Eksploatacija ranjivosti, detekcija, i Incident Response izveštaj

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

Pregled Ranljivosti

1.1 **Informacije o ranljivosti**ID ranljivosti (CVE): CVE-2014-3704  
Pogođen servis: Drupal  
CVSS ocena: 7.5  
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 *SQL injection* napad. 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**Izvor eksploita:

<https://github.com/rapid7/metasploit-framework/blob/master//modules/exploits/multi/http/drupal_drupageddon.rb>

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

Za svaku eksploatisanu ranljivost:

2.1 **Podešavanje eksploita**Ranljiv cilj:  
(Opis podešavanja ranjive mašine - koja je verzija servisa, na kom port-u trči)  
U pitanju je *metasploitable* 3 ranjiva mašina, koja pokreće ranjiv *Drupal* servis sa verzijom 7.5 koji trči na portu 80.   
Alati za eksploataciju:  
(Metasploit ili neki drugi alat, Python skripta, Perl skripta…)

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.

2.2 **Koraci eksploatacije**Objasnite proces eksploatacije korak po korak - DETALJNO:  
(Uključite screenshot-ove celog procesa)

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


Zatim se pronađe prethodno opisani eksploit i odabere.  






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

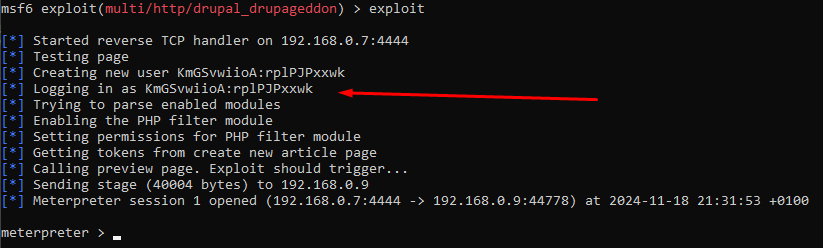
Neophodno je podesiti port i IPadresu ranjive mašine.





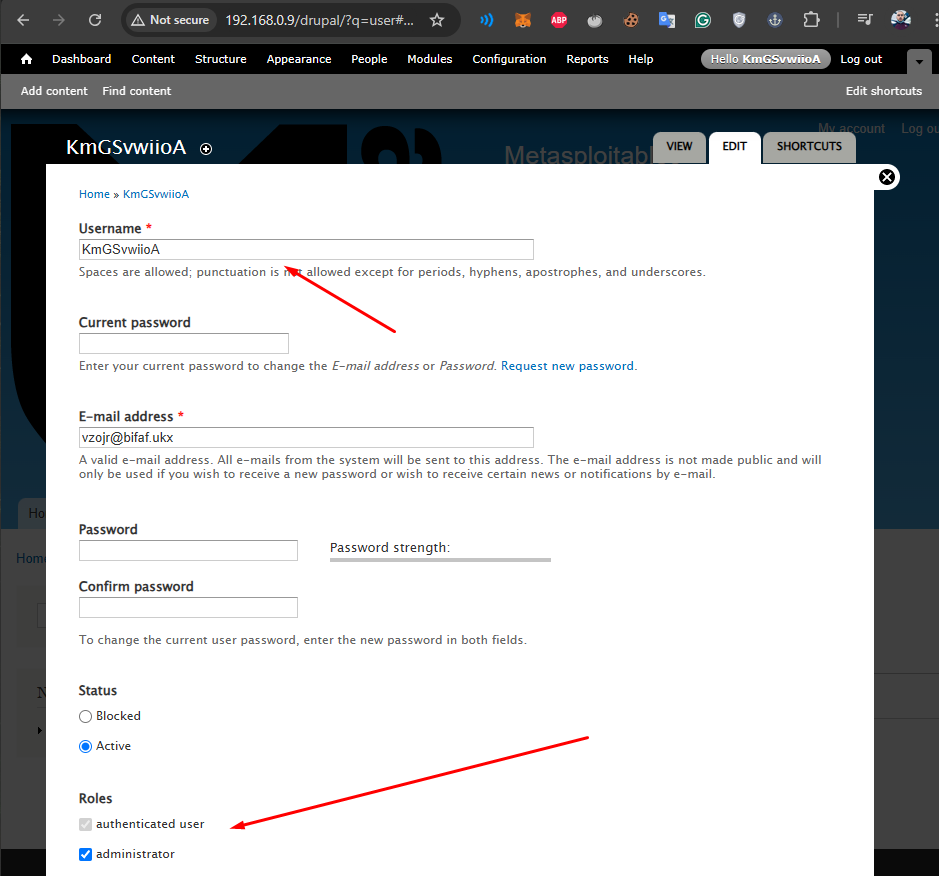
Od neophodnih parametara preostaje još samo podesiti *URI* servisa ranjive mašine.  


