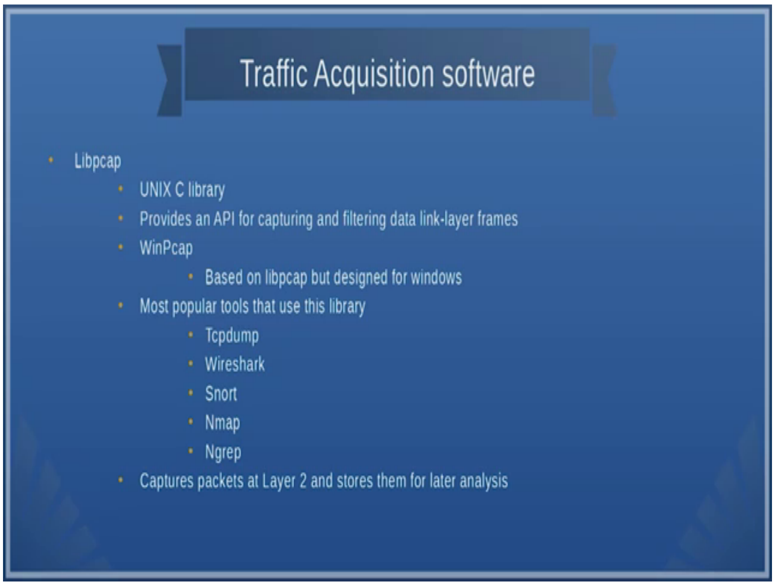
Information security - IV
Prof. M J Shankar Raman
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 37
Packet Capture Tools and Methods

Welcome to this session on network forensics. Until now, we have been seeing the technology that was used for traffic acquisition. Especially we were looking at what are all the types of devices and what is the forensic value that each of the divides provides. We were looking at security devices such as firewalls, ideas, IPs etcetera. You are also looking at the routers, the switches, and the application servers, authentication servers, and web proxies. Now, in all of this you have to acquire the traffic.

So, one of the ways the we acquire traffic is through software and the most important library that we make use of is the p cap library ok. It is actually unique c library, and it provides application programming interface for capturing and filtering the link layer frames.

(Refer Slide Time: 01:30)

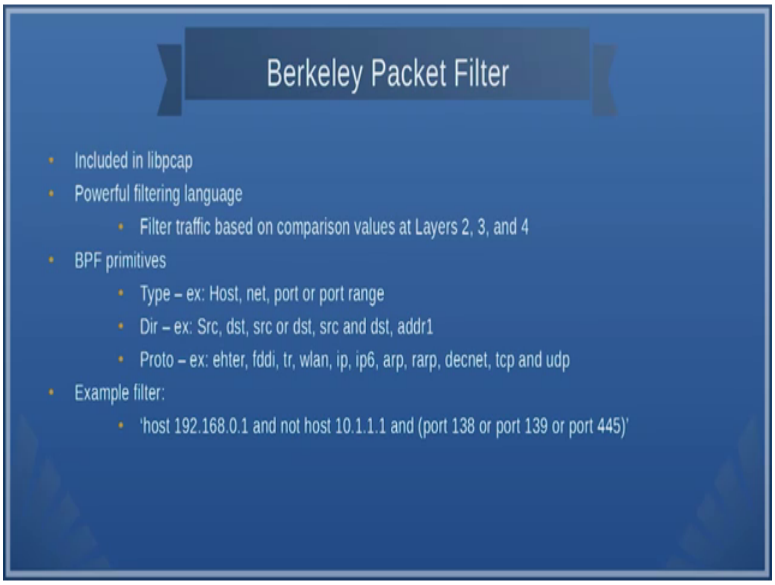


Traffic Acquisition software

- Libpcap
  - UNIX C library
  - Provides an API for capturing and filtering data link-layer frames
  - WinPcap
    - Based on libpcap but designed for windows
  - Most popular tools that use this library
    - Tcpdump
    - Wireshark
    - Snort
    - Nmap
    - Ngrep
  - Captures packets at Layer 2 and stores them for later analysis

You also have been p cap and cap, p cap is used for windows and many of the tools such as TCPdump ok, wireshark, snort, nmap, ngrep etcetera. They actually make use of this libpcap. What it does is it takes layer two information and stores them for analysis.

(Refer Slide Time: 02:02)



Berkeley Packet Filter

- Included in libpcap
- Powerful filtering language
  - Filter traffic based on comparison values at Layers 2, 3, and 4
- BPF primitives
  - Type – ex: Host, net, port or port range
  - Dir – ex: Src, dst, src or dst, src and dst, addr1
  - Proto – ex: ehter, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp
- Example filter:
  - 'host 192.168.0.1 and not host 10.1.1.1 and (port 138 or port 139 or port 445)'

So, the libpcap actually uses Berkeley packet filter or BPF ok, and it is actually I should say its it is filtering language that we you use. See because you can capture all the data, and then as I told you once I capture all the packets this use of libpcap it actually captures all the packets flow on your network. Now, once I capture the packet I need to start filtering those packets because you want only take those that is how interest you do not I mean and if you look at this speed of many of these routers its megabytes per second or pita bytes etcetera. So, you can collect almost all the data. So, you just have to collect the data and filtered those wanted parts. How do you filter it?

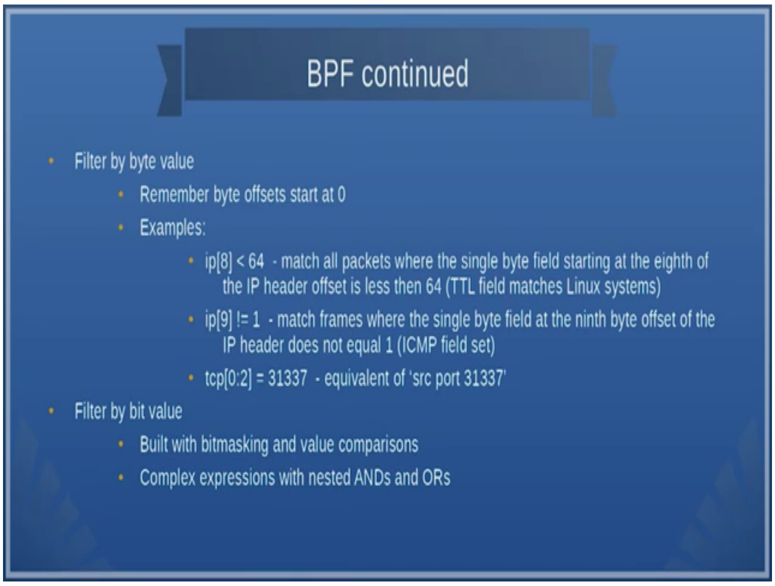
Now, this Berkeley packet filter actually provides a powerful filtering language and you can use this language to filter packets and classify them as layer 2 layer, 3 layer, 4 or you can have lot of combination I mean it is a huge task and in order to simplify this filtering activities you can use g y graphic user interface software called wireshark. But then before we go into that let us look at what you can do if you have to use only command line interfaces ok.

