



# Penetration Testing Professional

## ENUMERATION

### Section 2: Network Security – Module 3



3.1 Enumeration

3.2 NetBIOS

3.3 SNMP

3.4 Conclusions



## 3.1. Enumeration



3

# ENUMERATION

eLearnSecurity  
Forging security professionals



## 3.1. Enumeration



MAP



REF



VIDEO



LAB

4

After the scanning phase, the next phase we are going to dive into is known as **enumeration**.

The goal is to gather more detailed information on both devices and resources attached to the network. This includes account names, shares, misconfigured services and so on.

Like the scanning phase, enumeration involves active connections to the remote devices in the network.

LearnSecurity  
Forging security professionals



## 3.1. Enumeration



5

There are many protocols on networked systems that one can easily exploit if administrators do not take the necessary steps to either secure the protocols or, disable them.

In this module, we will see ways of identifying these protocols, with the intent to eventually learn how to exploit them in later phases.





## 3.1. Enumeration



For example, NetBIOS (**N**etwork **B**asic **I**nput **O**utput **S**ystem) is the service that allows Windows systems to share files, folders, and printers among machines on a LAN. If not properly configured, it can lead to large amount of information leakage.

NetBIOS can be extremely useful in determining types of system information such as user IDs and open shares.

LearnSecurity  
Forging security professionals



## 3.1. Enumeration



7

In addition to NetBIOS, a protocol that we will explore in this module is SNMP (Simple Network Management Protocol). It is a protocol used to both gather information and configure network devices (printers, switches, servers...).

It is worth noting that these are not the only protocols that you may find in a network therefore, the techniques that we will see in later sections may depend on the platforms in use.

ClearSecurity  
Forging security professionals



## 3.2. NetBIOS



# NETBIOS

eLearnSecurity  
Forging security professionals





## 3.2.1. What is NetBIOS?



Before seeing what techniques and tools we can use to gather information from NetBIOS, it is important to understand how NetBIOS and NetBIOS over TCP (NBT) work.

The very first version of NetBIOS was developed late in 1983. It was designed as an API (not as a protocol as many suspect) that served its purpose in developing client/server applications.

eLearnSecurity  
Forging security professionals



## 3.2.1. What is NetBIOS?



Since this old version was not intended to be encapsulated within TCP and UDP packets, in 1987 a new version was released: NetBIOS over TCP/IP (NetBT or NBT).

This new version of NetBIOS, developed to work where the TCP/IP protocol suite is available, is considered a true protocol and it is described in the following two RFCs: [1001](https://tools.ietf.org/html/rfc1001) and [1002](https://tools.ietf.org/html/rfc1002).

In the coming slides we will focus our tests on this protocol.

<https://tools.ietf.org/html/rfc1001>

<https://tools.ietf.org/html/rfc1002>



### 3.2.1. What is NetBIOS?



The main purpose of NetBIOS is to allow applications on different systems to communicate with one another over the LAN. It is used for a multitude of purposes including: sharing printers and files, remote procedure calls, exchange messages and much more. As expected, these features may reveal additional information such as computer names, user names, domains, printers, available shares...

*eLearnITSecurity*  
Forging security professionals

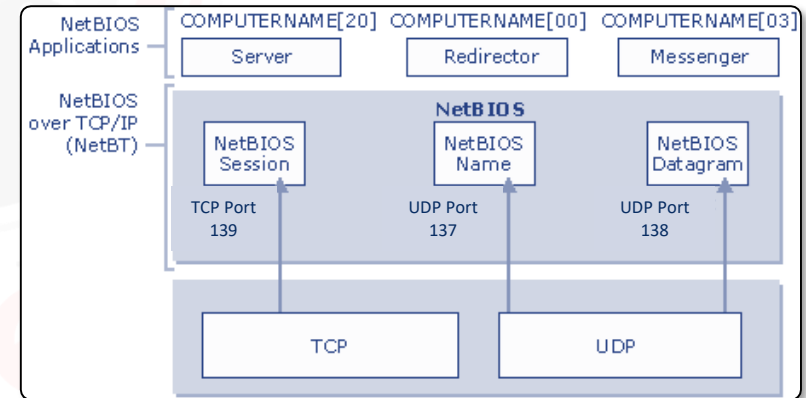


## 3.2.2. How NetBIOS works



PC's on a NetBIOS LAN communicate either by establishing a session or by using datagrams. To do this, NetBIOS uses the following TCP and UDP ports:

- UDP 137 for **name services**
- UDP 138 for **datagram services**
- TCP 139 for **session services**



<https://technet.microsoft.com/en-us/library/bb962072.aspx>



### Name service

The name service has the same purpose of a DNS record, it translates and maps a NetBIOS name to an IP address.

A name is an unique 16-byte address that identifies a NetBIOS resource on the network and is dynamically registered when either services or applications start. Names can be registered as unique names or as group names.

You can find more information about NetBIOS name resolution [here](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738412(v=ws.10)).

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738412\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738412(v=ws.10))



### Name service

In order to locate a resource, a NetBIOS Name Query is used to resolve the NetBIOS name to an IP address. A name is composed of 16 characters: the firsts 15 characters can be specified by the user, while the 16<sup>th</sup> character is used to indicate the resource type and goes from 00 to FF (hexadecimal).

The next table shows some of the names and types available.



## 3.2.2. How NetBIOS works



### Name service

Name	Service / Type	Name	Service / Type
[computer_name]00	Workstation Service	[user_name]03	Messenger Service
[computer_name]03	Messenger Service	[domain_name]1D	Master Browser
[computer_name]06	RAS Server Service	[domain_name]1B	Domain Master Browser
[computer_name]1F	NetDDE Service	[domain_name]00	Domain Name
[computer_name]20	Server Service	[domain_name]1C	Domain Control
[computer_name]21	RAS Client Service	[domain_name]1E	Broser Service Elections
[computer_name]BE	Network Monitor Agent	__MSBROWSE__	Master Browser
[computer_name]BF	Network Monitor Application		

More suffixes can be found [here](#).

<https://msdn.microsoft.com/en-us/library/cc224454.aspx>



## 3.2.2. How NetBIOS works



### Name service

The screenshot below shows the NetBIOS names on our machine. To see this information we can run the following command:

```
nbtstat -n
```

```
Wireless Network Connection:  
Node IpAddress: [192.168.0.14] Scope Id: []
```

#### NetBIOS Local Name Table

Name	Type	Status
LITSNARF-NB-PC <00>	UNIQUE	Registered
WORKGROUP <00>	GROUP	Registered
LITSNARF-NB-PC <20>	UNIQUE	Registered
WORKGROUP <1E>	GROUP	Registered
WORKGROUP <1D>	UNIQUE	Registered
.._MSBROWSE_. <01>	GROUP	Registered





### Name service

The service that actually maps NetBIOS names to IP address is called [Windows Internet Name Service](#) (WINS).

If you want to dig deeper into WINS, here are some valid Microsoft resources that you can leverage:

- [WINS Overview](#)
- [What is WINS](#)
- [WINS defined](#)

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc725802\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc725802(v=ws.11))

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784180\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784180(v=ws.10))

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784707\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784707(v=ws.10))



### Datagram service

The NetBIOS Datagram Service (NBDS) permits the sending of messages to a NetBIOS name. It runs on UDP port 138, therefore making it a connectionless communication.

The datagram service allows the sending and receiving of the datagram messages to and from:

- a specific NetBIOS name
- broadcast the datagram to all NetBIOS names



### Datagram service

The datagram and broadcast methods allow one computer to communicate with several other computers at the same time however, these communications are limited in terms of message size. There is no error detection / correction using the datagram or broadcast methods however, datagram communication allows communications without the need for a session to be established.

Forging security professionals



### Session service

The NetBIOS Session Service (NBSS) is the most commonly known of the NetBIOS services. It allows two names to establish a connection in order to exchange data.

For example, when a device creates a file sharing connection the session service is used. Once the session has been established, the two workstations use the *Server Message Block* (SMB) protocol, which we will explore later on.

Forging security professionals



### Session service

The following steps are used to establish the connection:

1. The NetBIOS name is resolved into an IP address
2. A TCP connection is established between the two devices, using the TCP port 139
3. The device starting the connection sends a NetBIOS Session Request over the TCP connection
  1. This includes the NetBIOS name of the application that wants to establish the connection and the NetBIOS name to which to connect
4. If the remote device is listening on that name, there will be a positive response and the session will be established.



Before seeing how to use NetBIOS, there is another popular protocol that we have to understand: [Sever Message Block](#). SMB lets you share files, disks, directories, printers and, in some cases, even COM ports across a network.

