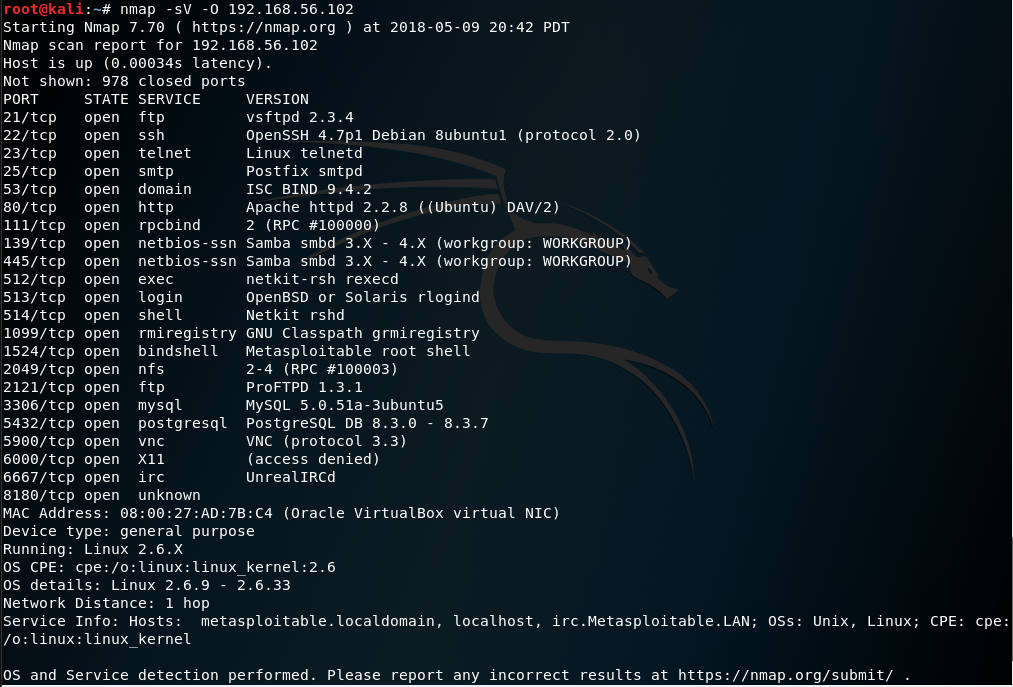
Patrick Austin

CS 445 Homework 4

9 May, 2018

1. In order to get the maximum amount of information, service/version detection and OS detection were conducted with nmap -sV -O 192.168.56.102, with the following results:

This command retrieves the operating system and OS version of the metasploitable VM being scanned, detects open ports, and attempts to determine the version of the services being run on the open ports. This information is highly useful when attempting to use Metasploit exploits on the Metasploitable VM.

2. For each module specified, I attempt to describe what the module does, the targeted port, and how the module works in high level language.

2.1. **exploit/windows/smb/ms09\_050\_smb2\_negotiate\_func\_index** exploits an out-of-bounds function table dereference in the SMB, or server message block, request validation code for Windows Vista SP1/SP2 and Windows Server 2008. The module sends an SMB request that causes this vulnerability. The port targeted is 445. Exploiting this out-of-bounds vulnerability can be used to allow remote execution of arbitrary code, which is a massive security vulnerability.

2.2. **exploit/multi/samba/usermap\_script** exploits the Samba implementation of SMB/CIFS networking protocol by specifying a username with shell meta characters in Samba 3.0.20 through 3.0.25rc3. The port targeted is 139. Specifying a username in this fashion via the username map script option actually allows the execution of arbitrary commands on the target machine.

2.3. **auxiliary/gather/mongodb\_js\_inject\_collection\_enum** does a NoSQL injection on the MongoDB platform for versions less than 2.4. The port targeted is 80, and a target vulnerable URI must be specified. A NoSQL injection attack aims at a NoSQL input that is not sanitized to check against inputs containing NoSQL statements; if the input is not sanitized then this attack allows the attacker to enumerate the collections available in the target data.

2.4. **auxiliary/dos/ssl/dtls\_fragment\_overflow** causes a denial of service attack targeting the Datagram TLS platform in OpenSSL for various versions, including all before 0.9.8za, 1.0.0 before version 1.0.0m, and 1.0.1. Before version 1.0.1h. The port targeted is 4433. The denial of service is caused by sending a fragmented message where later fragments have a larger size than the first fragment; this overflows the space in memory allocated to the later fragments and causes buffer overflow, which denies service by causing a segmentation fault crash.

2.5. **exploit/windows/scada/yokogawa\_bkfsim\_vhfd** exploits a stack-based buffer overflow attack specific to the Yokogawa CS3000 system, which is used for process control in large-scale manufacturing plants. The port targeted is 20010. By causing the stack-based buffer overflow, this attack allows for the remote execution of arbitrary code via crafted packets.

2.6. **post/firefox/manage/webcam\_chat** allows a user running a privileged Firefox Javascript shell to do a P2P stream of the webcam of the associated device without any visible prompting to the user or device. Essentially a “silent” ICE P2P chat is created which can be set up to never prompt the user or display a chat window, but it can still be used to stream webcam information. The ICE server communicates on port 80.

2.7. **exploit/android/local/futex\_requeue** exploits a bug in the futex\_requeue function of the Linux kernel used by Android devices built before June 2014. This exploit does not involve networking and runs on a local device, so no port is applicable. The futex\_requeue function does not check that calls have different futex (used for locking system resources) addresses, which allows a user to gain inappropriate privileges in the system that can be used for total integrity/confidentiality/availability compromise.

