

1 Отчет по OSCP

1.1 Введение

Данный пейпер посвящен подробному описанию по поиску и эксплуатированию уязвимостей в сети при прохождении сертификации OSCP (Offensive Security Certified Pentester)

Весь нижеуказанный материал предоставлен исключительно в ознакомительных и целях. Автор данного текста не несет ответственности за последствия.

1.2 Исходные данные

Имеется подсеть в диапазоне от 10.11.1.1 по 10.11.1.255. Ничего об этой сети более не известно. Сеть является vpn и эмулирует полностью работу небольшой компании. Сеть представляет из себя набор подсетей, для доступа к которым необходимо пробрасывать тунели со взломанных машин.

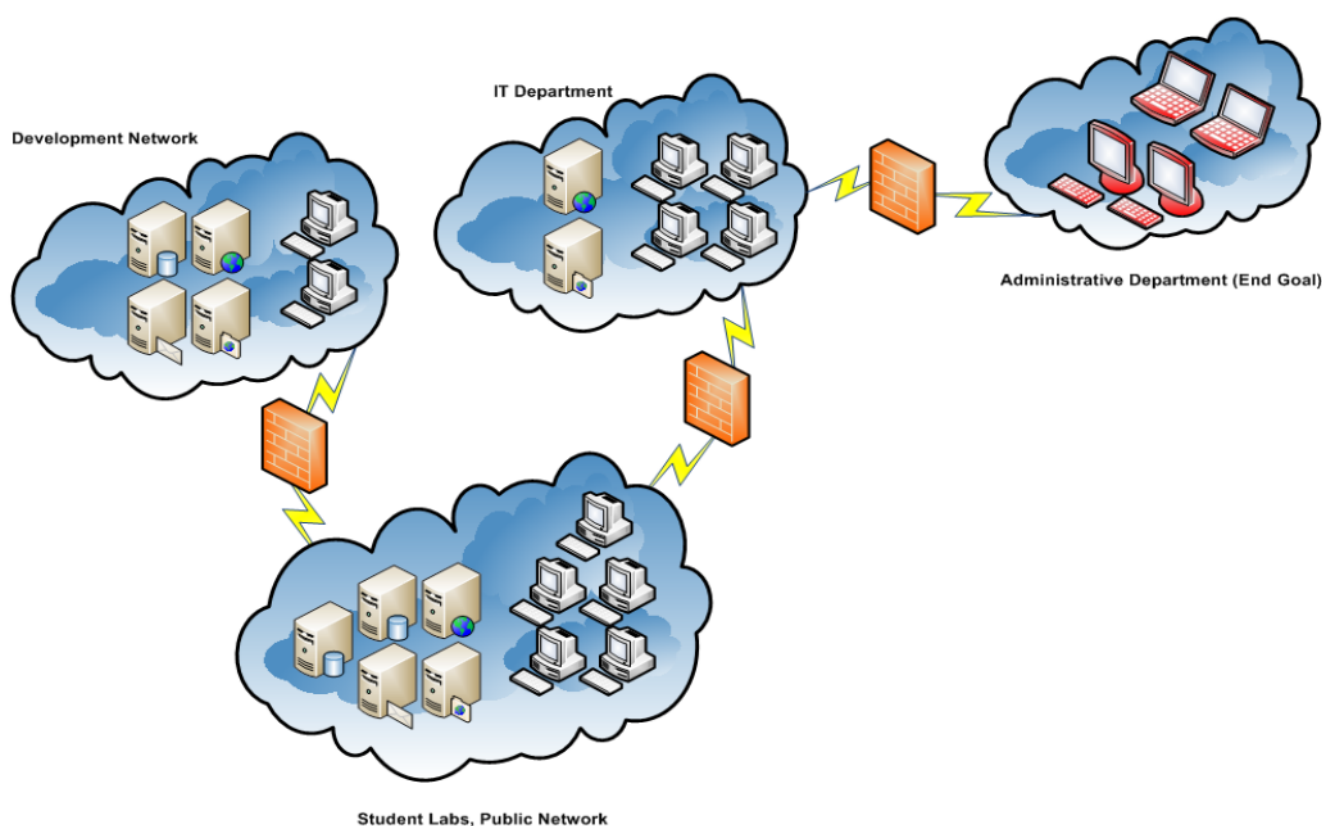


Рис. 1: Макет сети лаборатории.

Задача: собрать как можно больше данных о сети и выбрать первую цель для атаки.

