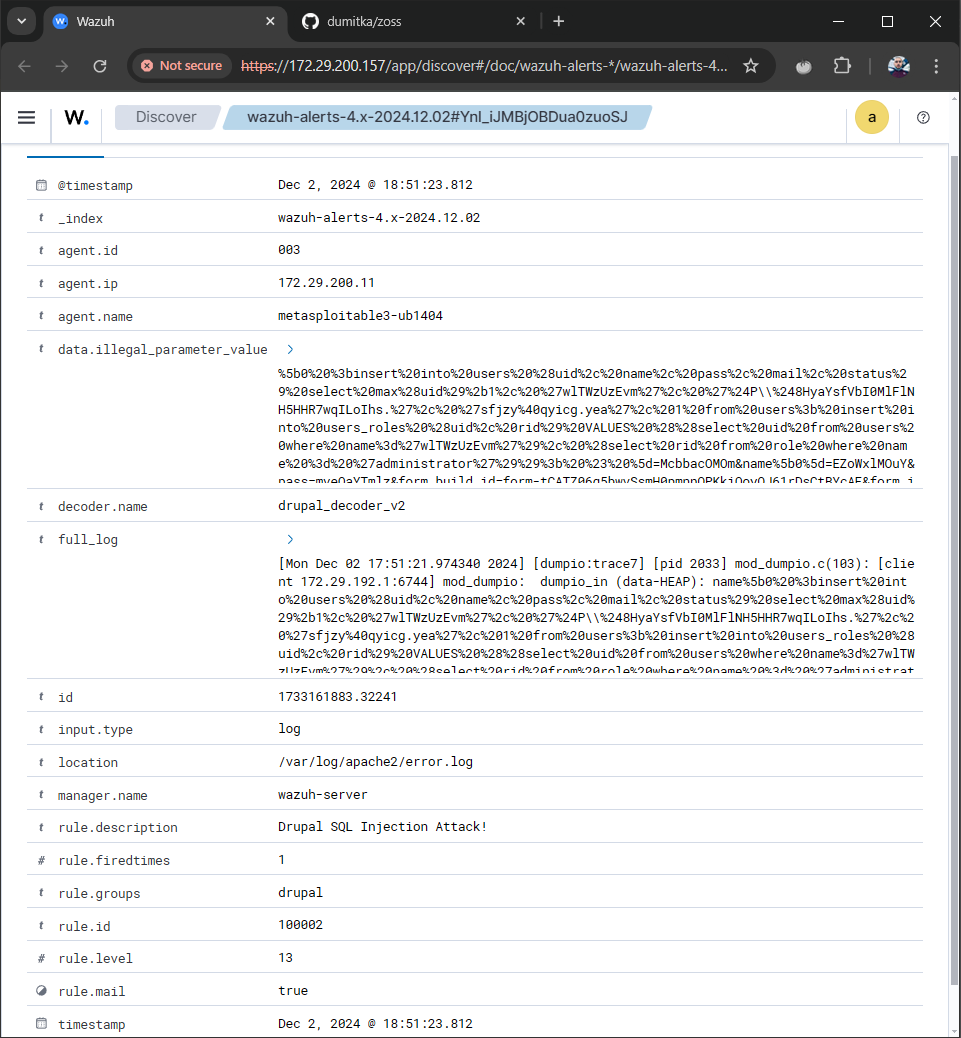
<description>Drupal SQL Injection Attack!</description>

</rule>

</group>

* 1. **Proces detekcije**

U sekciji *Wazuh* menadžera *Threat Intelligence >* *Threat Hunting* je moguće uočiti pretnju koja se tiče *SQL injection* napada što je bio cilj.  
  




Incident Response sa The Hive-om

4.1 **Podešavanje integracije**

**Postupak u *TheHive-u***

Ukoliko je prethodno instaliran *Docker*, instalacija *TheHive-a* se može jednostavno obaviti pokretanjem komandi *docker pull* i *docker run* u zavisnosti od odabrane verzije *TheHive-a.* Konkretnije komande kao i verzije su dostupne na [*Docker Hub-u*](https://hub.docker.com/r/strangebee/thehive/tags)*.*

Nakon što je instaliran *TheHive* potrebno je napraviti organizaciju sa administratorskim nalogom. Zatim kreirati korisnika uz administratorske permisije koji je dodeljen toj organizaciji. Njemu je potrebno dodati lozinku kako bi mogao da se prijavi. Tako kreiran korisnik ima mogućnost da kreira nove korisnike, upravlja slučajevima i uzbunama. Integracija sa *Wazuh-om* je moguća *pomoću TheHive REST API-a*. Za potrebe ovoga kreiramo korisnika sa permisijama *„analyst*“ i za njega generišemo *API* ključ.

**Postupak u *Wazuh menadžeru***

Ukoliko *Wazuh* menadžer prethodno nije instaliran, dovoljno je [preuzeti .ova sliku](https://documentation.wazuh.com/current/installation-guide/index.html) i importovati je u alat za pokretanje virtuelnih mašina (npr. *Oracle VirtualBox*).

Potrebno je instalirati python modul za *TheHive*, komandom:

sudo /var/ossec/framework/python/bin/pip3 install thehive4py==1.8.1

Kreirati [integracionu skriptu](https://pastebin.com/KLPaKmWL) u fajlu */var/ossec/integrations/custom-w2thive.py*. Vrednost *lvl\_threshold* unutar skripte odgovara minimalnom *level-u* upozorenja koji će biti prosleđen *TheHive-u*. Postavka *lvl\_threshold-a* na 0 obezbeđuje registrovanje svih događaja. Potrebno je napraviti i [*bash* skriptu](https://pastebin.com/dDrQmPmQ) u istom direktorijumu sa nazivom *custom-w2thive.* Ona će će pokrenuti *custom-w2thive.py*.

Potrebno je još dodeliti adekvatne permisije *Wazuh-u* tako što unosimo sledeće komande:

sudo chmod 755 /var/ossec/integrations/custom-w2thive.py

sudo chmod 755 /var/ossec/integrations/custom-w2thive

sudo chown root:ossec /var/ossec/integrations/custom-w2thive.py

sudo chown root:ossec /var/ossec/integrations/custom-w2thive

Ukoliko grupa *root:ossec* ne postoji:

sudo groupadd ossec

Što se tiče integracije pravila, da bismo obezbedili da se izvrši integraciona skripta neophodno je da u */var/ossec/etc/ossec.conf* unesemo sledeći kod:

<ossec\_config>

…

<integration>

<name>custom-w2thive</name>

<hook\_url>http://adresa\_hive\_servera(lokalne\_masine):9000</hook\_url>

<api\_key>vBewU6WoZ2KaPWGgEGjToJnPHi/ASBmL</api\_key>

<alert\_format>json</alert\_format>

</integration>

…

</ossec\_config>

Na kraju je potrebno restartovati *Wazuh* menadžera, a to postižemo komandom:

sudo systemctl restart wazuh-manager

4.2 **Kreiranje slučaja u The Hive-u**

**Detalji o slučaju:** 