Before Windows 2000, SMB ran only with NetBIOS over TCP/IP (port 139), therefore a NetBIOS Session was required.



Windows 2000 and higher allow us to run [SMB directly over TCP/IP](#) (direct hosting), without the need to run over NetBIOS sessions. To do this, the TCP port 445 is used.

Since SMB provides several features such as manipulating files, sharing, messaging, Interprocess Communication (IPC) and more, it is one of the most attractive services to explore during our enumeration phase.



## 3.2.4. NetBIOS Commands and Tools



In the next few sections, we will explore different commands and tools that can be used to enumerate system information using NetBIOS. The one we will use the most is the famous `nbtstat`.

By using the data we gathered in the scanning phase, we can search for systems with ports 137/139/445 open.







Nbtstat is a tool developed to troubleshoot NetBIOS name resolution problems. The main options it offers are as follows:

-a	(adapter status)	Lists the remote machine's name table given its name
-A	(Adapter status)	Lists the remote machine's name table given its IP
-c	(cache)	Lists NBT's cache of remote [machine] names and their IP addresses
-n	(names)	Lists local NetBIOS names.
-r	(resolved)	Lists names resolved by broadcast and via WINS
-R	(Reload)	Purges and reloads the remote cache name table
-S	(Sessions)	Lists sessions table with the destination IP addresses
-s	(sessions)	Lists sessions table converting destination IP addresses to computer NETBIOS names.
-RR	(ReleaseRefresh)	Sends Name Release packets to WINS and starts Refr



Let's suppose that, during our scanning phase, we came across a machine (192.168.99.162) that has the following open ports:

```
Starting Nmap 6.49BETA5 ( https://nmap.org ) EST
Nmap scan report for 192.168.99.162
Host is up (0.085s latency).
Not shown: 1990 closed ports
PORT      STATE      SERVICE
135/tcp   open      msrpc
139/tcp   open      netbios-ssn
445/tcp   open      microsoft-ds
123/udp   open      ntp
137/udp   open      netbios-ns
```



Using this machine as target of our tests, we can open a Windows terminal and use `nbtstat -A` to start gathering information about it. Notice that on Windows systems, `nbtstat` is already installed.

The full command will look like the following:

```
C:\>nbtstat -A <target_IP_Address>
```



The following is the output we will obtain:

```
C:\>nbtstat -a 192.168.99.162

Local Area Connection 2:
Node IpAddress: [192.168.99.100] Scope Id: []

        NetBIOS Remote Machine Name Table

    Name                  Type               Status
    -----
    ELS-WINXP              <00>    UNIQUE         Registered
    WORKGROUP              <00>    GROUP           Registered
    ELS-WINXP              <20>    UNIQUE         Registered
    WORKGROUP              <1E>    GROUP           Registered
    WORKGROUP              <1D>    UNIQUE         Registered
    ._. _MSBROWSE_.        <01>    GROUP           Registered

    MAC Address = 00-50-56-B1-94-80
```

From here we can see that the computer name is *ELS-WINXP* and that the domain name is *WORKGROUP*.



A very important line in the previous output is:

ELS-WINXP	<20>	UNIQUE	Registered
-----------	------	--------	------------

If you recall from the table displayed earlier, the number <20> identifies a server service. This means that the host has file and printer shares enabled, therefore we may be able to access them.



If we are using Linux, there are others tools that allow us to obtain similar information. One of the most popular is `nbtscan`. The most basic command to run is the following:

```
nbtscan -v [target_IP_Address]
```

`-v` is used to set the verbosity of the output

NetBIOS Name Table for Host 192.168.99.162:

Name	Service	Type
ELS-WINXP	<00>	UNIQUE
WORKGROUP	<00>	GROUP
ELS-WINXP	<20>	UNIQUE
WORKGROUP	<1e>	GROUP
WORKGROUP	<1d>	UNIQUE
00000000-0000-0000-0000-00000000 MSBROWSE_00000000	<01>	GROUP

Adapter address: 00:50:56:b1:94:80



It is worth noting that `nbtscan` is also able to scan multiple addresses. For example we can instruct the tool to scan all the IP addresses in our target network, see below:

```
stduser@kalisana:~$ nbtscan -v 192.168.99.0/24
Doing NBT name scan for addresses from 192.168.99.0/24

192.168.99.0    Sendto failed: Permission denied
192.168.99.255 Sendto failed: Permission denied

NetBIOS Name Table for Host 192.168.99.162:

Name           Service      Type
-----
ELS-WINXP      <00>         UNIQUE
WORKGROUP      <00>         GROUP
ELS-WINXP      <20>         UNIQUE
WORKGROUP      <1e>         GROUP
WORKGROUP      <1d>         UNIQUE
[00] MSBROWSE [00] <01>         GROUP

Adapter address: 00:50:56:b1:32:3c
```

```
nbtscan -v 192.168.99.0/24
```



The tools we have seen up to this point operate using the NetBIOS Naming Service (NBNS). If we inspect the traffic with Wireshark we will see a few messages like these:

No.	Time	Source	Destination	Protocol	Length	Src Port	Dst Port	Info
1	0.000000000	192.168.99.102	192.168.99.162	NBNS	92	34669	137	Name query NBSTAT *<00><00><00><00><00>
4	0.166290000	192.168.99.162	192.168.99.102	NBNS	271	137	34669	Name query response NBSTAT

▶ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▶ Ethernet II, Src: 22:74:cd:64:08:54 (22:74:cd:64:08:54), Dst: Vmware_b1:94:80 (00:50:56:b1:94:80)
▶ Internet Protocol Version 4, Src: 192.168.99.102 (192.168.99.102), Dst: 192.168.99.162 (192.168.99.162)
▶ User Datagram Protocol, Src Port: 34669 (34669), Dst Port: 137 (137)
▼ NetBIOS Name Service
Transaction ID: 0x01b0
▶ Flags: 0x0010 (Name query)
Questions: 1
Answer RRs: 0





With this information, we can now move on and verify what the target machine is sharing over the network.

To do so we can use the Microsoft [net](#) command. The `net` command offers many features such as the ability to update user accounts, display, view and modify services, connect computers to shared resources and much more.

For our purposes we will focus on the [net view](#) command.



### 3.2.4.3. Net command



Net view allows us to list domains, computers and resources shared by a computer in the network. Let us see how to use it against our previous target. The command we are going to run is the following:

```
C:\>net view 192.168.99.162
```

```
C:\>net view 192.168.99.162  
Shared resources at 192.168.99.162
```

Share name	Type	Used as	Comment
------------	------	---------	---------

C	Disk		
Frank	Disk		
FrankDocs	Disk		
My Documents	Disk		
WorkSharing	Disk		

The command completed successfully.



### 3.2.4.3. Net command



As we can see from the previous screenshot, the command completes successfully and lists the resources shared by the target machine.

To explore these shares we can simply browse them by utilizing the [net use](#) command.



### 3.2.4.3. Net command



The `net use` command can be used to connect or disconnect a computer from a shared resource. This means that we can connect our computer to the remote shared folder in order to navigate the remote folders.

For example, if we want to start a connection on the C resource, we can use the following command:

```
net use K: \\192.168.99.162\C
```



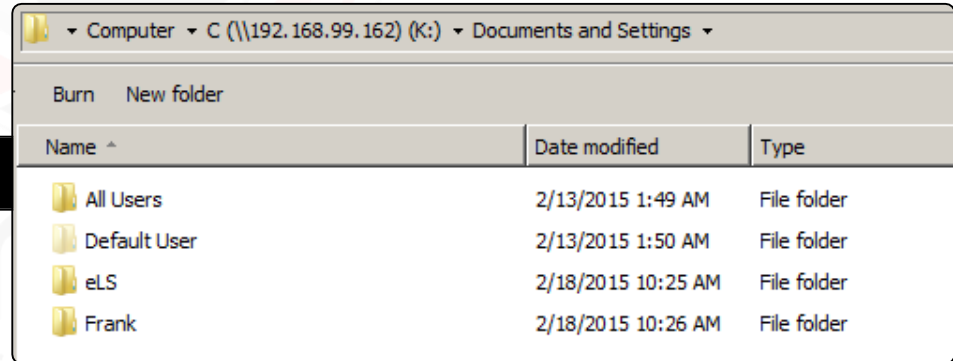
### 3.2.4.3. Net command



37

The command in the previous slide, will add the K: drive to our mapped drives thus, containing the files and folders from the victim C resource. We can now explore these files or attach others shares to see if the machine exposed useful information.

```
C:\>net use K: \\192.168.99.162\C
The command completed successfully.
```



The screenshot shows a Windows File Explorer window with the address bar set to 'Computer > C (\\192.168.99.162) (K:) > Documents and Settings'. The window displays a list of folders under the 'Name' column, with 'Date modified' and 'Type' columns also visible. The folders listed are 'All Users', 'Default User', 'eLS', and 'Frank', all of which are file folders. The 'Date modified' column shows dates ranging from 2/13/2015 to 2/18/2015. The 'Type' column indicates that all listed items are 'File folder'.