So, the BPF primitives are divided into at least 3 categories ok, one is the type for example, what type of host, what is the kind of network, what is the port or the port range is one category. Second what is the source the destination whether you want to filter the packets based on the address, I mean each one of these is a filter parameter whether you want to filter based on the host, you want to filter based on the net you want to filter based on the fort etcetera and you can also split based on the protocols.

For example whether you want to look at the internet package you want to look at the IP packets whether you want to IP look at IP version 6 etcetera. So, here is an example of a filter that we show. It says that host 192 dot 168 dot 0 dot 1 and not host 10 dot 1 dot 1 dot 1 and one of these ports that is port 128 or port 139 or port 445. So, what does it mean? It tells you that he want to capture all the packets from host 192 dot 168 dot 0 dot 1. And not that is not related with 10 dot 1 dot 1 1, but the ports on which the data packet uses or on which the software uses is 138, 139 or 445. So, it capture all these packets.

You can filter by byte value because if you look at IP and other address you actually have a 32 bit and if you looked at the diagram that would be familiar with which we had seen in the previous classes I mean each byte makes a huge difference. For example, what does this first byte to what is the second byte to an even within the byte the bit each bit is important. So, Berkeley packet filter provides you address each of this words bits and to the bit level ok.

(Refer Slide Time: 05:41)



BPF continued

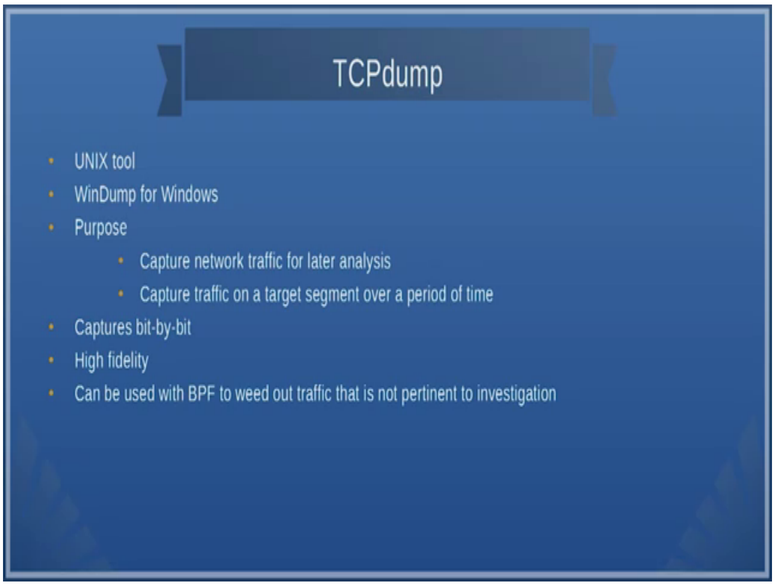
- Filter by byte value
  - Remember byte offsets start at 0
  - Examples:
    - ip[8] < 64 - match all packets where the single byte field starting at the eighth of the IP header offset is less then 64 (TTL field matches Linux systems)
    - ip[9] != 1 - match frames where the single byte field at the ninth byte offset of the IP header does not equal 1 (ICMP field set)
    - tcp[0:2] = 31337 - equivalent of 'src port 31337'
- Filter by bit value
  - Built with bitmasking and value comparisons
  - Complex expressions with nested ANDs and ORs

So, for example, it says tcp colon 0 colon 2 is equal to 3 1 3, 3 it says that the source port is 31377 ok. So, find out what is this tcp 0 colon 2. So, it say 0 and 1 ok. What is 0, 1 and 2. So, what does it means? Just take a look at the tcp header you will understand. So, when you are reading the slide take care keep next to you tcp header or ip header diagrams and see what actually it means then you will understand what is for example, ip of 8 less than 64 it says match all packets for the single byte field starting at the 8th of the ip header offset is less than 64 which essential means that TTL fields matches the Linux systems, so any way.

So, ip of 9 not equal to 1 which means match frames were the single byte field at the 9th byte offset the ip address does not equal to 1. So, essentially what you should do here is you take the ip the packet the headers and then divided into bytes and then find out what each of those bytes mean, and then out of this what does exactly the statements. So, so that is what I we told you that you should have an idea about ip headers and tcp headers etcetera before you do some forensic analysis. So, you should know how to filter, what to filter.

You can also filter based on the bit values ok. So, you can do something known as bit masking those were familiar with C programming language will understand the use of the AND operator, and OR operator for bit masking. You can create complex expressions with and then or switch it seen earlier, like host 192, 168 dot 0 dot 1 and not host 10 dot 1 dot 1 dot 1 etcetera.

(Refer Slide Time: 07:42)

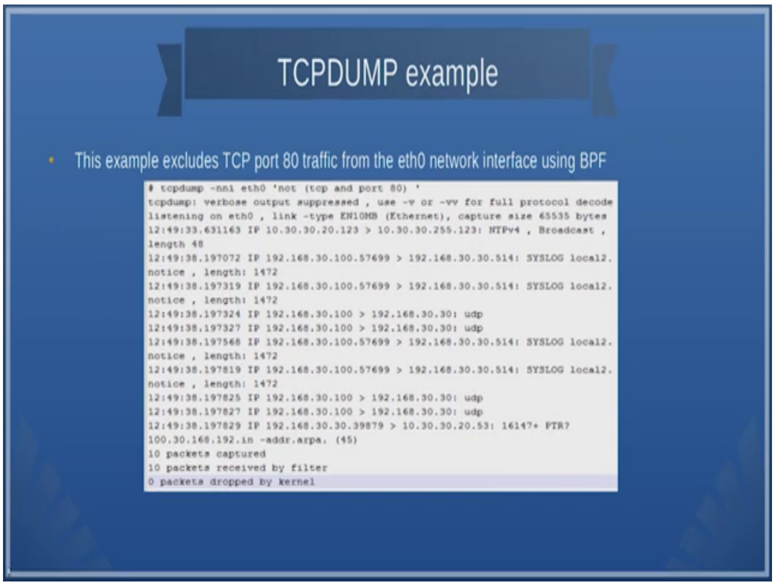


TCPdump

- UNIX tool
- WinDump for Windows
- Purpose
  - Capture network traffic for later analysis
  - Capture traffic on a target segment over a period of time
- Captures bit-by-bit
- High fidelity
- Can be used with BPF to weed out traffic that is not pertinent to investigation

The TCPdump is a tool which can be used to dump a filter and then dump packets ok. It is actually a unique tool and in windows is called windump ok. So, it actually bit by bit and it use can be used with BPF to read out those packets that are not pertinent for our investigations. So, here is an example of how to use TCPdump.

