Creating workspaces to organize your attack

First, we need to set up a workspace. Workspaces are a big help in keeping your testing in order. The workspaces hold all your collected data of the test, including any login credentials that are collected and any system data collected during an exploit. It's best to keep your testing data separate so you can compare the results of a previous test later. We're going to set up a project called `TestCompany-int-20150402`. This is a way to name projects, with `<client-name>-[ int (internal) | ext (external) ]-<start-date (unix-style)>` This will help you 6 months down the road to remember which test is what.

To create a new project type:

```
workspace -a TestCompany-int-20150402
```

To enter the workspace type:

```
workspace TestCompany-int-20150402
```

```
msf > workspace -h
Usage:
    workspace                  List workspaces
    workspace [name]           Switch workspace
    workspace -a [name] ...    Add workspace(s)
    workspace -d [name] ...    Delete workspace(s)
    workspace -r <old> <new>   Rename workspace
    workspace -h               Show this help information

msf > workspace -a TestCompany-int-20150402
[*] Added workspace: TestCompany-int-20150402
msf > workspace TestCompany-int-20150402
[*] Workspace: TestCompany-int-20150402
msf > workspace
  default
  kalibook-int-20150300
* TestCompany-int-20150402
msf >
```

Notice that after entering the workspace and typing the workspace command again, the asterisk has moved the `TestCompany` project. The asterisk shows the working workspace.

We can pull data from a scan into the workspace using the `db_import` command from an XML file generated by the scanning application. All scanning applications will export their data to xml and Metasploit will automatically import the data from the major scanning applications.

```
msf > cd kalibook/scans-docs   Changing directory to the scans
msf > ls
[*] exec: ls

201503150408 Intense scan, no ping on 192.168.202.0_24.xml
lab1-report.xml
openvas-vul-scan.xml
report-b82a186a-9b82-41e6-9b30-38b1c0d38ad9.pdf
msf > db_import openvas-vul-scan.xml   Importing scan data into the database
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.6.6.2'
[*] Importing host 192.168.202.1
[*] Importing host 192.168.202.128
[*] Importing host 192.168.202.130
[*] Importing host 192.168.202.131
[*] Successfully imported /root/kalibook/scans-docs/openvas-vul-scan.xml
msf >
```

You can also import hosts, services, and network information using Nmap and directly import Nmap's output into Metasploit using the `msfconsole's db_nmap` command. This command works with all the normal `nmap` command-line flags. The `db_` informs Metasploit to import the data. Running just `nmap` will run the scan but no data will be imported into Metasploit; you will just see the output of the command.

We have run the command:

```
db_nmap -A -sV -O 192.168.202.0/24
```

The `-A` tells `nmap` to run all tests. The `-sV` tells `nmap` to record the versioning of any running services. The `-O` tells `nmap` to record the operating system of any running hosts. We will see the output of the running scan; however, this data is also collected in the database. Then, we can also see the results after importing by running the `hosts` and `services` commands.

```
msf > db_nmap -A -sV -O 192.168.202.0/24
[*] Nmap: Starting Nmap 6.47 ( http://nmap.org ) at 2015-05-02 17:54 EDT
[*] Nmap: Nmap scan report for 192.168.202.1
[*] Nmap: Host is up (0.00012s latency).
[*] Nmap: Not shown: 996 closed ports
[*] Nmap: PORT    STATE SERVICE         VERSION
[*] Nmap: 22/tcp  open  ssh             (protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   1024 8a:9b:c3:89:a3:5d:d8:04:67:76:a2:1b:a4:a8:55:db (DSA)
[*] Nmap: |   2048 ae:9e:00:2a:6e:93:e1:4d:59:d8:5a:96:b0:03:53:06 (RSA)
[*] Nmap: |_  256 b7:d3:80:c1:b2:3f:5f:5b:48:c8:13:0e:9f:4e:73:eb (ECDSA)
[*] Nmap: 111/tcp open  rpcbind         2-4 (RPC #100000)
[*] Nmap: | rpcinfo:
[*] Nmap: |   program version   port/proto  service
[*] Nmap: |   100000  2,3,4        111/tcp  rpcbind
[*] Nmap: |   100000  2,3,4        111/udp  rpcbind
[*] Nmap: |   100024  1          32927/udp  status
[*] Nmap: |_  100024  1          49336/tcp  status
[*] Nmap: 443/tcp open  ssl/http        VMware VirtualCenter Web service
[*] Nmap: |_http-methods: No Allow or Public header in OPTIONS response (status code 501)
[*] Nmap: |_http-title: Site doesn't have a title (text; charset=plain).
[*] Nmap: | ssl-cert: Subject: commonName=VMware/countryName=US
[*] Nmap: | Not valid before: 2015-02-28T06:34:52+00:00
[*] Nmap: |_Not valid after:  2016-02-28T06:34:52+00:00
[*] Nmap: 902/tcp open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
[*] Nmap: 1 service unrecognized despite returning data. If you know the service/version, please sub
mit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
[*] Nmap: SF-Port22-TCP:V=6.47%I=7%D=5/2%Time=554547DB%P=x86_64-unknown-linux-gnu%r(
[*] Nmap: SF:NULL,29,"SSH-2\.0-OpenSSH_6\.6\.1p1\x20Ubuntu-2ubuntu2\r\n");
[*] Nmap: MAC Address: 00:50:56:C0:00:01 (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 3.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:3
[*] Nmap: OS details: Linux 3.11 - 3.14
[*] Nmap: Network Distance: 1 hop
[*] Nmap: TRACEROUTE
```