Name ^	Date modified	Type
All Users	2/13/2015 1:49 AM	File folder
Default User	2/13/2015 1:50 AM	File folder
eLS	2/18/2015 10:25 AM	File folder
Frank	2/18/2015 10:26 AM	File folder



If you are using a Linux machine, the same results can be obtained with different tools. If we want to list all the shares of a specific computer we can use `smbclient` as follows:

```
smbclient -L 192.168.99.162
```

```
stduser@kalisana:~$ sudo smbclient -L 192.168.99.162
Enter root's password:
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

  Sharename      Type            Comment
  -----
  My Documents   Disk
  IPC$           IPC             Remote IPC
  Frank          Disk
  C              Disk
  WorkSharing    Disk
  FrankDocs      Disk
  ADMIN$         Disk           Remote Admin
  C$             Disk           Default share
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```



### 3.2.4.4. Smbclient and mount



As you can see, `smbclient` also displays a few other hidden shares such as `IPC$`, `C$` and `ADMIN$`. You can identify hidden shares since they have a `$` sign at the end.

The three shares listed above are the default administrative shares and have their own specific purpose. For example `IPC$` is used for *Inter-Process Communications* and can be used to leverage *null session* attacks.



### 3.2.4.4. Smbclient and mount



If we wish to navigate the target shares, we can use `mount` as an alternative to `net use`. For example if we want to navigate the C share, we will run something similar to the following command.

```
sudo mount.cifs //192.168.99.162/C /media/K_share/ user=,pass=
```

After we run it, we will be able to browse the remote share:

```
stduser@kalisana:/media/K_share$ ls
AUTOEXEC.BAT  Documents and Settings  NTDETECT.COM  Program Files
boot.ini      IO.SYS                 ntldr          System Volume Information
CONFIG.SYS    MSDOS.SYS              pagefile.sys   WINDOWS
stduser@kalisana:/media/K_share$
```





We have seen thus far some examples of basic NetBIOS information that we can gather from Windows and Linux Operating Systems.

Now we will move on to other tools and commands that can also provide a great deal of additional information. We will also query the NetBIOS API and exploit null sessions.





## 3.2.5. Null Session



MAP



REF



VIDEO



LAB

42

Before digging into a tool or command, let's clarify what a **null session** is and how it works.

Null sessions are one of the oldest and most known attacks performed on Windows 2000 and Windows NT environments. Thanks to this weakness, malicious users are able to establish a connection to the victim in order to gather information such as shares, users, groups, registry keys and much more.

ClearSecurity  
Forging security professionals



Null sessions rely on Common Internet File System (CIFS) and Server Message Block (SMB) API, that return information even to an unauthenticated user.

In other words, a malicious user can establish a connection to a Windows system without providing any username or password. In order for the attack to work, the connection must be established to the administrative share named IPC (Inter Process Communication).

clear security  
Forging security professionals



The easiest way to test if a machine is vulnerable to null session is by running the `net` command. Note that in contrast to the connection we established in the previous slides, we are going to target the `IPC$` share instead:

```
C:\Windows\system32\cmd.exe  
  
C:\>net use \\192.168.99.162\IPC$ "" /u:""  
The command completed successfully.
```

As we can see in the screenshot, the command works. From this moment on, we have an active connection to our victim.



## 3.2.5. Null Session



```
net use \\192.168.99.162\IPC$ "" /u:""
```

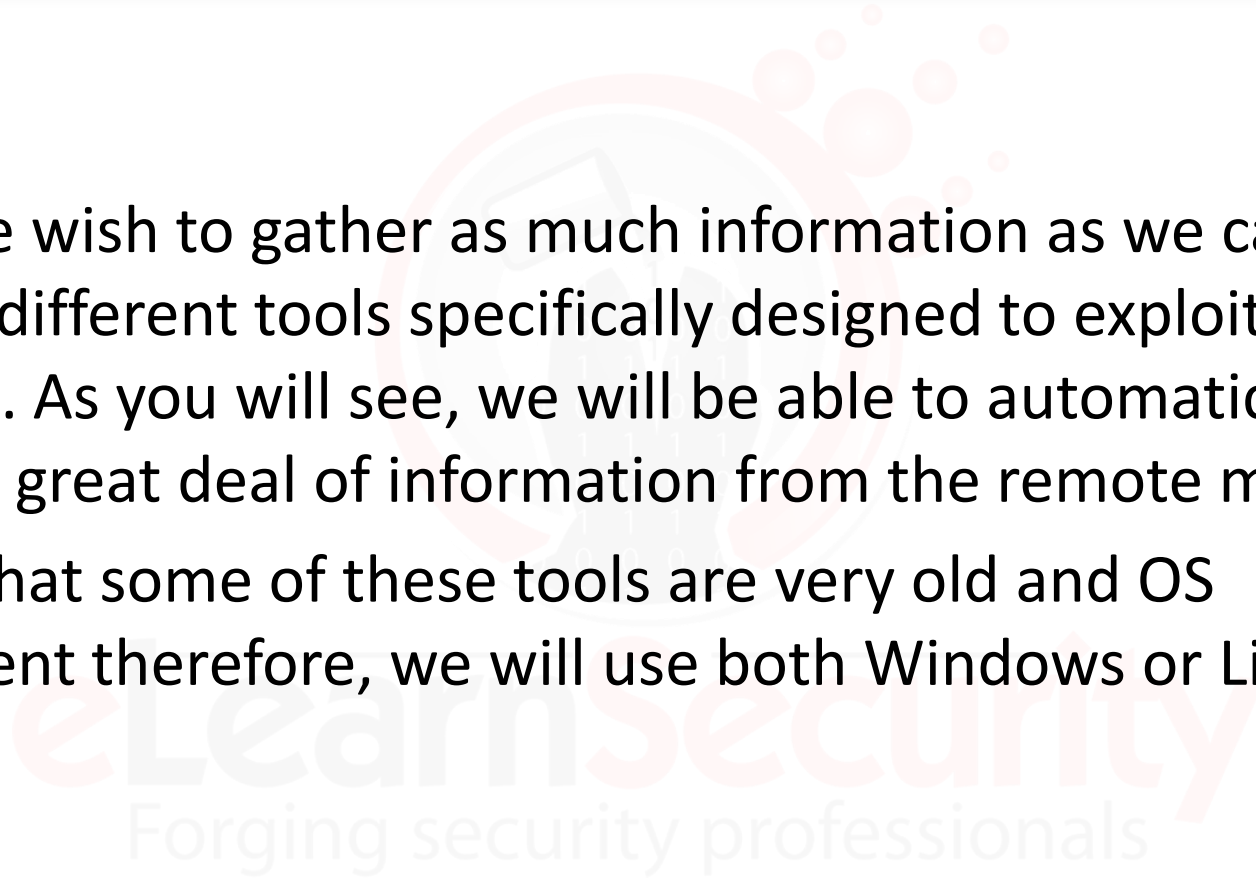
Let's explain the previous command. We are:

- establishing a connection to the hidden share **IPC\$**
- at the IP address **192.168.99.168**
- with a **null password**
- and an **empty (anonymous) username**.

Once we have a connection, we can use other tools to gather information from the remote machine.



Since we wish to gather as much information as we can, we will use different tools specifically designed to exploit null sessions. As you will see, we will be able to automatically gather a great deal of information from the remote machine. Notice that some of these tools are very old and OS dependent therefore, we will use both Windows or Linux.





### 3.2.5.1. Winfingerprint



The first tool we are going to use is called [Winfingerprint](#). Winfingerprint is an administrative network resource scanner that allows us to scan machines in our LAN in order to gather details about each host. This includes NetBIOS shares, disk information, services, users, groups, and more.

By selecting the boxes in its GUI, we are able to enumerate information such as user SID, password policy, users, shares and much more.

LearnSecurity  
Forging security professionals



As shown in the following screenshot, we just need to specify the hosts we want to scan, the NIC to use and the information we want to gather.

The screenshot shows the Winfingerprint application window with the following settings:

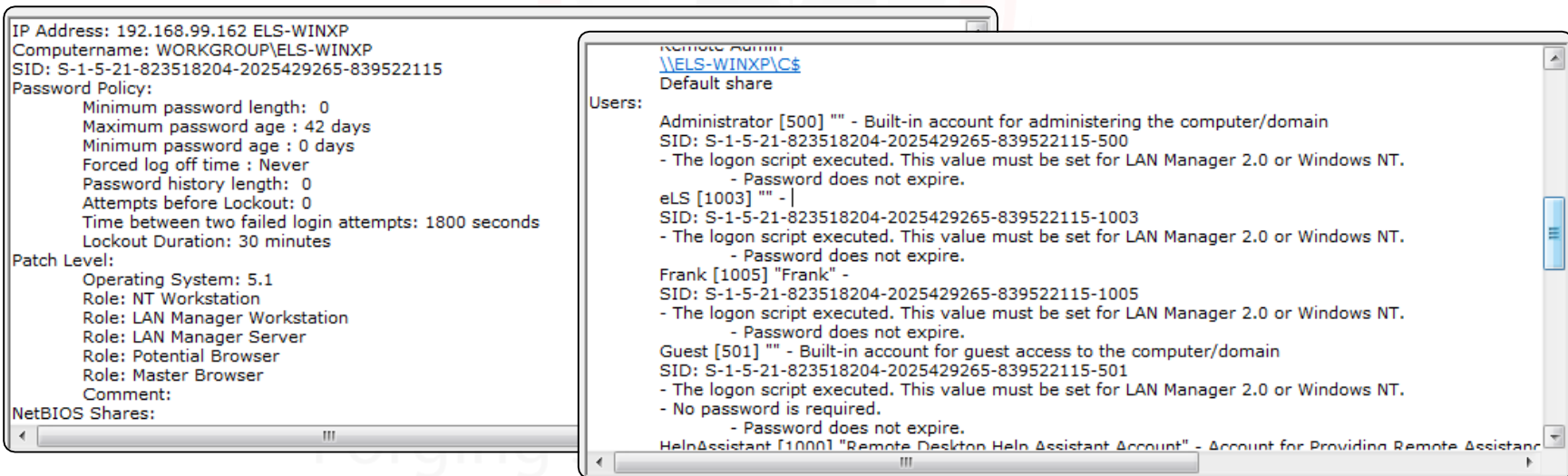
- Input Options:**
  - ☐ IP Range ☐ IP List
  - ☒ Single Host ☐ Neighborhood
  - IP Address:
- Scan Options:**
  - ☒ Domain ☐ Active Directory ☐ WMI API
  - ☒ Win32 OS Version ☒ Users ☐ Patch Level
  - ☒ Null IPC\$ Sessions ☒ Services ☒ MAC Address
  - ☒ NetBIOS Shares ☒ Disks ☒ Sessions
  - ☐ Date and Time ☒ Groups ☐ Event Log
  - ☐ Ping Host(s) ☐ RPC Bindings ☐ Show Errors
  - ☐ Traceroute Host
- General Options:**
  - Adapter:
  - Timeout for TCP/UDP/ICMP/SNMP:
  - Retries:  Max Connections:
  - ☐ TCP Portscan Range:
  - ☐ UDP Portscan Range:
  - ☐ SNMP Community String:

Buttons on the right:





Once everything is set, we can click on *Scan* and the tool will automatically exploit the null session. The results will be displayed in the bottom section of the GUI.





### 3.2.5.1. Winfingerprit



As you can see, it is very simple to use and it returns a very healthy amount of valuable information.

By inspecting the results, we can read users available on the machine, password policies (useful for later bruteforce attacks), groups, users SIDs and much more.





**Wininfo** is another simple enumeration tool for NetBIOS as it displays all the information through null sessions. In contrast with the previous tool, it does not offer a graphical interface. Instead we will have to run it from our command prompt.

The command to execute it is very simple:

```
wininfo <target_IP_Address> -n
```

where `-n` tells the tool to establish a null session before trying to dump the information.



The following snippet shows the `wininfo` results:

```
Null session established.  
USER ACCOUNTS:  
* Administrator  
  (This account is the built-in administrator account)  
* eLS  
* Frank  
* Guest  
  (This account is the built-in guest account)  
* HelpAssistant  
* netadmin  
* SUPPORT_388945a0  
SHARES:  
* My Documents  
* IPC$  
* Frank  
...
```

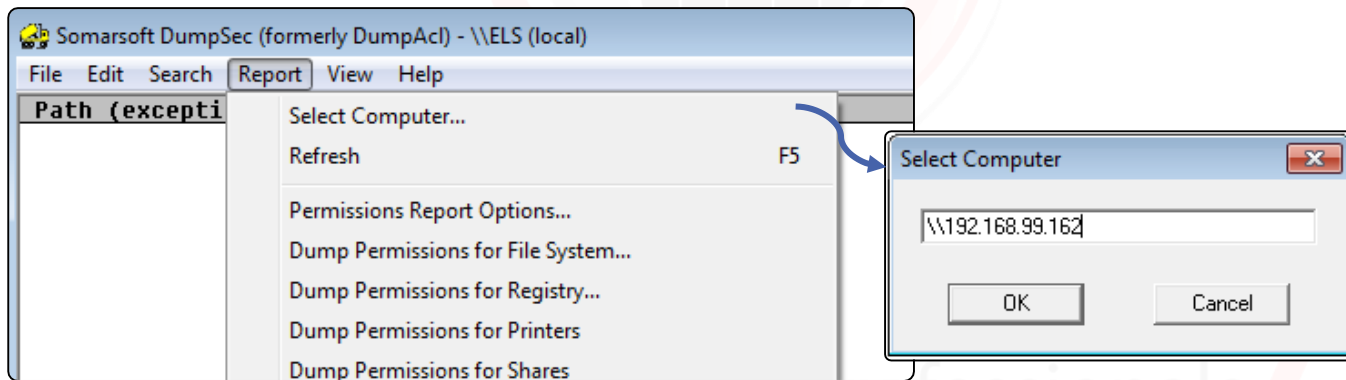


The final Windows tool we are going to inspect is called **DumpSec** (former *DumpAcl*).

DumpSec is an auditing tool that is able to gather file system information, registry, shares, users, groups and much more. It can be used both via its graphical interface or the command line.



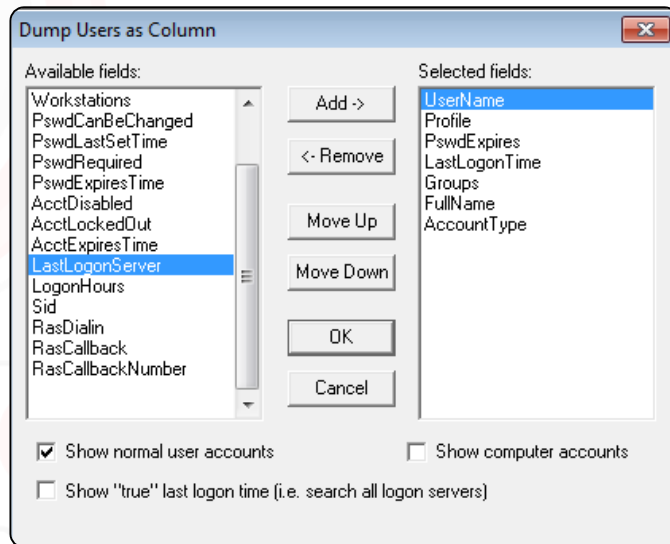
Once the we start the tool, we have to select our target by clicking on Report->Select Computer. Then we just need to type the IP address of our target: \\192.168.99.162.





After the target has been selected, we can instruct the tool to gather the information we want by clicking on *Report* and selecting what to dump. Here we are going to click on *Dump Users as column* and select some of the available fields.

When everything is set, we have to click on *OK* and wait for the tool to gather the information selected.





Once the tool completes its tasks, the information will be well organized in the main windows.

