Ex060

Joshua Main-Smith

2020-10-08

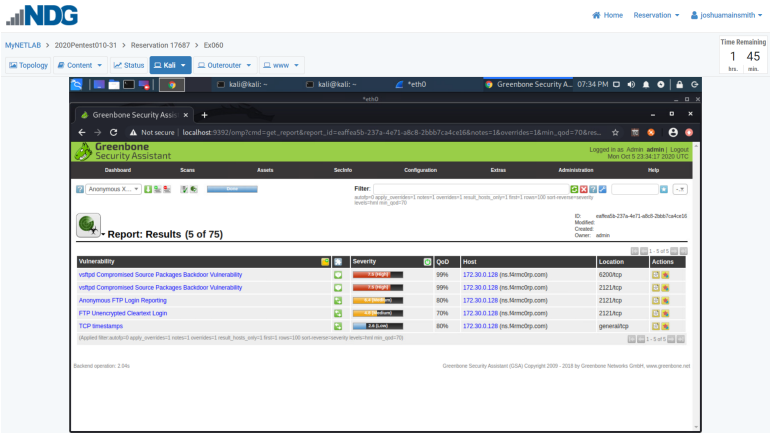
Contents

Technical Report

Risk Assessment

Vulnerability Description

Using OpenVAS to scan for any vulnerabilities on www.f4rmc0rp.com (172.30.0.128), we found that there are two high, two medium, and one low ranking vulnerabilities. A screenshot of the vulnerabilities found can be seen below.



The first two high ranking vulnerabilities are the same vulnerability located on different ports (2121 and 6200). This vulnerability allows an attacker to execute arbitrary code in a backdoor attack. Originally, TCP port 6200 was seen to be closed when running an nmap scan. But, after applying known attack credentials associated with a known vulnerability regarding vsptd 2.3.4 on port 2121, port 6200 was toggled open. Netcat can then be used to connect to the new listening port. This can be mitigated by updating to the most recent version of vsftpd.

The Anonymous FTP Login Reporting [Medium] can be used to connect to the ftp port on 2121 using the anonymous login credentials. This can be mitigated by not allowing anonymous logins.

The FTP Unencrypted Cleartext Login [Medium] allows attackers to sniff traffic between user and host in recovering login credentials because the traffic is in cleartext. This can be mitigated by encrypting all traffic between user and host.

The TCP Timestamps [Low] vulnerability shows uptime of a service, allowing attackers to potentially corrupt remote hosts during uptime. This can be mitigated by not enabling tcp timestamps.

Attack Narrative

OpenVAS Vulnerability Scan

The first two high ranking vulnerabilities are the same vulnerability located on different ports. This vulnerability allowed us to setup a backdoor on port 6200 and to access the contents on the server. There were two ways we were able to achieve this.

Accessing Server Contents via Metasploit

Our research indicated that the service running on port 2121, vsftpd 2.3.4, is vulnerable to a backdoor execution attack. The corresponding Metasploit module for this execution is **exploit/unix/ftp/vsftpd_234_backdoor**. After starting Metasploit on our attack box (**msfconsole**), we ran the command **use exploit/unix/ftp/vsftpd_234_backdoor**. Then, using the **show options** command we needed to specify a remote host (**set RHOST 172.30.0.128**) to exploit and change the default port from **21** to **2121** (**set RPORT 2121**). After these are set up, we can simply run the command **exploit**. Before doing this, we opened Wireshark to observe the interaction between our attack box and the remote host. Observing the flow stream of the packets between remote and local host was an attempt to connect to port 2121 with the credentials **USER dw:)** and **PASS Tit**. Running the exploit several times along with Wireshark, we observed a variety of usernames and passwords used. The commonality between all of these iterations is the username consistently ended with **:)**. According to the backdoor code, the username is a string of alphanumeric characters ending with **:)**. The password is simply random alphanumeric characters.

```
sock.put("USER #{rand_text_alphanumeric(rand(6)+1)}:)\r\n")
resp = sock.get_once(-1, 30).to_s
print_status("USER: #{resp.strip}")
```

```
sock.put("PASS #{rand_text_alphanumeric(rand(6)+1)}\r\n")
```