2 Сбор информации

Здесь я опущу большую часть подробностей. Будем использовать классический ping sweep для поиска в сети живых машин. Можно было бы написать свой скрипт на bash, но проще воспользоваться уже готовым вариантом в составе софта nmap. Собственно, просканируем сеть и запишем полученные данные в hosts.txt

Отлично! Теперь посмотрим, что у нас получилось.

```

root@kali:~# nmap -sn 10.11.1.1-255 -oG hosts.txt

Starting Nmap 7.60 ( https://nmap.org ) at 2018-07-17 09:37 MSK
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 18.82% done; ETC: 09:38 (0:00:30 remaining)
Stats: 0:00:10 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 22.16% done; ETC: 09:38 (0:00:35 remaining)
Nmap scan report for 10.11.1.5
Host is up (0.044s latency).
MAC Address: 00:50:56:89:18:41 (VMware)
Nmap scan report for 10.11.1.7
Host is up (0.044s latency).
MAC Address: 00:50:56:89:13:D7 (VMware)
Nmap scan report for 10.11.1.8
Host is up (0.059s latency).
MAC Address: 00:50:56:89:24:79 (VMware)
Nmap scan report for 10.11.1.10
Host is up (0.051s latency).
MAC Address: 00:50:56:89:7F:34 (VMware)

```

Рис. 2: Сканируем всю подсеть на наличие "живых" машин.

```

root@kali:~# cat hosts.txt
# Nmap 7.60 scan initiated Tue Jul 17 09:37:52 2018 as: nmap -sn -oG hosts.txt 10.11.1.1-255
Host: 10.11.1.5 ( ) Status: Up
Host: 10.11.1.7 ( ) Status: Up
Host: 10.11.1.8 ( ) Status: Up
Host: 10.11.1.10 ( ) Status: Up
Host: 10.11.1.13 ( ) Status: Up
Host: 10.11.1.14 ( ) Status: Up
Host: 10.11.1.22 ( ) Status: Up
Host: 10.11.1.24 ( ) Status: Up
Host: 10.11.1.31 ( ) Status: Up
Host: 10.11.1.35 ( ) Status: Up
Host: 10.11.1.39 ( ) Status: Up
Host: 10.11.1.44 ( ) Status: Up
Host: 10.11.1.49 ( ) Status: Up
Host: 10.11.1.50 ( ) Status: Up
Host: 10.11.1.71 ( ) Status: Up
Host: 10.11.1.72 ( ) Status: Up
Host: 10.11.1.73 ( ) Status: Up
Host: 10.11.1.115 ( ) Status: Up
Host: 10.11.1.116 ( ) Status: Up
Host: 10.11.1.125 ( ) Status: Up
Host: 10.11.1.128 ( ) Status: Up
Host: 10.11.1.133 ( ) Status: Up
Host: 10.11.1.136 ( ) Status: Up
Host: 10.11.1.141 ( ) Status: Up
Host: 10.11.1.145 ( ) Status: Up

```

Рис. 3: Выводим полученный список "живых" машин.

Как видно, у нас теперь есть список живых машин, но он представлен не совсем в чистом виде. Попробуем написать маленький скрипчик, чтобы список содержал только айпишники.

```
root@kali:~# cat hosts.txt | grep "Status: Up" | cut -d " " -f2
10.11.1.5
10.11.1.7
10.11.1.8
10.11.1.10
10.11.1.13
10.11.1.14
10.11.1.22
10.11.1.24
10.11.1.31
10.11.1.35
10.11.1.39
10.11.1.44
```

Рис. 4: Получаем чистый список айпишников

Теперь этот список можно скармливать различным скриптам и программам. Попробуем получить названия машин в сети. Для этого напишем еще один простой bash-скрипт.

```
root@kali:~# for ip in $(cat clear_hosts.txt); do nslookup $ip; done
Server:                172.16.206.2
Address:               172.16.206.2#53

** server can't find 5.1.11.10.in-addr.arpa: NXDOMAIN

Server:                172.16.206.2
Address:               172.16.206.2#53

** server can't find 7.1.11.10.in-addr.arpa: NXDOMAIN

Server:                172.16.206.2
Address:               172.16.206.2#53

** server can't find 8.1.11.10.in-addr.arpa: NXDOMAIN

Server:                172.16.206.2
Address:               172.16.206.2#53

** server can't find 10.1.11.10.in-addr.arpa: NXDOMAIN

Server:                172.16.206.2
```

Рис. 5: Пробуем получить названия машин в сети.

Как видим, названия получить не удастся. Все дело в том, что названия машины в сети это информация типа CNAME, за которую отвечает DNS-сервер, а наш DNS-сервер из обычной сети понятия не имеет о том, что происходит в vpn.

Чтобы поправить ситуацию найдем конфиг, который отвечает за настройки DNS-сервера, который в данный момент