```
Somarsoft DumpSec (formerly DumpAcl) - \\192.168.99.162
File Edit Search Report View Help

UserName
Administrator
  Profile
  PswdExpires No
  LastLogonTime Never
  Groups Administrators (Local, Administrators have complete and unrestricted access to the computer/domain)
  FullName
  AccountType User
eLS
  Profile
  PswdExpires No
  LastLogonTime 2/23/2015 2:57 PM
  Groups Administrators (Local, Administrators have complete and unrestricted access to the computer/domain)
  FullName
  AccountType User
Frank
  Profile
  PswdExpires No
  LastLogonTime 2/18/2015 10:25 AM
  Groups Users (Local, Users are prevented from making accidental or intentional system-wide changes. Thus, Users can run certified applications, but not most legacy
  FullName Frank
  AccountType User
Guest
  Profile
  PswdExpires No
  LastLogonTime 12/11/2015 3:03 PM
  Groups Guests (Local, Guests have the same access as members of the Users group by default, except for the Guest account which is further restricted)
  FullName
  AccountType User
HelpAssistant
  Profile
  PswdExpires No

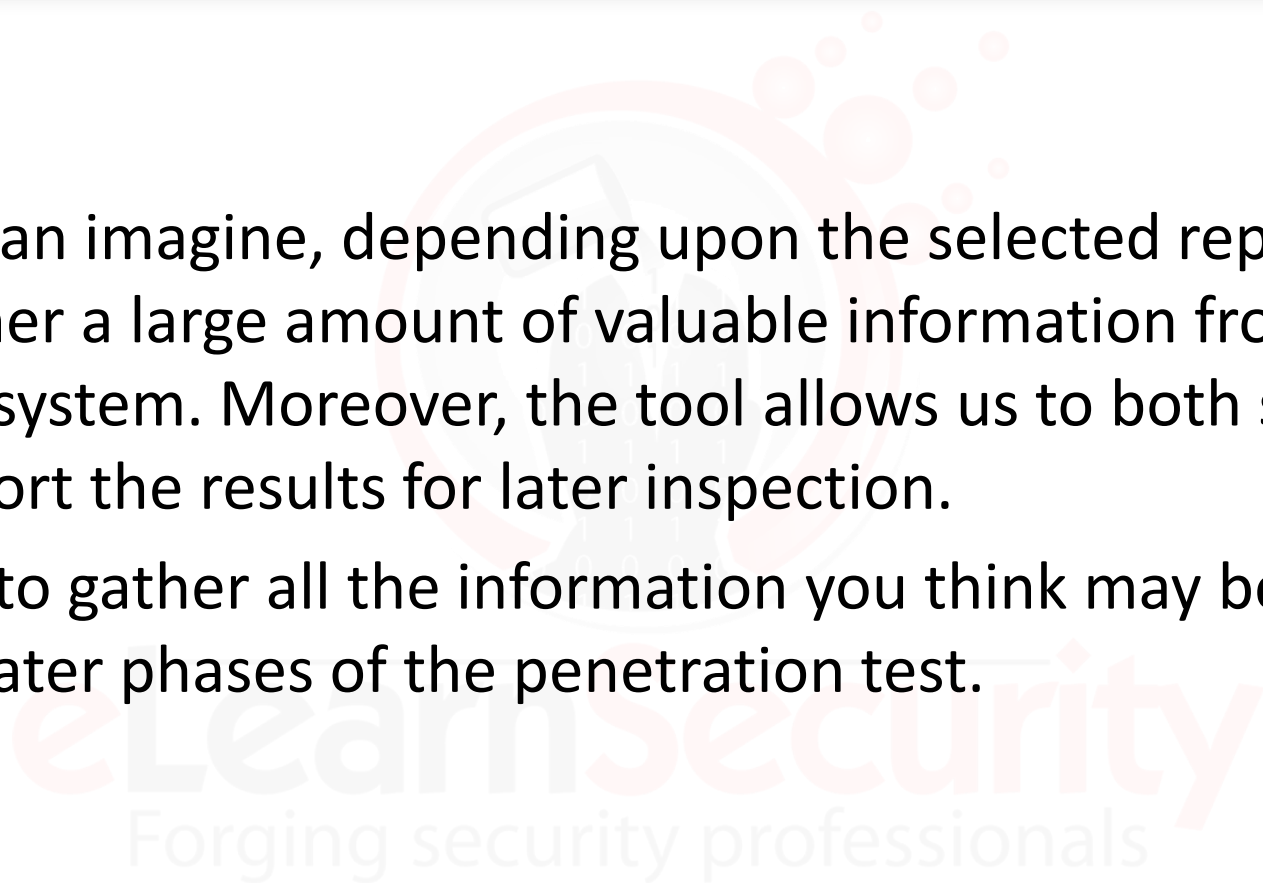
Found 7 users 00001
```





As you can imagine, depending upon the selected report you can gather a large amount of valuable information from the remote system. Moreover, the tool allows us to both search and export the results for later inspection.

Be sure to gather all the information you think may be useful for the later phases of the penetration test.





The tools studied up to this point only on run the Windows operating systems. Let's now take a look at some similar tools for Linux.

The first tool we are going to explore is [enum4linux](#). This is basically a wrapper around *rpcclient*, *net*, *nmblookup* and *smbclient*.

As you will see, enum4linux is both very easy to use and returns a great deal of valuable information. We can run it with the following command:

```
enum4linux <target_IP_Address>
```



Using our previous target, we will acquire output similar to the following:

```
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Dec 10 12:13:42 2015

=====
|   Target Information   |
=====
Target ..... 192.168.99.162
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain administrator

=====
|   Enumerating Workgroup/Domain on 192.168.99.162   |
=====
[+] Got domain/workgroup name: WORKGROUP

=====
|   Nbtstat Information for 192.168.99.162   |
=====
Looking up status of 192.168.99.162
  ELS-WINXP      <00> - B <ACTIVE>
  WORKGROUP      <00> - <GROUP> B <ACTIVE>
  ELS-WINXP      <20> - B <ACTIVE>
  WORKGROUP      <1e> - <GROUP> B <ACTIVE>
  WORKGROUP      <1d> - B <ACTIVE>
  .._MSBROWSE_.. <01> - <GROUP> B <ACTIVE>

=====
|   Users on 192.168.99.162   |
=====
index: 0x1 RID: 0x1f4 acb: 0x00000210 Account: Administrator Name: (null) Desc: Built-in account for administering the computer/domain
index: 0x2 RID: 0x3eb acb: 0x00000210 Account: eLS Name: (null) Desc: (null)
index: 0x3 RID: 0x3ed acb: 0x00000210 Account: Frank Name: Frank Desc: (null)
index: 0x4 RID: 0x1f5 acb: 0x00000214 Account: Guest Name: (null) Desc: Built-in account for guest access to the computer/domain
index: 0x5 RID: 0x3e8 acb: 0x00000211 Account: HelpAssistant Name: Remote Desktop Help Assistant Account Desc: Account for Providing Remote Assistance
index: 0x6 RID: 0x3ec acb: 0x00000210 Account: netadmin Name: netadmin Desc: (null)
index: 0x7 RID: 0x3ea acb: 0x00000211 Account: SUPPORT_388945a0 Name: CN=Microsoft Corporation, N=Redmond, S=Washington, C=US Desc: This is a vendor's account for the Help and Support Service

user:[Administrator] rid:[0x1f4]
user:[eLS] rid:[0x3eb]
user:[Frank] rid:[0x3ed]
user:[Guest] rid:[0x1f5]
user:[HelpAssistant] rid:[0x3e8]
user:[netadmin] rid:[0x3ec]
user:[SUPPORT_388945a0] rid:[0x3ea]

=====
|   Share Enumeration on 192.168.99.162   |
=====
Domain=[WORKGROUP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
Domain=[WORKGROUP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```



Although the previous screenshots show a small subset of the information obtained, [enum4linux](#) will gather and organize the information in the following way:

- Target Information
- Workgroup/Domain
- Domain SID
- OS information
- Users
- Share Enumeration
- Password Policy
- Groups
- Users SID
- Printer info



Another tool we can use on Linux operating systems is [rpcclient](#), a tool that can execute Microsoft RPC (Remote Procedure Call) functionalities. As we will see in future slides, it offers a multitude of commands that we can run on a remote machine.

In order to use it however, we must first establish a connection to the remote machine.

<https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html>



We can achieve this with the following command:

```
rpcclient -N -U "" <target_IP_Address>
```

where:

- `-N` instructs `rpcclient` not to ask for the password
- `-U ""` sets the network username (none in this case)

Once the command completes, the prompt changes and we are able to interact with it. We can list all the available commands with the `help` command.



## 3.2.5.5. Rpcclient



As you will see in the help output, there is a very long list of commands we can run. Each option will return specific information from the remote system. For example, we can retrieve the users available on the machine using the following command:

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[eLS] rid:[0x3eb]
user:[Frank] rid:[0x3ed]
user:[Guest] rid:[0x1f5]
user:[HelpAssistant] rid:[0x3e8]
user:[netadmin] rid:[0x3ec]
user:[SUPPORT_388945a0] rid:[0x3ea]
rpcclient $>
```





Notice that the tool offers the autocomplete feature! This simple means you can type part of the desired command and then list all the available options by hitting tab two times, see below:

```
rpcclient $> enum
enumalsgroups      enumdomusers      enummonitors      enumprocs
enumdata           enumdrivers      enumports         enumtrust
enumdataex        enumforms        enumprinters
enumdomains        enumjobs         enumprivs
enumdomgroups      enumkey          enumprocdatatypes
rpcclient $> enum
```

A small list of useful commands you may wish to run are as follows: enumalsgroups, srvinfo, lookupnames, queryuser, enumprivs.





As you can see in the following screenshot, rpcclient offers some very interesting commands. We suggest you test them in order to see how they work.