(Refer Slide Time: 08:00)



TCPDUMP example

- This example excludes TCP port 80 traffic from the eth0 network interface using BPF

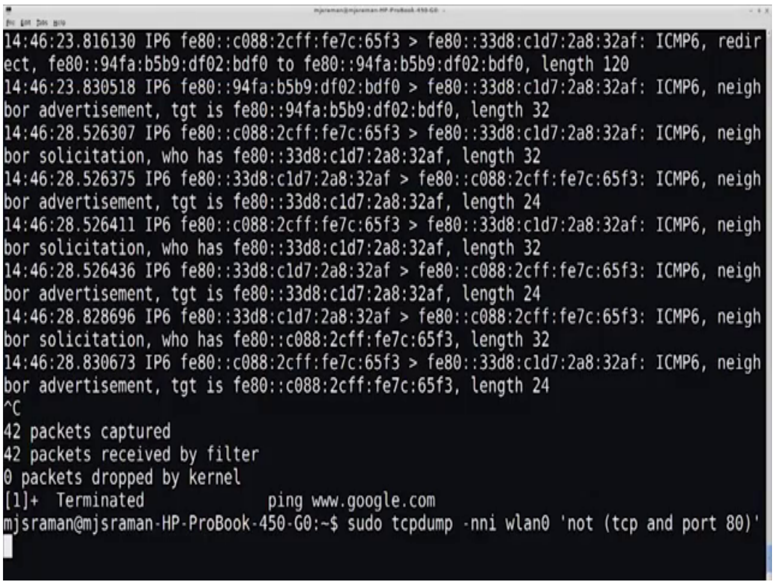
```
# tcpdump -nni eth0 'not (tcp and port 80) '
tcpdump: verbose output suppressed , use -v or -vv for full protocol decode
listening on eth0 , link -type EN10MB (Ethernet), capture size 65535 bytes
12:49:33.631163 IP 10.30.30.20.123 > 10.30.30.255.123: NTPv4 , Broadcast ,
length 48
12:49:38.197072 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197319 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197324 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197327 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197568 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197819 IP 192.168.30.100.57699 > 192.168.30.30.514: SYSLOG local2.
notice , length: 1472
12:49:38.197825 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197827 IP 192.168.30.100 > 192.168.30.30: udp
12:49:38.197829 IP 192.168.30.30.39879 > 10.30.30.20.53: 16147+ PTR?
100.30.168.192.in -addr.arpa. (45)
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

So, if you look at this example exclude TCP port traffic 80 from the Ethernet 0 network interface using BPF. So, let us take example of how to use TCPdump ok.

The following example that is sly that is presented tells you how to capture data packets on eth0 this Ethernet and what it tells is that we need to capture TCP and port 80 and not of this. So, what does it mean? I do not want TCP and packets that go to port 80, but I want our packets.

Now, just to show you a demo see here we have mentioned Ethernet 0, but I can show you that this is this can also be done with wireless interface also.

(Refer Slide Time: 08:48)