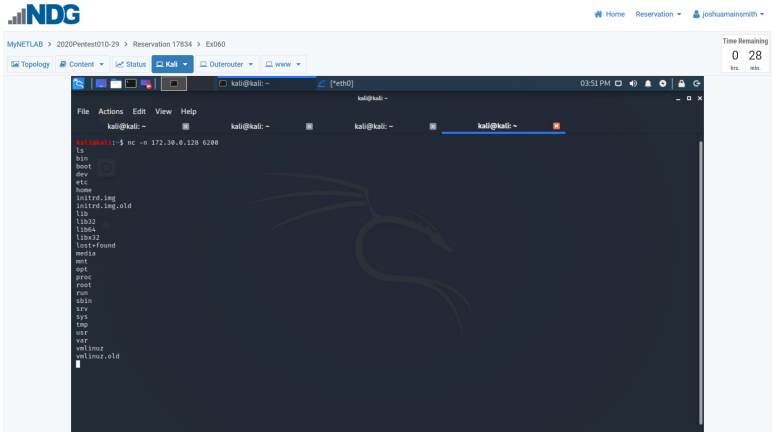
This indicates that the username and password required are irrelevant. The only requirement being that the username must end with **:)**. The exploit then attempts to connect to port 6200 without reading what the response from the password is, as indicated by the code below.

```
# Do not bother reading the response from password, just try the backdoor
nsock = self.connect(false, {'RPORT' => 6200}) rescue nil
if nsock
    print_good("Backdoor service has been spawned, handling...")
    handle_backdoor(nsock)
    return
end
```

This works because using **:)** at the end of a username toggles port 6200 to open, which we can then connect to. This can also be done manually using Netcat.

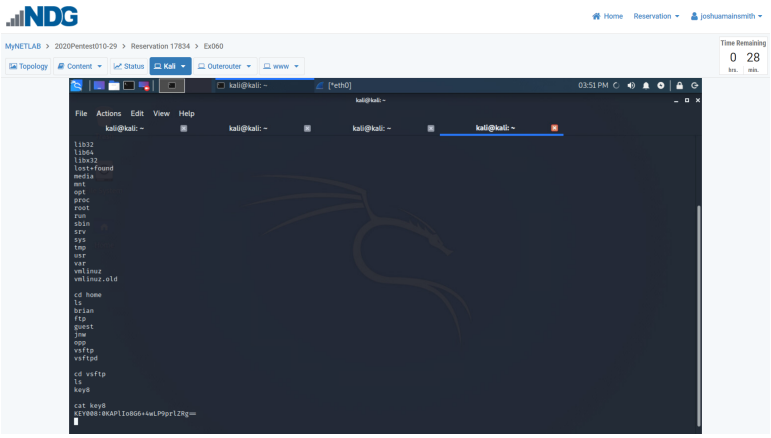
Accessing Server Contents via Netcat

Accessing manually is fairly straightforward, given our observation of the exploitation code above. Connecting to the ftp port on 2121, we issue the command **ftp -p 172.30.0.128 2121**. Here, we were asked to give a username and password. We gave a random alphanumeric char/string followed by a **:)**. The password is likewise a random alphanumeric char/string. The connection stalled at this point (it can be observed using an nmap scan on tcp port 6200 before the attempted ftp connection is closed, then opens after the attempted connection). We then connected using Netcat by issuing the command **nc -vvn 172.30.0.128 6200** (where **-vv** is double verbose and **-n** is a numeric address and **6200** is the port we're connecting to). The Netcat command and evidence that we were able to gain access can be seen in the image below.



KEY008

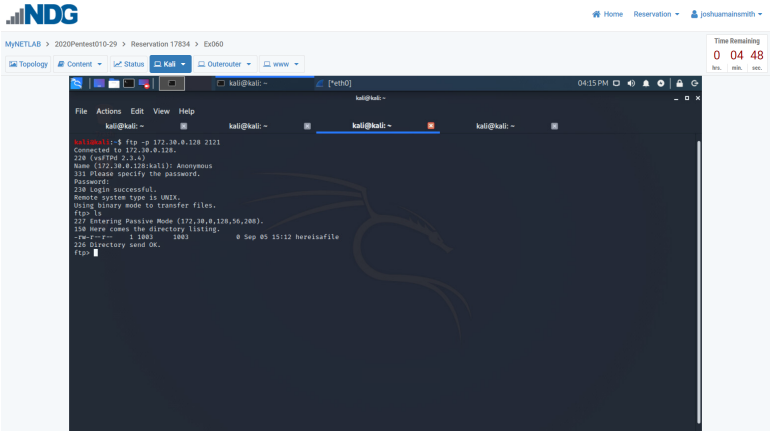
While observing the different files, we noticed something interesting while exploring the **/home** directory. Under **/home/vsftp** we found what appears to be a key, included in the image below.



The value of this key is **KEY008:0KAPlIo8G6+4wLP9prlZRg==**.

Anonymous FTP Login

Something else worth mentioning is, in exploiting the Anonymous FTP Login vulnerability we were able to connect via the tcp port 2121 by issuing the command **ftp -p 172.30.0.128 2121** and using the credentials **Anonymous** for the username and providing no password (just pressing **Enter**). Here, we were able to observe and download a file named **hereisafile**, the contents of which was empty. A screenshot of this can be seen below.



Zoom Link

October 5, 2020