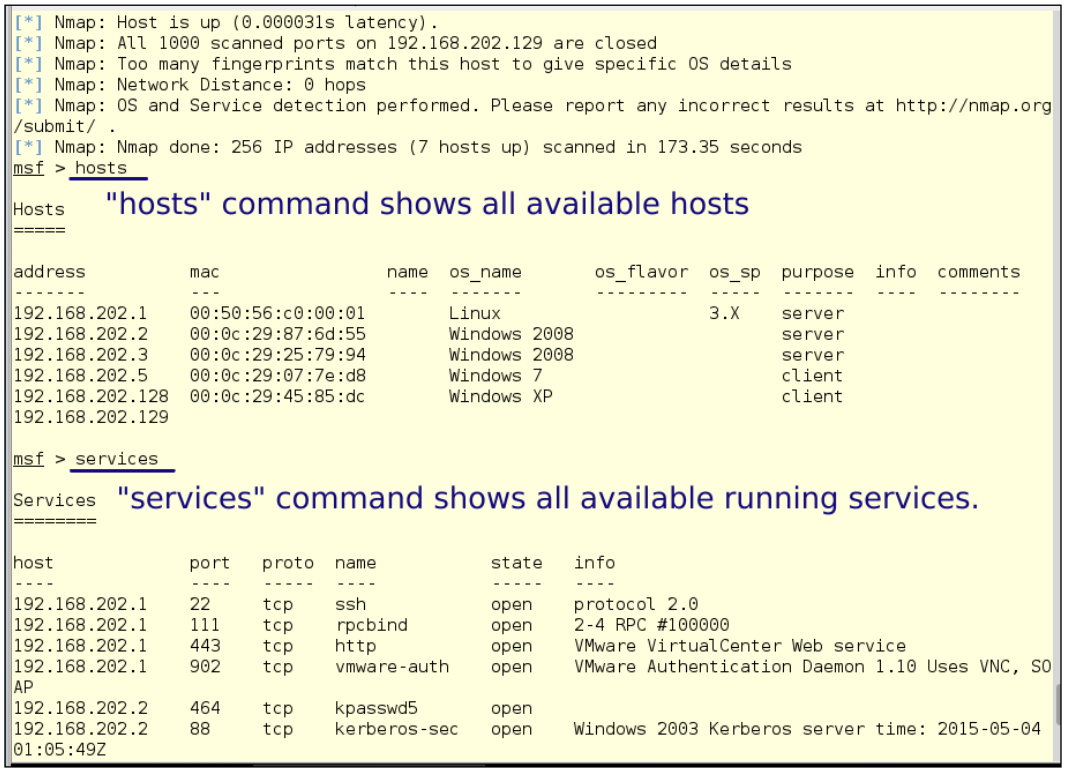
Using the hosts and services commands

Next, we see the results of running the following commands:

`hosts`

`services`

With the `hosts` command, we get a list of all active IP addresses, any collected machine names, and the operating system of the machine. By running the `services` command, we get a list of all running services on the network and their related IP address. You can change the table listings from the command by using the `-c` flag. The help information for these commands is shown in the following screenshot.



Using advanced footprinting

Vulnerability scans only provide minimal information. When actually attacking the machine, you want to perform some deep level probes to check for helpful information leaks. From the scans, we can see that both a Windows Domain Controller and a Windows File Server run Windows 2008 Server. Both have SMB/NetBIOS services running. A good first attack vector in a case like this is to exploit the SMB/NetBIOS services, which are known to have exploitable weaknesses. So, let's look closer at these services.

SMB and Netbios

Both **SMB (Server Message Block)** and **NetBIOS (Network Basic Input/Output System)** are protocols used in Windows and Unix-based networks for **file sharing, remote access, and communication**. They often work together but serve different purposes.

1. SMB

SMB is a file-sharing protocol that allows computers to **share files, printers, and other resources** over a network. It is commonly used in **Windows networks** and implemented in **Samba** for Linux.

How SMB Works

- Clients request access to shared resources (files, printers).
- The server responds and provides access if the user has permissions.
- Authentication can be required to access shared resources.

smb ports

port	protocol	Description
445	SMB over TCP	Direct SMB connection without NetBIOS.
139	SMB over NetBIOS	Used when NetBIOS is enabled.

SMB Versions

SMB Version	Description	Vulnerabilities
SMBv1	Oldest version (Windows NT, XP)	Vulnerable to **EternalBlue (MS17-010), Wannacry, Petya**.
SMBv2	Improved security & performance (Windows Vista+)	Fewer known vulnerabilities.
SMBv3	Latest version (Windows 8, 10, 11)	Supports encryption and better security.

use ‘auxiliary/scanner/smb/smb_version’ to check smb version.