```
14:46:23.816130 IP6 fe80::c088:2cff:fe7c:65f3 > fe80::33d8:c1d7:2a8:32af: ICMP6, redir
ect, fe80::94fa:b5b9:df02:bdf0 to fe80::94fa:b5b9:df02:bdf0, length 120
14:46:23.830518 IP6 fe80::94fa:b5b9:df02:bdf0 > fe80::33d8:c1d7:2a8:32af: ICMP6, neigh
bor advertisement, tgt is fe80::94fa:b5b9:df02:bdf0, length 32
14:46:28.526307 IP6 fe80::c088:2cff:fe7c:65f3 > fe80::33d8:c1d7:2a8:32af: ICMP6, neigh
bor solicitation, who has fe80::33d8:c1d7:2a8:32af, length 32
14:46:28.526375 IP6 fe80::33d8:c1d7:2a8:32af > fe80::c088:2cff:fe7c:65f3: ICMP6, neigh
bor advertisement, tgt is fe80::33d8:c1d7:2a8:32af, length 24
14:46:28.526411 IP6 fe80::c088:2cff:fe7c:65f3 > fe80::33d8:c1d7:2a8:32af: ICMP6, neigh
bor solicitation, who has fe80::33d8:c1d7:2a8:32af, length 32
14:46:28.526436 IP6 fe80::33d8:c1d7:2a8:32af > fe80::c088:2cff:fe7c:65f3: ICMP6, neigh
bor advertisement, tgt is fe80::33d8:c1d7:2a8:32af, length 24
14:46:28.828696 IP6 fe80::33d8:c1d7:2a8:32af > fe80::c088:2cff:fe7c:65f3: ICMP6, neigh
bor solicitation, who has fe80::c088:2cff:fe7c:65f3, length 32
14:46:28.830673 IP6 fe80::c088:2cff:fe7c:65f3 > fe80::33d8:c1d7:2a8:32af: ICMP6, neigh
bor advertisement, tgt is fe80::c088:2cff:fe7c:65f3, length 24
^C
42 packets captured
42 packets received by filter
0 packets dropped by kernel
[1]+  Terminated              ping www.google.com
mjsraman@mjsraman-HP-ProBook-450-G0:~$ sudo tcpdump -nni wlan0 'not (tcp and port 80)'
```

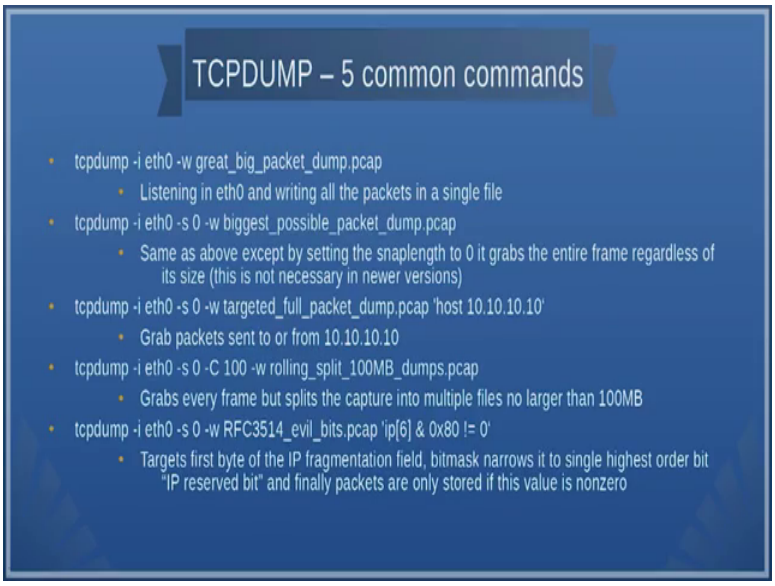
Here is what I have done to show you the wireless LAN interface. So, I have used just added the sudo in front of the TCPdump command this is to ensure that I have the necessary permissions the root permissions because all these tools most of these tools you should use has root in a Linux machine you should have full administrative rights.

So, here what we are doing is that I will just using the same command, but this time on the wireless interface. So, what I now, I am planning to do is that I am trying to do a ping of some website and try to see whether this command actually dumps the packets that I need. So, what I will do is I will now try to dump let us say nptel dot i t m dot ac dot in ok, or let say I dump from a ping to Google.

So, what is happening here is that once I start pinging some websites say I am trying to ping Google ok, what it does is that it actually starts capturing what are all the packets that go from my machine to Googles website ok. So, if you look at this. So, now, I will stop pinging Google. So, now, you see the data the capture has stopped again I will try to send some dns packets. So, what I will try to do is. So, I am try to find out what is the IP address of Google dot com and if you see this ok. So, I am able to send the dns packets. So, this guy is able to capture all those dns packets and then it is able to get the response then it also capturing the ntp packets it is network time protocol packets etcetera.

So, if you look at this. So, other than TCP packets it is capturing all IP packets and other than those which I sent to port 20 port 80 it captures all the packets. So, this is one way of catching the packet and of course, you can write it to a file. So, this TCP command has ok, so let it be capturing the packets. So, this TCPdump command has lot of other options.

(Refer Slide Time: 11:11)



TCPDUMP – 5 common commands

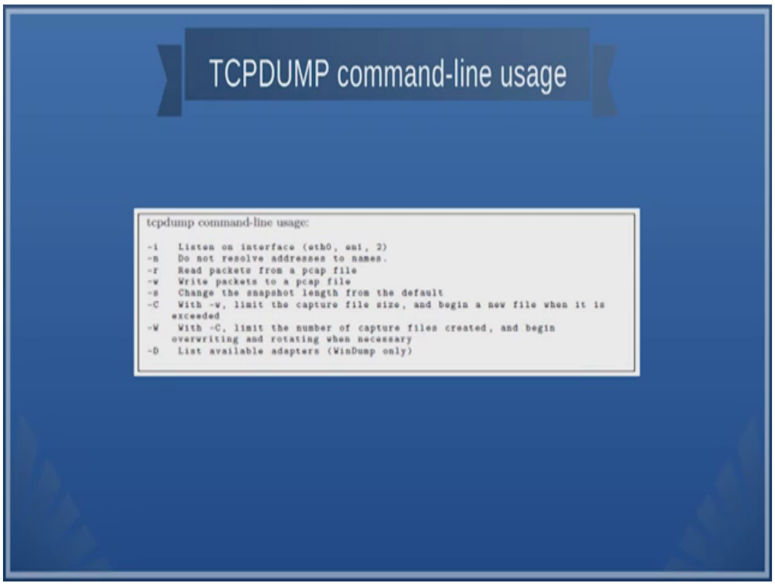
- tcpdump -i eth0 -w great_big_packet_dump.pcap
  - Listening in eth0 and writing all the packets in a single file
- tcpdump -i eth0 -s 0 -w biggest_possible_packet_dump.pcap
  - Same as above except by setting the snaplength to 0 it grabs the entire frame regardless of its size (this is not necessary in newer versions)
- tcpdump -i eth0 -s 0 -w targeted_full_packet_dump.pcap 'host 10.10.10.10'
  - Grab packets sent to or from 10.10.10.10
- tcpdump -i eth0 -s 0 -C 100 -w rolling_split_100MB_dumps.pcap
  - Grabs every frame but splits the capture into multiple files no larger than 100MB
- tcpdump -i eth0 -s 0 -w RFC3514_evil_bits.pcap 'ip[6] & 0x80 != 0'
  - Targets first byte of the IP fragmentation field, bitmask narrows it to single highest order bit "IP reserved bit" and finally packets are only stored if this value is nonzero

So, here are 5 command options. So, one of them ok, so if you look at 1 2 3 4th, 4th command where it says minus w rolling underscores split underscore 100 MB underscore dumps dot pcap. So, essentially what it does is grabs every frame, but splits the capture into multiple files no longer than 100 MB. So, so in this way you can actually capture the file and you can actually right it to you can actually capture the packets you can actually right it to a particular file and then you can do lots all sorts of filtering etcetera.

So, each of the command tell you what you are supposed to do for example, if I want to listen to eth 0 and write all the packets in a single file then I give the command TCPdump minus i that is the interface and then minus w tells you what file to I need to write. Then I used say and then similar to this suppose I just want to grab all the packets sent to or from 10 dot 10 dot 10 dot 10 then I say that TCPdump minus I which is the interface then I say almost source I say from almost all the packets whatever is goes to host 10 10 10 10 you just capture all those things and then write it into targeted underscore full packet underscore dump dot pcap etcetera. So, in this way you can use

TCPdump and capture all the packets and remember this is a kind of live dump that we are doing right now.

(Refer Slide Time: 12:50)

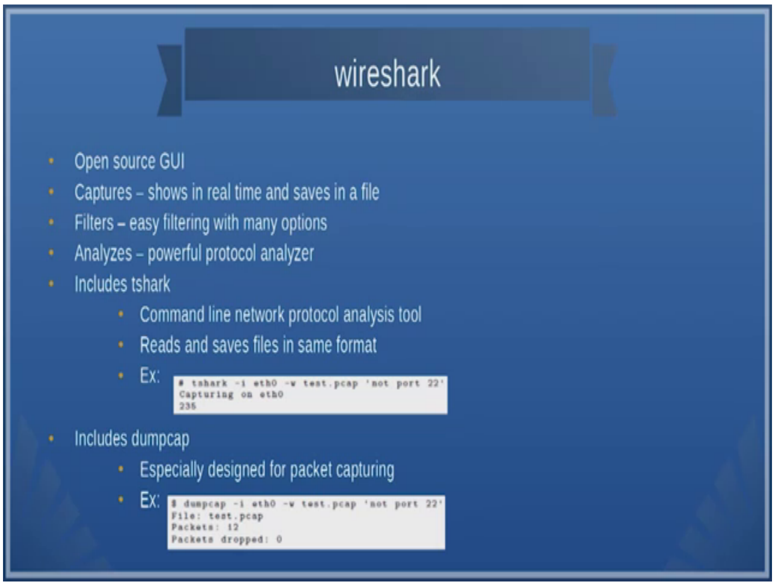


TCPDUMP command-line usage

```
tcpdump command-line usage:

-i   Listen on interface (eth0, en1, 2)
-n   Do not resolve addresses to names.
-r   Read packets from a pcap file
-w   Write packets to a pcap file
-s   Change the snapshot length from the default
-C   With -w, limit the capture file size, and begin a new file when it is
    exceeded
-W   With -C, limit the number of capture files created, and begin
    overwriting and rotating when necessary
-D   List available adapters (WinDump only)
```

You can also look at TCPdump command line usage ok. It might there could be one or two minor differences depending on how it is implemented, but then in general this has options like writing into a file and then what should be the file size etcetera.

(Refer Slide Time: 13:03)



wireshark

- Open source GUI
- Captures – shows in real time and saves in a file
- Filters – easy filtering with many options
- Analyzes – powerful protocol analyzer
- Includes tshark
  - Command line network protocol analysis tool
  - Reads and saves files in same format
  - Ex:

```
# tshark -i eth0 -w test.pcap 'not port 22'
Capturing on eth0
235
```

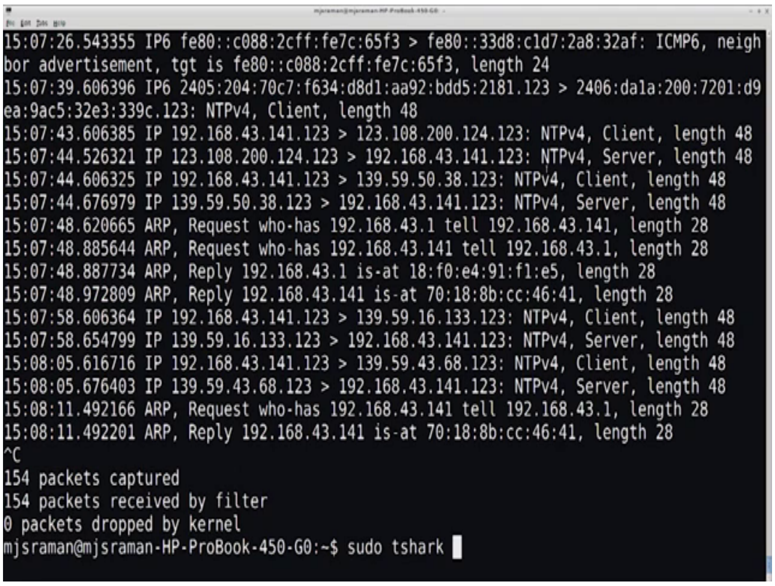
- Includes dumpcap
  - Especially designed for packet capturing
  - Ex:

```
$ dumpcap -i eth0 -w test.pcap 'not port 22'
File: test.pcap
Packets: 12
Packets dropped: 0
```

The next tool if you are not very familiar with the using command lines then you could actually use the open source GUI which is the wire shark ok. The advantage of using wire shark is that it provides a nice GUI, but it is also a command line option available for wire shark ok.

So, what we can do is that it can actually capture the packet, it can filter the packet then it can analyze the advantage of using wireshark is that you will be able to to group the packets together we told you that much more easily than then using any command line tools. But of course so wireshark also includes the command line called t shark and the similar to TCPdump I mean the usage is similar to TCPdump ok. So, let us take a look at the usage now. So, let us stop this sometime and then let us try to look at the usage ok.

(Refer Slide Time: 14:08)



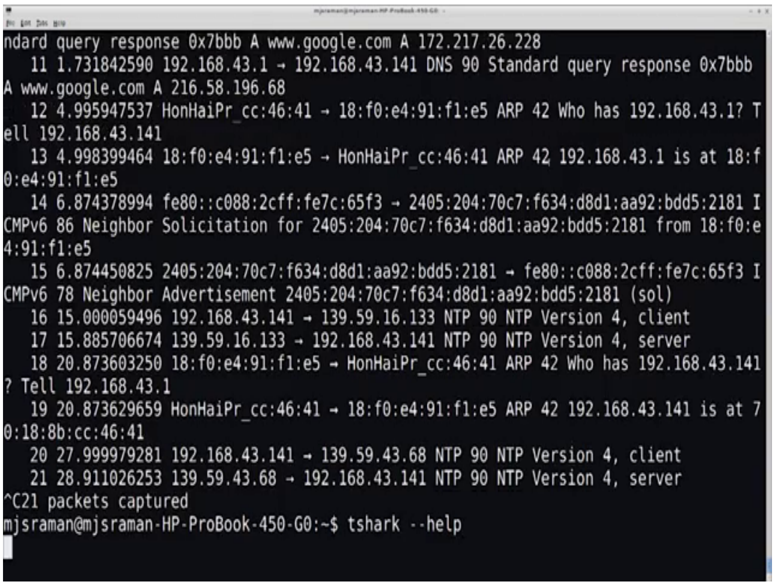
```
15:07:26.543355 IP6 fe80::c088:2cff:fe7c:65f3 > fe80::33d8:c1d7:2a8:32af: ICMP6, neigh
bor advertisement, tgt is fe80::c088:2cff:fe7c:65f3, length 24
15:07:39.606396 IP6 2405:204:70c7:f634:d8d1:aa92:bdd5:2181.123 > 2406:da1a:200:7201:d9
ea:9ac5:32e3:339c.123: NTPv4, Client, length 48
15:07:43.606385 IP 192.168.43.141.123 > 123.108.200.124.123: NTPv4, Client, length 48
15:07:44.526321 IP 123.108.200.124.123 > 192.168.43.141.123: NTPv4, Server, length 48
15:07:44.606325 IP 192.168.43.141.123 > 139.59.50.38.123: NTPv4, Client, length 48
15:07:44.676979 IP 139.59.50.38.123 > 192.168.43.141.123: NTPv4, Server, length 48
15:07:48.620665 ARP, Request who-has 192.168.43.1 tell 192.168.43.141, length 28
15:07:48.885644 ARP, Request who-has 192.168.43.141 tell 192.168.43.1, length 28
15:07:48.887734 ARP, Reply 192.168.43.1 is-at 18:f0:e4:91:f1:e5, length 28
15:07:48.972809 ARP, Reply 192.168.43.141 is-at 70:18:8b:cc:46:41, length 28
15:07:58.606364 IP 192.168.43.141.123 > 139.59.16.133.123: NTPv4, Client, length 48
15:07:58.654799 IP 139.59.16.133.123 > 192.168.43.141.123: NTPv4, Server, length 48
15:08:05.616716 IP 192.168.43.141.123 > 139.59.43.68.123: NTPv4, Client, length 48
15:08:05.676403 IP 139.59.43.68.123 > 192.168.43.141.123: NTPv4, Server, length 48
15:08:11.492166 ARP, Request who-has 192.168.43.141 tell 192.168.43.1, length 28
15:08:11.492201 ARP, Reply 192.168.43.141 is-at 70:18:8b:cc:46:41, length 28
^C
154 packets captured
154 packets received by filter
0 packets dropped by kernel
mjsraman@mjsraman-HP-ProBook-450-G0:~$ sudo tshark
```

As usual I type and by the way if you need to install these tools before you start using them. If you are using lubuntu or ubuntu then its sudo app get install and then you give the tool name, otherwise you find out the appropriate way to install ok. So, here we can use this tools I can type the same command line like, but in this case I am going to capture it on the wireless interface I want to write it into test dot pcap and I want to capture all those files which are not for port. So, it actually wants you what you are supposed to do and then say it says the file which capture could be saved as, but could not be opened because its permission denied and so on.

So, what we can do is in that case I just want to write into a file I just want to display it ok. So, you can start capturing it and what I can do this I can just from this again I start doing the same command. So, now, if you see this it starts capturing the packets; So, with similar to what we had done earlier ok.

So, what I want to tell you the usage of all these tools are similar ok. So, you just have to look into the command line. So, if you want more information on these tools and the command line just go ahead and so you can do t shark minus minus help of it gives you all the command line options.

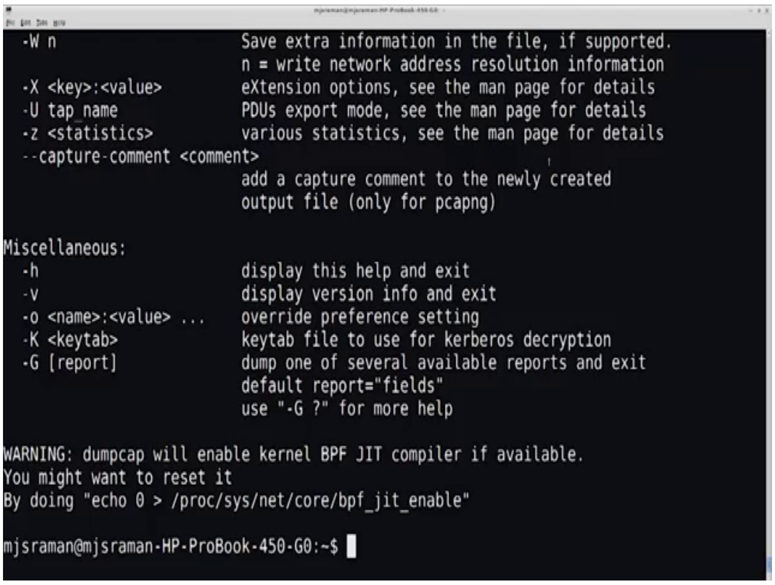
(Refer Slide Time: 15:37)