2.3 **Rezultat eksploatacije**Prikažite rezultate eksploatacije:  
(Uključite screenshot ekrana uspešne eksploatacije, dajte dokaz da ste eksploatisali sistem)

Naredbom *exploit* se pokreće eksploatacija.  


Kao što se može videtii kreiran je admin sa proizvoljnim kredencijalima i dodatno je otvorena *meterpreter* sesija.

U nastavku sledi dokaz uspešnog prijavljivanja na nalog sa tim kredencijalima.



Detekcija Korišćenjem Wazuh SIEM-a

Za svaku eksploatisanu ranljivost:

3.1 **Wazuh SIEM eravila**Pravila korišćena za detekciju:  
(Navedite specifična Wazuh pravila koja su se aktivirala ili prilagodila za detekciju eksploita)  
ID pravila:  
(Uključite ID pravila i kratak opis)

3.2 **Konfiguracija SIEM-a**Podešavanje Wazuh agenta:  
(Opis konfiguracije agenta na ranjivoj mašini i kako je povezan sa Wazuh Managerom)  
Prikupljanje logova:  
(Navedite koje logove pratite da biste otkrili pokušaje eksploatacije)  
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:   
192.168.0.3 - - [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 *log* datoteke. Eksploit prilikom slanja *HTTP POST* zahteva maliciozne parametre prosleđuje u telu zahteva. Dakle, nije moguće zapisati telo zahteva u *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 *SQL* injection napada. Treba napomenuti da je ovaj tip *SQL injection* napada prepoznat u slučaju da se u *URL*-u zahteva postavljaju maliciozni parametri.   
Pošto je neophodno iskoristiti neki drugi *logging* alat nad *Apache* serverom. Odabran je *ModSecurity.* Pokrenute su sledeće komande radi instalacije, postavke podrazumevane osnovne konfiguracije i osvežavanja servera.  
*$ sudo apt install libapache2-mod-security2*

*$ sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf*

*$ sudo service apache2 reload*

Izgled sloga u *modsec\_audit.log* datoteci:

--4a44714f-A--

[18/Nov/2024:10:14:52 +0000] ZzsTnH8AAAEAABLa3BYAAAAG 127.0.0.1 37044 127.0.0.1 80

--4a44714f-B--

GET /chat/read\_log.php HTTP/1.1

accept: \*/\*

Referer: about:blank

User-Agent: Node.js (linux; U; rv:v4.9.1) AppleWebKit/537.36 (KHTML, like Gecko)

Accept-Language: en

Origin: null

host: localhost

accept-encoding: gzip, deflate

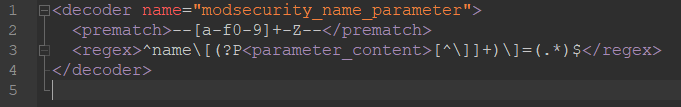
Connection: keep-alive

--4a44714f-Z—

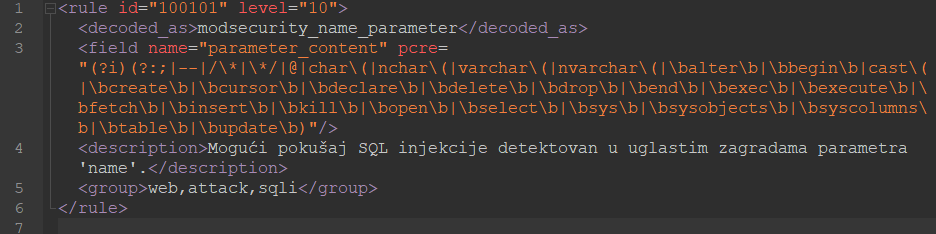
A predstavlja prvu liniju *HTTP POST* zahteva, B predstavlja zaglavlja, a Z predstavlja telo zahteva. Ovakava detaljnost sloga je konfigurisana u fajlu *modsecurity.conf*, na sledeći način.



