Service Identification a11y.text Service Identification Scanning Services Using Metasploit a11y.text Scanning Services Using Metasploit Again, other than using Nmap to perform scanning for services on our target network, Metasploit also includes a large variety of scanners for various services, often helping you determine potentially vulnerable running services on target machines. SSH Service a11y.text SSH Service A previous scan shows us we have TCP port 22 open on two machines. SSH is very secure but vulnerabilities are not unheard of and it always pays to gather as much information as possible from your targets. msf > services -p 22 -c name,port,proto

Services == == == == host name port proto

----   ---- ---- ----- 172.16 .194.163 ssh 22 tcp 172.16 .194.172 ssh 22 tcp Weâ€™ll load up the ssh_version auxiliary scanner and issue the set command to set the â€˜RHOSTSâ€™ option. From there we can run the module by simple typing run . msf > use auxiliary/scanner/ssh/ssh_version

```
msf  auxiliary(ssh_version) > set RHOSTS 172.16.194.163 172.16.194.172
RHOSTS => 172.16.194.163 172.16.194.172

msf  auxiliary(ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):

  Name     Current Setting                Required  Description
  ----     ---------------                --------  -----------
  RHOSTS   172.16.194.163 172.16.194.172  yes       The target address range or CIDR identifier
  RPORT    22                             yes       The target port
  THREADS  1                              yes       The number of concurrent threads
  TIMEOUT  30                             yes       Timeout for the SSH probe
```

```
msf  auxiliary(ssh_version) > run

[*] 172.16.194.163:22, SSH server version: SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7
[*] Scanned 1 of 2 hosts (050% complete)
[*] 172.16.194.172:22, SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
```

FTP Service a11y.text FTP Service Poorly configured FTP servers can frequently be the foothold you need in order to gain access to an entire network so it always pays off to check to see if anonymous access is allowed whenever you encounter an open FTP port which is usually on TCP port 21. Weâ€™ll set the â€˜THREADSâ€™ to â€˜1â€™ here as weâ€™re only going to scan 1 host.

```
msf > services -p 21 -c name,proto

Services
========

host            name  proto
----            ----  -----
172.16.194.172  ftp   tcp

msf > use auxiliary/scanner/ftp/ftp_version

msf  auxiliary(ftp_version) > set RHOSTS 172.16.194.172
RHOSTS => 172.16.194.172
```

```
msf  auxiliary(anonymous) > show options
Module options (auxiliary/scanner/ftp/anonymous):


   Name     Current Setting      Required  Description
   ----     ---------------      --------  -----------
   FTPPASS  mozilla@example.com  no        The password for the specified username
   FTPUSER  anonymous            no        The username to authenticate as
   RHOSTS   172.16.194.172       yes       The target address range or CIDR identifier
   RPORT    21                   yes       The target port
   THREADS  1                    yes       The number of concurrent threads


msf  auxiliary(anonymous) > run


[*] 172.16.194.172:21 Anonymous READ (220 (vsFTPd 2.3.4))
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

In a short amount of time and with very little work, we are able to acquire a great deal of information about the hosts residing on our network thus providing us with a much better picture of what we are facing when conducting our penetration test. There are obviously too many scanners for us to show case. It is clear however the Metasploit Framework is well suited for all your scanning and identification needs.

```
msf > use auxiliary/scanner/
Display all 485 possibilities? (y or n)

...snip...
```

Next Password Sniffing Prev Hunting for MSSQL