SMB Commands

List Shared Folders

smbclient -L //TARGET_IP -N

You may get error from the above command if anonymous access (guest login) is not allowed on the target SMB server.

Connect to a Share

smbclient //TARGET_IP/SHARE_NAME -U username

Scan for SMB Vulnerabilities

nmap --script smb-vuln* -p 445 TARGET_IP

---

2. NetBIOS

NetBIOS is an older protocol used for name resolution and communication between computers in a local network. It allows devices to be identified by their NetBIOS names instead of IP addresses.

NetBIOS Functions

- Name registration and resolution.
- Session communication (allowing applications to talk over a network).
- Datagram distribution (sending messages without a session).

NetBIOS Ports

port	protocol	Description
137	NetBIOS Name Service (NBNS)	Resolves NetBIOS names to IPs.
138	NetBIOS Datagram Service	Used when NetBIOS is enabled.Used for browser announcements.
139	NetBIOS Session Service	Supports SMB traffic over NetBIOS.

NetBIOS Commands

Scan for NetBIOS Names

nbtscan -v TARGET_IP/24

List NetBIOS Information

nmblookup -A TARGET_IP

Check NetBIOS Sessions

netstat -an | grep 139

How SMB and NetBIOS Work Together

- Originally, SMB relied on **NetBIOS over TCP/IP** (port 139).
- **Modern SMB (Windows 2000+) runs directly over TCP/IP (port 445)** without NetBIOS.
- Some **older networks still use NetBIOS for name resolution**.

Below is the output of 'services' command after 'db_nmap -A -sV 172.17.165.0/24'

```
msf6 > services
Services
========

host            port   proto  name            state  info
----            ----   -----  ----            -----  ----
172.17.165.17   80     tcp    http            open   Apache httpd 2.4.58 (Ubuntu)
172.17.165.41   135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.41   139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.41   445    tcp    microsoft-ds    open
172.17.165.44   7070   tcp    ssl/realserver  open
172.17.165.58   135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.58   139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.58   445    tcp    microsoft-ds    open
172.17.165.62   135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.62   139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.62   445    tcp    microsoft-ds    open
172.17.165.100  135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.100  139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.100  445    tcp    microsoft-ds    open   Microsoft Windows 7 - 10 microsoft-ds workgroup: WORKGROUP
172.17.165.100  5357   tcp    http            open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
172.17.165.100  49152  tcp    msrpc           open   Microsoft Windows RPC
172.17.165.100  49153  tcp    msrpc           open   Microsoft Windows RPC
172.17.165.100  49154  tcp    msrpc           open   Microsoft Windows RPC
172.17.165.100  49155  tcp    msrpc           open   Microsoft Windows RPC
172.17.165.120  135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.120  139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.120  445    tcp    microsoft-ds    open
172.17.165.171  135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.171  139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.171  445    tcp    microsoft-ds    open
172.17.165.173  139    tcp    netbios-ssn     open   Samba smbd 4.6.2
172.17.165.173  445    tcp    netbios-ssn     open   Samba smbd 4.6.2
172.17.165.173  7070   tcp    ssl/realserver  open
172.17.165.184  139    tcp    netbios-ssn     open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
172.17.165.184  445    tcp    netbios-ssn     open   Samba smbd 4.3.11-Ubuntu workgroup: WORKGROUP
172.17.165.222  135    tcp    msrpc           open   Microsoft Windows RPC
172.17.165.222  139    tcp    netbios-ssn     open   Microsoft Windows netbios-ssn
172.17.165.222  445    tcp    microsoft-ds    open
```

let us intercept this output

Observations:

- **Windows SMB Servers**: 172.17.165.41, 172.17.165.58, 172.17.165.62, 172.17.165.100, etc.
- **Samba (Linux SMB)**: 172.17.165.173 (Samba smbd 4.6.2), 172.17.165.184 (Samba smbd 4.3.11-Ubuntu).

IP Address	Port 139 (NetBIOS-SSN)	Port 445 (SMB - Microsoft-DS)	Description
172.17.165.41	✅ Open	✅ Open	Windows SMB
172.17.165.58	✅ Open	✅ Open	Windows SMB
172.17.165.62	✅ Open	✅ Open	Windows SMB
172.17.165.100	✅ Open	✅ Open	Windows SMB, Workgroup: `WORKGROUP`
172.17.165.120	✅ Open	✅ Open	Windows SMB
172.17.165.171	✅ Open	✅ Open	Windows SMB
172.17.165.173	✅ Open	✅ Open	**Samba 4.6.2 (Linux SMB)**
172.17.165.184	✅ Open	✅ Open	**Samba 4.3.11-Ubuntu, Workgroup: WORKGROUP**
172.17.165.222	✅ Open	✅ Open	Windows SMB

Now you find your targets which are using smb or netbios. Let us explore metasploit payloads on the target