Sledi konfiguracija dekodera kako bi se adekvatno iščitavao deo sloga svakog zahteva u kojem se posebno prepoznaje deo sa slanjem parametara u telu zahteva – Z sekcija. Poenta je uhvatiti sve unutar *name* parametra, odnosno *name[problematičan\_deo] = value*.

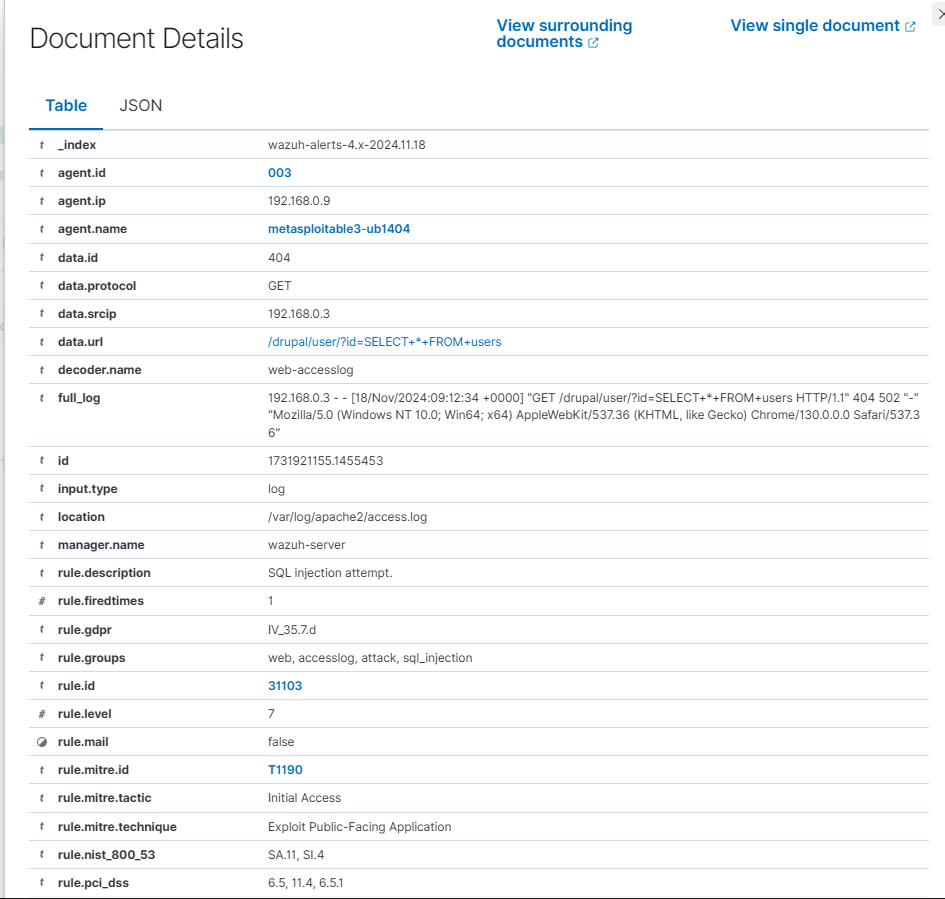


Zahvaljujući sledećem pravilu se iščitava maliciozni *string* parametra *name*.



\*Izgenerisao *ChatGPT.*

3.3 **Proces detekcije**Opišite proces detekcije:  
(Uključite logove ili screenshot-ove koji prikazuju da je napad detektovan pomoću Wazuha)

Ovako izgleda pokušaj *SQL* injekcije za sve eksploite koji parametre prosleđuju putem *URL-a.*   


Incident Response sa The Hive-om

4.1 **Podešavanje integracije**Opis integracije:  
(Objasnite kako je Wazuh integrisan sa The Hive-om za automatizovano kreiranje slučajeva)  
Integracija pravila:  
(Uključite kratak opis pravila koje pokreće kreiranje slučajeva u The Hive-u)

4.2 **Kreiranje slučaja u The Hive-u**Detalji o slučaju:  
(Dajte screenshot-ove koji prikazuju kreirani slučaj u The Hive-u nakon što se Wazuh pravilo aktiviralo)