```

NETLOGON
logonctrl2      Logon Control 2
getanydcname    Get trusted DC name
getdcname       Get trusted PDC name
dsr_getdcname   Get trusted DC name
dsr_getdcnameex Get trusted DC name
dsr_getdcnameex2 Get trusted DC name
dsr_getsitename Get sitename
dsr_getforesttrustinfo Get Forest Trust
logonctrl       Logon Control
samsync         Sam Synchronisation
samdeltras     Query Sam Deltas
samlogon        Sam Logon
change_trust_pw Change Trust Account Password
gettrustrid     Get trust rid
dsr_enumtrustdom Enumerate trusted domains
dsenumdomtrusts Enumerate all trusted domains
deregisterdnsrecords Deregister DNS records
netrenumtrusteddomains Enumerate trusted domains
netrenumtrusteddomainsex Enumerate trusted domainsex

```

```

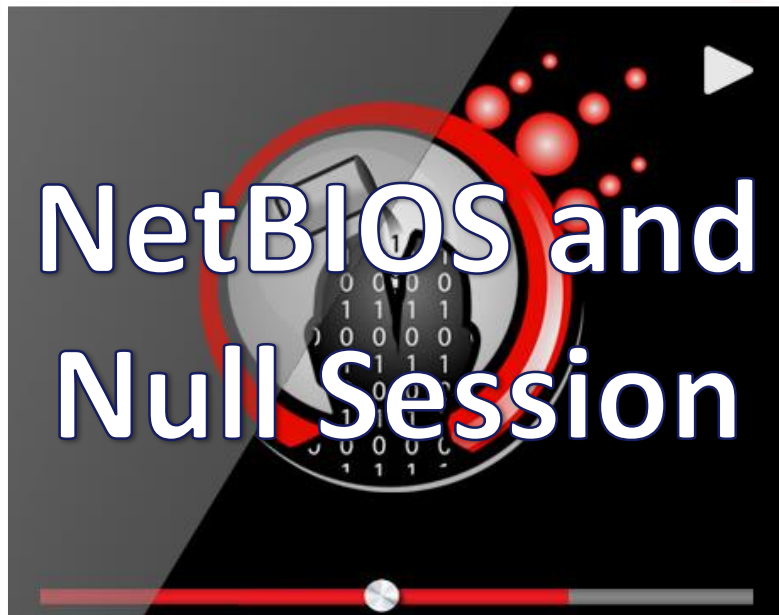
SAMR
queryuser       Query user info
querygroup      Query group info
queryusergroups Query user groups
queryuseraliases Query user aliases
querygroupmem   Query group membership
queryaliasmem   Query alias membership
queryaliasinfo  Query alias info
deletealias     Delete an alias
querydisplayinfo Query display info
querydisplayinfo2 Query display info
querydisplayinfo3 Query display info
querydomaininfo Query domain info
enumdomusers    Enumerate domain users
enumdomgroups   Enumerate domain groups
enumaliasgroups Enumerate alias groups
enumdomains     Enumerate domains
createdomuser    Create domain user
createdomgroup   Create domain group
createdomalias   Create domain alias
samlookupnames  Look up names
samlookuprids   Look up names
deletedomgroup   Delete domain group
deletedomuser    Delete domain user
samquerysecobj   Query SAMR security object

```

```

SRVSVC
srvinfo         Server query info
netshareenum    Enumerate shares
netshareenumall Enumerate all shares
netsharegetinfo Get Share Info
netsharesetinfo Set Share Info
netfileenum     Enumerate open files
netremotetod    Fetch remote time of day
netnamevalidate Validate sharename
netfilegetsec   Get File security
netsssdell      Delete Session
netsssenum      Enumerate Sessions
netdiskenum     Enumerate Disks
netconnenum     Enumerate Connections

```



If you have a **FULL** or **ELITE** plan you can click on the image on the left to start the video

Penetration Security  
Forging security professionals



# SIMPLE NETWORK MANAGEMENT PROTOCOL

eLearnSecurity  
Forging security professionals



### 3.3.1. What it is (and where it is used)



**SNMP** Stands for Simple Network Management Protocol and it is used for exchanging management information between network devices.

For example, SNMP may be used to configure a router or simply check its status.





### 3.3.1. What it is (and where it is used)



MAP



REF



VIDEO



LAB

69

In the SNMP protocol there is a manager and a number of agents. The agents either wait for the commands from the manager or send critical messages (trap) to the manager. The manager is usually a system administrator.





### 3.3.1. What it is (and where it is used)



Read

Write

Trap

Traversal  
Operations

There are four types of SNMP commands used to control and monitor managed devices:

- Read
- Write
- Trap
- Traversal Operations



### 3.3.1. What it is (and where it is used)



The **read** command is used to monitor devices, while the **write** command is used to configure devices and change device settings.

The **trap** command is used to "trap" events from the device and report them back to the monitoring system.

**Traversal operations** are used to determine what variables a certain device supports.



### 3.3.1. What it is (and where it is used)



There are multiple versions of SNMP however, all have their challenges with security.

SNMPv1 is both the original and most vulnerable (clear text protocol) however, SNMPv2 is just as likely to be compromised given its inherent weaknesses.

SNMPv3 is the newest version and, although it uses encryption, it is still susceptible to attacks like brute forcing.

ClearSecurity  
Forging security professionals



SNMP receives general messages on UDP port 161 and trap messages on UDP 162. SNMP works on the basis that network management systems send out a request and the managed devices (agents) return a response.

This is implemented using one of four operations (similar to HTTP verbs): Get, GetNext, Set, and Trap.



### 3.3.2. How it works (Agents, NMS, MIB...)



MAP



REF



VIDEO



LAB

74

SNMP messages consist of a header and a PDU (protocol data units). The **headers** consist of the SNMP version number and the *community string*, which is used as a form of “secure” password authentication in SNMP.

It is important to know that there are two types of community names or strings: *Private* and *Public*.

- Private community strings allow access to "write" rights
- Public allows for "read" rights on the system.

Forging security professionals



### 3.3.2. How it works (Agents, NMS, MIB...)



MAP



REF



VIDEO



LAB

75

The **PDU** depends on the type of message that is being sent.

The `Get`, `GetNext` and `Set`, as well as the PDU responses, consist of PDU type, Request ID, Error status, Error index and Object/variable fields.

The `Trap` contains fields like Enterprise, Agent, Agent address, Generic trap type, Specific trap code, Timestamp and Object/Value.

**eLearnSecurity**  
Forging security professionals

MIBs (Management Information Base) are a collection of definitions which define the properties of the managed object on the device (such as a router, switch, etc.).

In other words, it is a database of information that is relevant to the network manager.

[https://docs.oracle.com/cd/E13161\\_01/tuxedo/docs10gr3/snmpmref/1tmib.html#wp1032892](https://docs.oracle.com/cd/E13161_01/tuxedo/docs10gr3/snmpmref/1tmib.html#wp1032892)



### 3.3.2. How it works (Agents, NMS, MIB...)



MAP



REF



VIDEO



LAB

77

In order to keep items well organized, the database is structured as a tree thus, each *object* of this tree has a number and a name.

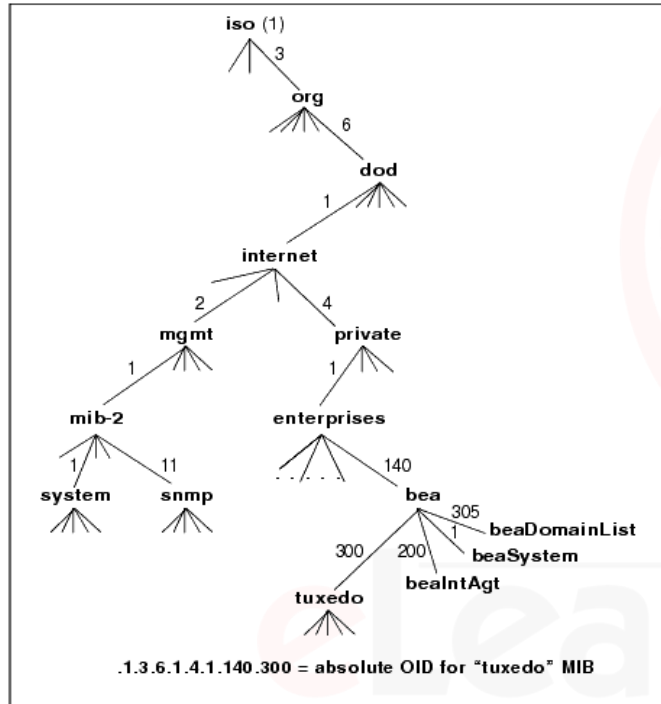
The complete path, from the top of the tree down to the point of interest, forms the name of that point called an **OID** (Object Identifier).

Nodes near the top of the tree are extremely general in nature.

eLearnSecurity  
Forging security professionals



## 3.3.2. How it works (Agents, NMS, MIB...)



You will find that all of the OID's will start with 1.3.6.1. Each leaf in the tree is a property of the device that can be read/written by the manager. A query will have to specify the OID address such as 1.3.6.1.4.1.140.305 (*beaDomainList*).

A sample of OID tree from an Oracle device

For example, to navigate to the `Internet` object, one has to reach to the fourth tier of the tree.

As one moves further down, the names become more and more specific. Once near the bottom, each node represents a particular feature on a specific device (or agent).





The following is a brief list of attacks that one can run against SNMP, which we will inspect in detail in future slides:

#### Flooding

- DOS attack which involves spoofing an SNMP agent and flooding the SNMP trap management with tens of thousands of SNMP traps, varying in size from 50 bytes to 32 kilobytes, until the SNMP management trap is unable to function properly.

#### Community

- Using Default community strings to gain privileged access to systems

#### Brute force

- Using a tool to guess the community strings used on a system to achieve elevated privileges.





Enumeration of SNMP information happens by utilizing tools and methods to list the information available within the system.

The type and amount of information will depend on the community string obtained, therefore the first skill to master is how to obtain the community strings.



### 3.3.3.2. Obtaining the Community Strings



MAP



REF



VIDEO



LAB

82

One of the easiest ways to obtain a community string is to sniff the network traffic.

Since SNMPv1 and SNMPv2 utilize clear text communications, it is easy to sniff the passwords coming from the network management systems.





### 3.3.3.2. Obtaining the Community Strings



Another way of obtaining the string is by using a dictionary attack. As you can imagine, having a good dictionary is key when performing this type of attack.

Beware though, most current Network Intrusion Detection Systems will alert to this activity as it sees the multiple login attempts with different strings.





### 3.3.3.2. Obtaining the Community Strings



MAP



REF



VIDEO



LAB

84

Once we acquire the string, we can move on to other tools in order to extract information from the remote device. Notice that **read** access is enough to extract a wealth of information (useful for later attacks).

Now that we know the steps to perform, let us see what tools we can use to achieve our goal.





Snmpwalk (part of the Net-SNMP suite) uses SNMP GETNEXT requests to query a network entity for a tree of information. Since an object identifier (OID) may be given on the command line, knowing the OID of the target device may be very useful.



This OID specifies which portion of the object identifier space will be searched using `GETNEXT` requests.

All variables in the subtree below the given OID are queried and their values presented to the user. If no OID is present, *snmpwalk* will search the subtree rooted at `SNMPv2-SMI::mib-2` (including any MIB object values from other MIB modules that are defined as lying within this subtree).



If the network entity has an error processing the request packet, an error packet will be returned. A message will then be shown, helping to pinpoint the reason the request was malformed.

If the tree search attempts to search beyond the end of the MIB, the message "End of MIB" will be displayed.





### 3.3.3.3. Snmpwalk



In a basic scenario, snmpwalk takes a single OID, and displays a list of all the results. These resided within the subtree rooted on this OID.

We will show a snippet of the output in the next slide. Notice that, since the tool outputs a wealth of information, we may want to pipe the requests to files for later inspections.







Here is the output of snmpwalk against our target machine:

```
stduser@els:~$ snmpwalk -v 2c 192.168.102.149 -c public
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: Intel64 Family 6 Model 42
Stepping 7 AT/AT COMPATIBLE - Software: Windows Version 6.1 (Build
7601 Multiprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.1
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (872094) 2:25:20.94
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: els
SNMPv2-MIB::sysLocation.0 = STRING:
```

The `-v` option specifies the SNMP version to use (`2c`), while `-c` sets the community string to use (`public`).



## IMPORTANT

If the output returns the OID numerically, as the following example,

```
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family..."
```

please be sure to install the `snmp-mibs-downloader` package. Once installed, comment the fourth line in the following file `/etc/snmp/snmp.conf`:

```
# As the snmp packages come without MIB files due to license reasons, loading
# of MIBs is disabled by default. If you added the MIBs you can reenale
# loading them by commenting out the following line.
#mibs :
```



Snmpwalk can also be used with either a single MIB object, or even an exact OID instance (returning the corresponding value). We will see an example in the next slide.

Conversely, it is also possible to start the walk at a higher level, retrieving more than one group of information. This would typically retrieve all the information known to an agent.



If we want to list only the software installed on the machine we can specify the following OID:

```
stduser@els:~$ snmpwalk -c public -v1 192.168.102.149 hrSWInstalledName
HOST-RESOURCES-MIB::hrSWInstalledName.1 = STRING: "Microsoft Visual Studio
2010 Tools for Office Runtime (x64)"
HOST-RESOURCES-MIB::hrSWInstalledName.2 = STRING: "OpenVPN 2.3.8-I601 "
HOST-RESOURCES-MIB::hrSWInstalledName.4 = STRING: "WinRAR 5.01 beta 1"
HOST-RESOURCES-MIB::hrSWInstalledName.5 = STRING: "Microsoft Visual C++ 2010
x64 Redistributable - 10.0.40219"
HOST-RESOURCES-MIB::hrSWInstalledName.6 = STRING: "Java 8 Update 31"
HOST-RESOURCES-MIB::hrSWInstalledName.7 = STRING: "VMware Tools"
HOST-RESOURCES-MIB::hrSWInstalledName.8 = STRING: "Microsoft Visual C++ 2008
Redistributable - x64 9.0.30729.6161"
HOST-RESOURCES-MIB::hrSWInstalledName.9 = STRING: "Java SE Development Kit 7"
HOST-RESOURCES-MIB::hrSWInstalledName.11 = STRING: "Microsoft .NET Framework
4.5.1"
```



[Snmpwalk](#) is very useful in gaining information from a system but, as stated earlier, one must minimally understand how SNMP works. Moreover, we strongly suggest you check its manual, since it offers both useful and customizable options.

<http://www.net-snmp.org/wiki/index.php/TUT:snmpwalk>





[Snmpset](#) (part of the Net-SNMP suite) is an SNMP application that uses `SNMP SET` requests to either set or change information on a network entity.

In other words, the `SET` operation allows either the management application or, the manager, to set the value of an attribute (of a managed object) in the agent.

<http://www.net-snmp.org/docs/man/snmpset.html>



### 3.3.3.4. Snmpset



Please note that one or more object identifiers (OIDs) must be given as arguments on the command line.

In addition to the OID, a `type` (string, integer, etc.) and a `value` must also be provided.





Before actually setting the new value for a specific object, let's first check its actual value with `snmpwalk`. In our example we will target the `sysContact` OID:

```
>>snmpwalk -v 2c -c public 192.168.102.149 system.sysContact.0  
SNMPv2-MIB::sysContact.0 = STRING: admin@els.com
```

As we can see, at the moment, the value is set to `admin@els.com` and the type is `STRING`.

LearnSecurity  
Forging security professionals





### 3.3.3.4. Snmpset



Let us now try to both change its value with the following *snmpset* command and then print its value to verify the changes.