2.8. **exploit/unix/misc/distcc\_exec** exploits any system running the DistCC platform, which is used for distributed compiling of code on a network to take advantage of idle processing on other devices. Port 3632, used by DistCC, is targeted. Remote attackers can execute arbitrary code by sending DistCC compilation jobs to the targeted device; DistCC simply executes this code without any authorization check.

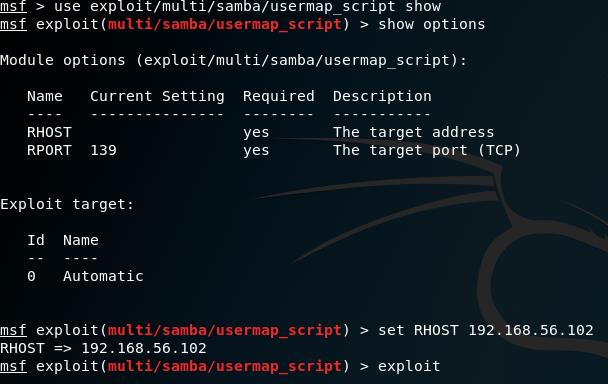
2.9. **auxiliary/admin/scada/advantech\_webaccess\_dbvisitor\_sqli** allows a remote user to target a device running Advantech WebAccess 7.1. One of the component .dll files has a SQL injection vulnerability that allows malicious requests to the web service to extract usernames and hashes. The targeted port is TCP port 80. By injecting a malicious SQL request in a unsanitized SQL field, unintended data can be accessed- in this case passwords and hashes, which could then be used to compromise login for weak passwords via eg a rainbow table attack.

3. Many of the attacks described above fail to target the Metasploitable2 VM simply because Metasploitable is built on the Unix/Ubuntu platform, where attacks targeting e.g. Windows or Android are not applicable. Also, several of the targeted services are not run by Metasploitable2, such as MongoDB. There was no webcam on the tested device for the Firefox exploit, and so on.

The working exploits on the Metasploitable2 VM from the list above are **exploit/multi/samba/usermap\_script** and **exploit/unix/misc/distcc\_exec**. Below I show the commands used to execute each exploit on the Kali device and the resulting remote access to the Metasploitable2 shell achieved.

**exploit/multi/samba/usermap\_script:**

First I configured the exploit to target the IP 192.168.56.102, which was the address of the Metasploitable2 VM. Then I executed the exploit. This is shown in the screenshot below.

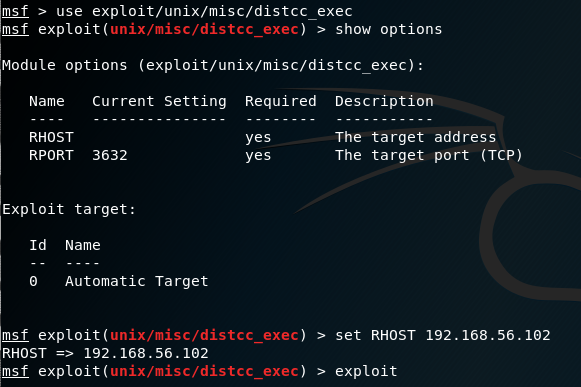


Next I show the execution of the exploit and the results. The below screenshot shows execution results, and then the commands I was able to remotely execute on the Metasploitable2 VM. The ifconfig command confirms that I was getting results from Metasploitable2 at 192.168.56.102, not the Kali device which was located at 192.168.56.103.



**exploit/unix/misc/distcc\_exec:**

As before, I first configured the exploit to target the Metasploitable2 VM at 192.168.56.102.



Then I executed the exploit and demonstrated that I had access to the Metasploitable2 shell, located at 192.168.56.102, even though I was actually using the Kali device at 192.168.56.103.



4. The major security flaws of the Metasploitable2 VM relate to configuration issues, the open ports of the device, and the services running on those ports.

An initial problem with the Metasploitable2 machine is that it has been deliberately misconfigured, with several services simply allowing remote access from any host when they should not. Obviously correct configuration of the services to enforce user authentication and limit access as much as possible to appropriate users would be the more secure course of action. Also, where authentication is used for database and server accounts, the Metasploitable VM uses extremely weak, common, or default username/password pairs. Some examples of logins used in Metasploitable2 include msfadmin/msfadmin, user/user, service/service, klog/123456789, and etc. that can be easily guessed by the attacker. Strong passwords should be used as a matter of course.

Clearly, as seen in part 1, the Metasploitable2 machine runs a wide variety of services in an open, obvious fashion; nmap detected no filtered ports, for example. A more secure system should run as few services as possible for the server to function as the organization requires, and employ port filtering via firewall as much as possible in order to protect the few necessary services that are running from any known vulnerabilities or undesired use. This filtering would also allow firewall rules to be put into place that could detect and filter incoming packets crafted to target known exploits, such as the ones used in part 3.

Moreover, the services that are running on the Metasploitable2 machine are out-of-date, vulnerable versions for which known exploits and backdoors exist. By keeping services updated as much as possible many of these exploits could be avoided even with the port staying open. For example, simply updating Samba to the most recent version would have prevented us from using the exploit/multi/samba/usermap\_script exploit in part 3, without any need for filtering to protect the port (though that should be done as well).

In summary, a secure system should be correctly configured and use strong passwords, run as few services as possible with robust firewall rules on the listening ports, and the services running should be updated to the most recent version as quickly as possible to avoid vulnerability to exploits. The Metasploitable2 machine deliberately fails to accomplish any of these goals, making it extremely vulnerable to attack.