```
>>snmpset -v 2c -c public 192.168.102.149 system.sysContact.0 s new@els.com  
SNMPv2-MIB::sysContact.0 = STRING: new@els.com
```

Above, **s** tells *snmpset* that we want to use a STRING type, while **new@els.com** is the new value for the entity. Let us run the *snmpwalk* once again and see what we get:

```
snmpwalk -v 2c -c public 192.168.102.149 system.sysContact.0  
SNMPv2-MIB::sysContact.0 = STRING: new@els.com
```

Changed!



Notice that in *snmpset* the `-v` and `-c` options are used in the same way as *snmpwalk*. The only difference really is that we have two new arguments: one for the *type* and one for the *value* we are going to set.

We can see all the available types in the *snmpset* manual:

```
TYPE: one of i, u, t, a, o, s, x, d, b
i: INTEGER, u: unsigned INTEGER, t: TIMETICKS, a: IPADDRESS
o: OBJID, s: STRING, x: HEX STRING, d: DECIMAL STRING, b: BITS
U: unsigned int64, I: signed int64, F: float, D: double
```



### 3.3.3.5. Nmap SNMP script



As you already know, Nmap is one of the most powerful tools we can use during our scanning and enumerating phase.

When facing SNMP services, Nmap comes with some basic, yet useful scripts: `snmp-brute`, `snmp-info`, `snmp-interfaces`, `snmp-netstat`, `snmp-processes`, `snmp-sysdescr`, `snmp-win32-services` and few more.

You can list them by navigating into the Nmap script folder and then running the following command:

```
stduser@els:/usr/share/nmap/scripts$ ls -l | grep -i snmp
```



Depending on the script you wish to run, you may have to set different options. Most of these can be executed with the following syntax as long as you are running as root:

```
nmap -sU -p 161 --script=<script_name> <IP_address>
```

Let's take a look at some examples to better understand what information we can obtain from these scripts.

Clear Security  
Forging security professionals



The first script we want to run allows us to enumerate the services available on the target machine:

```
sudo nmap -sU -p 161 --script=snmp-win32-services 192.168.102.149
```

```
stduser@els:~$ sudo nmap -sU -p 161 --script=snmp-win32-services 192.168.102.149

Starting Nmap 7.00 ( https://nmap.org ) at 2015-12-15 08:41 EST
Nmap scan report for 192.168.102.149
Host is up (0.00027s latency).
PORT      STATE SERVICE
161/udp   open  snmp
| snmp-win32-services:
|   Application Information
|   Background Intelligent Transfer Service
|   Base Filtering Engine
|   COM+ Event System
|   Cryptographic Services
|   DCOM Server Process Launcher
|   DHCP Client
|   DNS Client
|   Desktop Window Manager Session Manager
```



### 3.3.3.5. Nmap SNMP script



As you can imagine, depending upon the script run, we will be able to gather very specific information from the remote host.

We suggest you try these scripts in order to see how they work and what kind of information SNMP may reveal.





### 3.3.3.5. Nmap SNMP script



MAP



REF



VIDEO



LAB

103

Another useful script that we can use from Nmap is `snmp-brute`. The script tries to brute force the community name used by the remote SNMP device by using its own wordlists. *Snmp-brute* is quite fast and is able to find the community names in a matter of minutes.





Let us suppose we have found a machine running a SNMP service but, unfortunately, we do not know the correct community string. We can run the Nmap snmp-brute script to find the correct string to use.

The easiest way to run it is using this command:

```
sudo nmap -sU -p 161 192.168.102.149 --script snmp-brute
```

Notice that the default wordlist used by Nmap is stored here:  
`/usr/share/nmap/nselib/data/snmpcommunities.lst`.





Let's see results from the previous command:

```
stduser@els:~$ sudo nmap -sU -p 161 192.168.102.149 --script snmp-brute

Starting Nmap 7.00 ( https://nmap.org ) at 2015-12-16 04:52 EST
Nmap scan report for 192.168.102.149
Host is up (0.00024s latency).
PORT      STATE      SERVICE
161/udp    open|filtered snmp
| snmp-brute:
|   public - Valid credentials
|_  admin  - Valid credentials
MAC Address: 00:0C:29:24:DD:54 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.85 seconds
```

As we can see from the screenshot, Nmap finds two valid community strings: `public` and `admin`.



Since the default community string wordlist is quite small, Nmap offers the ability to use a custom wordlist by adding the following option to the previous command:

```
--script-args snmp-brute.communitiesdb=<wordlist>
```

We just need to replace `<wordlist>` with the path of our own wordlist. If you have the [seclists](#) package on your machine, you can find a good wordlist at the following path:

```
/usr/share/seclists/Misc/wordlist-common-snmp-community-strings.txt
```

<https://github.com/danielmiessler/SecLists>



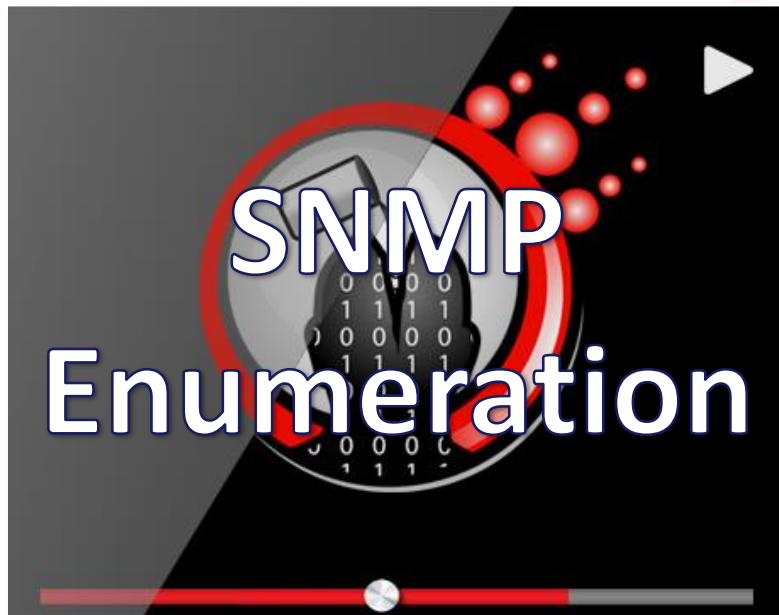
Let's modify the previous command and use our own wordlist this time. The complete command will read as follows:

```
sudo nmap -sU -p 161 192.168.102.149 --script snmp-brute --script-args  
snmp-brute.communitiesdb=/usr/share/seclists/Misc/wordlist-common-  
snmp-community-strings.txt
```

```
Starting Nmap 7.00 ( https://nmap.org ) at 2015-12-16 05:01 EST  
Nmap scan report for 192.168.102.149  
Host is up (0.00021s latency).  
PORT      STATE      SERVICE  
161/udp   open|filtered snmp  
| snmp-brute:  
|   public - Valid credentials  
|   admin  - Valid credentials  
|   internal - Valid credentials  
MAC Address: 00:0C:29:24:DD:54 (VMware)  
Nmap done: 1 IP address (1 host up) scanned in 6.48 seconds
```

We have found a new community string: **internal**

# Video: SNMP Enumeration



If you have a **FULL** or **ELITE** plan you can click on the image on the left to start the video

Penetration Security  
Forging security professionals



As we have seen, NetBIOS and SNMP provide a wealth of information to a pen tester. As previously stated, be sure to store all information obtained from these protocols as it may be valuable later in additional tests.

These are not the only services that may reveal information: SSH, FTP, Telnet, DNS, HTTP/S, LDAP, SQL servers, NFS, IPSec and many more should also be the targets of our tests. We will explore some of them in the next modules.

Forging security professionals



# REFERENCES

eLearnSecurity  
Forging security professionals



## NetBIOS RFC 1001

<https://tools.ietf.org/html/rfc1001>



## NetBIOS RFC 1002

<https://tools.ietf.org/html/rfc1002>



## NetBIOS

<https://technet.microsoft.com/en-us/library/bb962072.aspx>



## NetBIOS name resolution

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738412\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc738412(v=ws.10))



## Name Service Suffixes

<https://msdn.microsoft.com/en-us/library/cc224454.aspx>



## WINS Overview

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc725802\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc725802(v=ws.11))



## What is WINS

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784180\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784180(v=ws.10))



## WINS defined

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784707\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784707(v=ws.10))



# References



## SMB directly over TCP/IP

<https://support.microsoft.com/en-us/help/204279/direct-hosting-of-smb-over-tcp-ip>



## Nbtstat

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc940106\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc940106(v=technet.10))



## Microsoft net command

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490949\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490949(v=technet.10))



## Net view command

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh875576\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh875576(v=ws.11))



## Net use command

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490717\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490717(v=technet.10))



## Enum4Linux

<https://labs.portcullis.co.uk/tools/enum4linux/>



## RPCClient

<https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html>



## SNMPwalk

<http://www.net-snmp.org/docs/man/snmpwalk.html>



# References



## Net-SNMP

<http://www.net-snmp.org/>



## Seclists

<https://github.com/danielmiessler/SecLists>



## SNMPset

<http://www.net-snmp.org/docs/man/snmpset.html>

eLearnSecurity  
Forging security professionals



## NetBIOS and Null Session



## SNMP Enumeration

eLearnSecurity  
Forging security professionals



## NetBIOS Hacking

You're asked to check if it is possible to access the documents of the organization from outside the corporate networks.



## SNMP Analysis

Your customer is Sportsfoo.com and they want your help in order to test the security of their company.