```
ndard query response 0x7bbb A www.google.com A 172.217.26.228
   11 1.731842590 192.168.43.1 → 192.168.43.141 DNS 90 Standard query response 0x7bbb
A www.google.com A 216.58.196.68
   12 4.995947537 HonHaiPr_cc:46:41 → 18:f0:e4:91:f1:e5 ARP 42 Who has 192.168.43.1? T
ell 192.168.43.141
   13 4.998399464 18:f0:e4:91:f1:e5 → HonHaiPr_cc:46:41 ARP 42 192.168.43.1 is at 18:f
0:e4:91:f1:e5
   14 6.874378994 fe80::c088:2cff:fe7c:65f3 → 2405:204:70c7:f634:d8d1:aa92:bdd5:2181 I
CMPv6 86 Neighbor Solicitation for 2405:204:70c7:f634:d8d1:aa92:bdd5:2181 from 18:f0:e
4:91:f1:e5
   15 6.874450825 2405:204:70c7:f634:d8d1:aa92:bdd5:2181 → fe80::c088:2cff:fe7c:65f3 I
CMPv6 78 Neighbor Advertisement 2405:204:70c7:f634:d8d1:aa92:bdd5:2181 (sol)
   16 15.000059496 192.168.43.141 → 139.59.16.133 NTP 90 NTP Version 4, client
   17 15.885706674 139.59.16.133 → 192.168.43.141 NTP 90 NTP Version 4, server
   18 20.873603250 18:f0:e4:91:f1:e5 → HonHaiPr_cc:46:41 ARP 42 Who has 192.168.43.141
? Tell 192.168.43.1
   19 20.873629659 HonHaiPr_cc:46:41 → 18:f0:e4:91:f1:e5 ARP 42 192.168.43.141 is at 7
0:18:8b:cc:46:41
   20 27.999979281 192.168.43.141 → 139.59.43.68 NTP 90 NTP Version 4, client
   21 28.911026253 139.59.43.68 → 192.168.43.141 NTP 90 NTP Version 4, server
^C21 packets captured
mjsraman@mjsraman-HP-ProBook-450-G0:~$ tshark --help
```

It takes some time to actually get used to this command lines, but I think hope you are very familiar in the in your previous courses you are done lot of shell scripting.

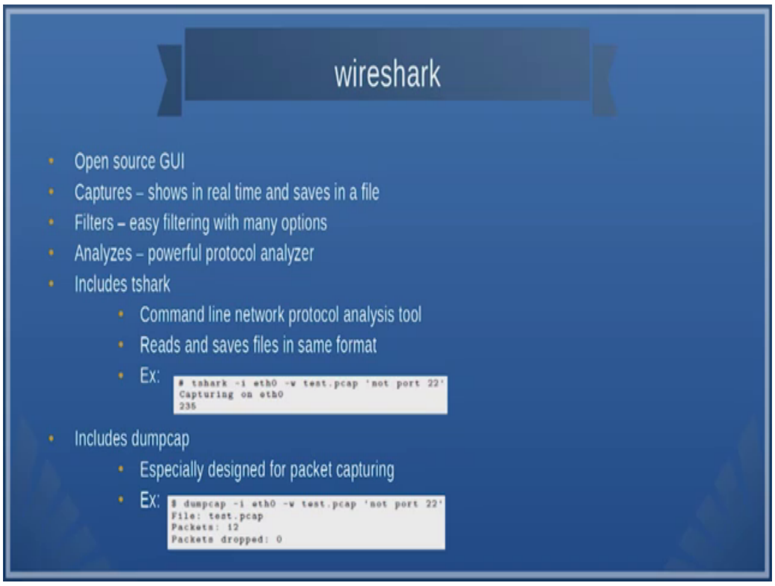
(Refer Slide Time: 15:42)



```
  -W n                     Save extra information in the file, if supported.
                           n = write network address resolution information
  -X <key>:<value>         eXtension options, see the man page for details
  -U tap_name              PDUs export mode, see the man page for details
  -z <statistics>          various statistics, see the man page for details
  --capture-comment <comment>
                           add a capture comment to the newly created
                           output file (only for pcapng)

Miscellaneous:
  -h                       display this help and exit
  -v                       display version info and exit
  -o <name>:<value> ...    override preference setting
  -K <keytab>              keytab file to use for kerberos decryption
  -G [report]              dump one of several available reports and exit
                           default report="fields"
                           use "-G ?" for more help

WARNING: dumpcap will enable kernel BPF JIT compiler if available.
You might want to reset it
By doing "echo 0 > /proc/sys/net/core/bpf_jit_enable"

mjsraman@mjsraman-HP-ProBook-450-G0:~$
```

If you remember all these things can be included into a shell script and then the whole process can be automated and that is the that is could be that is going to be our aim I mean if you want to do good forensic analysis.

(Refer Slide Time: 16:10)



wireshark

- Open source GUI
- Captures – shows in real time and saves in a file
- Filters – easy filtering with many options
- Analyzes – powerful protocol analyzer
- Includes tshark
  - Command line network protocol analysis tool
  - Reads and saves files in same format
  - Ex:

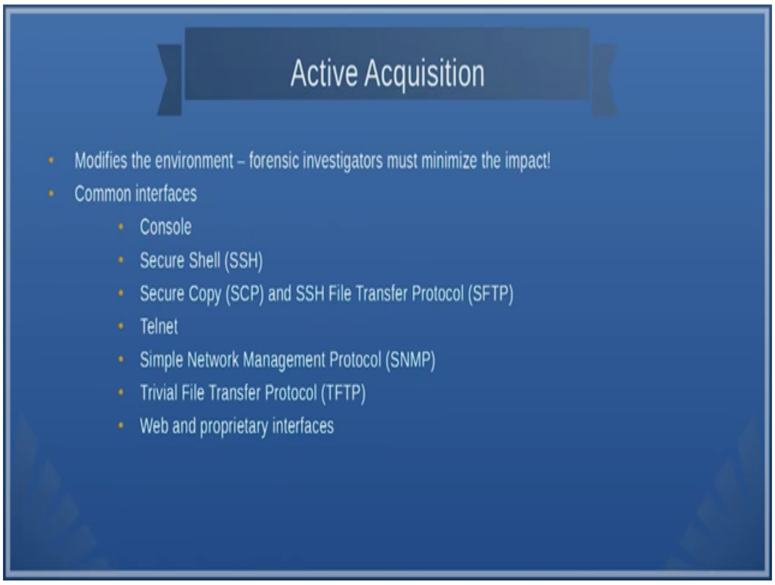
```
# tshark -i eth0 -w test.pcap 'not port 22'
Capturing on eth0
235
```

- Includes dumpcap
  - Especially designed for packet capturing
  - Ex:

```
$ dumpcap -i eth0 -w test.pcap 'not port 22'
File: test.pcap
Packets: 12
Packets dropped: 0
```

You can also use something called as dumpcap. I mean there are so many tools that are available I mean. So, you just have to, so dump cap also more or less the same kind of kind of command line ok.

(Refer Slide Time: 16:20)



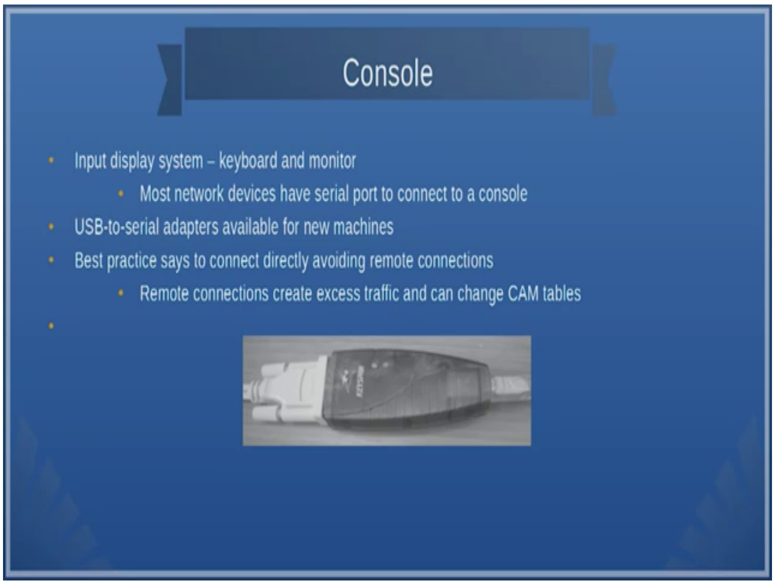
Active Acquisition

- Modifies the environment – forensic investigators must minimize the impact!
- Common interfaces
  - Console
  - Secure Shell (SSH)
  - Secure Copy (SCP) and SSH File Transfer Protocol (SFTP)
  - Telnet
  - Simple Network Management Protocol (SNMP)
  - Trivial File Transfer Protocol (TFTP)
  - Web and proprietary interfaces

So, all these use of wireshark and t shark actually gives you an active acquisition of data ok. And the problem with active acquisition of data is that it modifies the environment ok. And so what should I do if I do not want to modify the environment? Then you could use actually you need to log into the machine through various kinds of interfaces is the commonly knows interfaces are the console which you can use, the other one is secure shell if the machine is a remote machine you can use a secure shell.

Then you can copy the files using secure copy or SFTP or you can even telnet into a machine you can use SNMP to monitor a machine ok, I can use TFTP to move the files or there are some web and proprietary interfaces that are availability to you for doing this active acquisition.

(Refer Slide Time: 17:20)

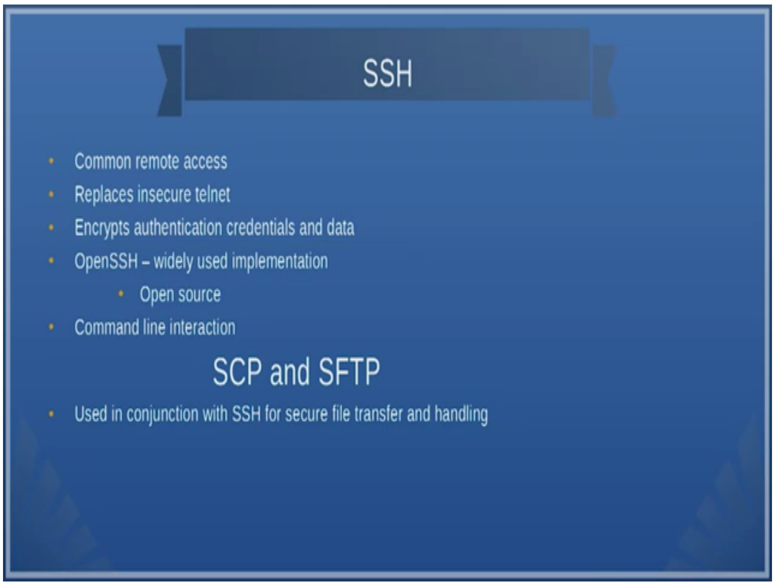


Console

- Input display system – keyboard and monitor
  - Most network devices have serial port to connect to a console
- USB-to-serial adapters available for new machines
- Best practice says to connect directly avoiding remote connections
  - Remote connections create excess traffic and can change CAM tables
-

So, you are all very familiar with console ok. So, usually you use USB kind of an adaptors I mean people who are developed embedded systems will understand how you capture data via the serial port which is the first communication mechanism for any embedded system ok. So, similarly you can actually connect to the console and then get the information.

(Refer Slide Time: 17:38)



SSH

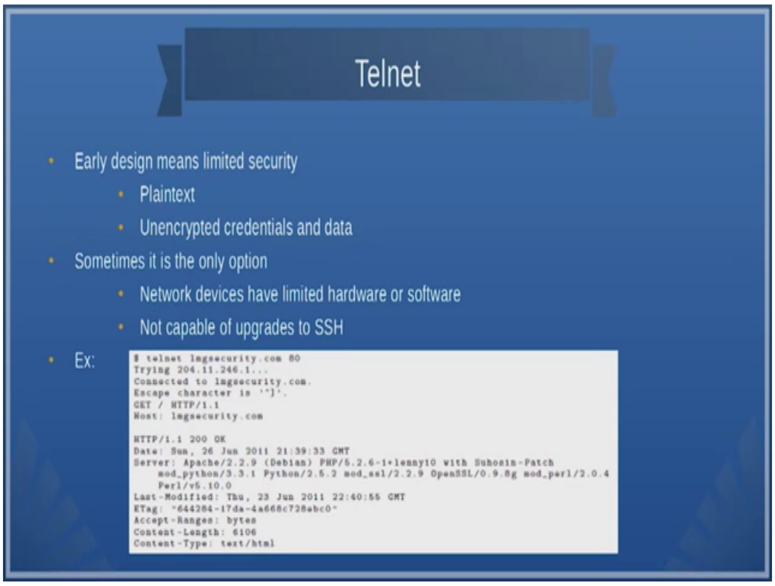
- Common remote access
- Replaces insecure telnet
- Encrypts authentication credentials and data
- OpenSSH – widely used implementation
  - Open source
- Command line interaction

SCP and SFTP

- Used in conjunction with SSH for secure file transfer and handling

The other as I told you provides SSH and SCP and SFTP or it provides actually secure way of transferring the files get the most insecure way of transferring the files are trying to find out this telnet, many of the hackers or attackers actually use this if the telnet port is open I mean they go ahead and use this port ok.

(Refer Slide Time: 17:47)



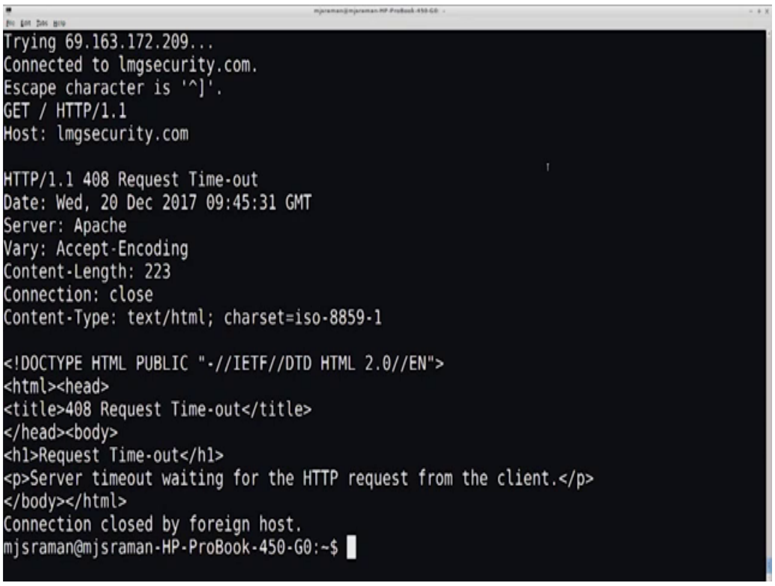
Telnet

- Early design means limited security
  - Plaintext
  - Unencrypted credentials and data
- Sometimes it is the only option
  - Network devices have limited hardware or software
  - Not capable of upgrades to SSH
- Ex:

```
$ telnet lmgsecurity.com 80
Trying 204.11.246.1...
Connected to lmgsecurity.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: lmgsecurity.com

HTTP/1.1 200 OK
Date: Sun, 26 Jun 2011 21:39:33 GMT
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny10 with Suhosin-Patch
    mod_python/3.3.1 Python/2.5.2 mod_ssl/2.2.9 OpenSSL/0.9.8g mod_perl/2.0.4
    Perl/v5.10.0
Last-Modified: Thu, 23 Jun 2011 22:40:55 GMT
ETag: "644284-17da-4a668c728ebc0"
Accept-Ranges: bytes
Content-Length: 6106
Content-Type: text/html
```

So, here you could look at some examples ok, we have we have showns some examples probably I mean we can have a demo of this, so how telnet can be used ok.

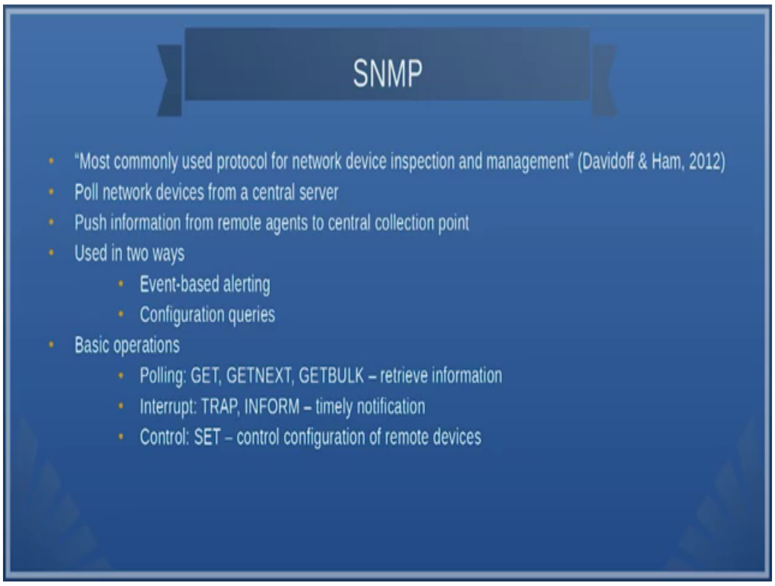
(Refer Slide Time: 18:12)



```
Trying 69.163.172.209...
Connected to lmgsecurity.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: lmgsecurity.com

HTTP/1.1 408 Request Time-out
Date: Wed, 20 Dec 2017 09:45:31 GMT
Server: Apache
Vary: Accept-Encoding
Content-Length: 223
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>408 Request Time-out</title>
</head><body>
<h1>Request Time-out</h1>
<p>Server timeout waiting for the HTTP request from the client.</p>
</body></html>
Connection closed by foreign host.
mjsraman@mjsraman-HP-ProBook-450-G0:~$
```

So, we will just go to the website ok. So, one of the things that it shows that the request is timed out and you can get information like this if you use telnet, but the point is never try to use telnet because it has limited amount of security ok. So, usually you are suppose to close the ports telnet ports.

(Refer Slide Time: 18:56)



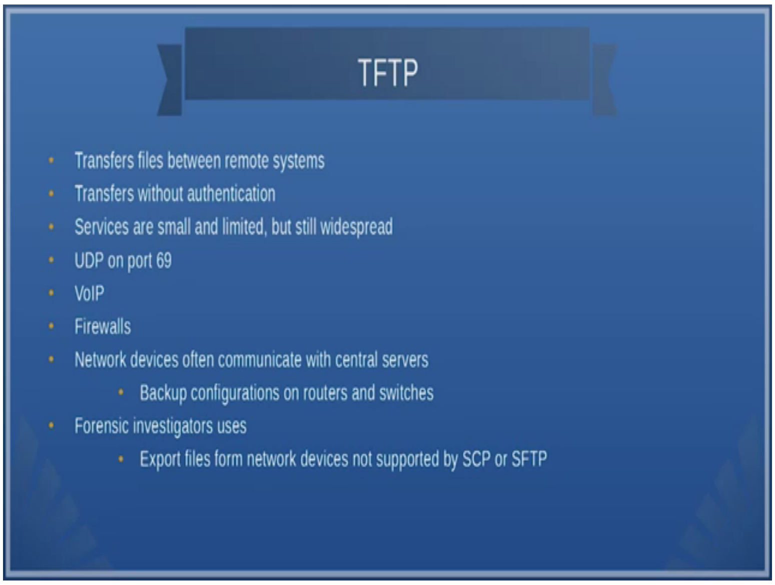
SNMP

- "Most commonly used protocol for network device inspection and management" (Davidoff & Ham, 2012)
- Poll network devices from a central server
- Push information from remote agents to central collection point
- Used in two ways
    - Event-based alerting
    - Configuration queries
- Basic operations
    - Polling: GET, GETNEXT, GETBULK – retrieve information
    - Interrupt: TRAP, INFORM – timely notification
    - Control: SET – control configuration of remote devices

So, then the other place where you can try to get information is without intruding into the network is SNMP. SNMP actually collects lot of data you can use any of the SNMP, SNMP management network management tool and then get the data.

So, essentially what SNMP does is a polls the devises and gets the data instead of actually logging into the devise and if you look say SCP, SSH, telnet etcetera you actually log into the device here, you need not to login into the device. You use a nice client which just goes into a device and gets whatever data you want to get.

(Refer Slide Time: 19:32)



TFTP

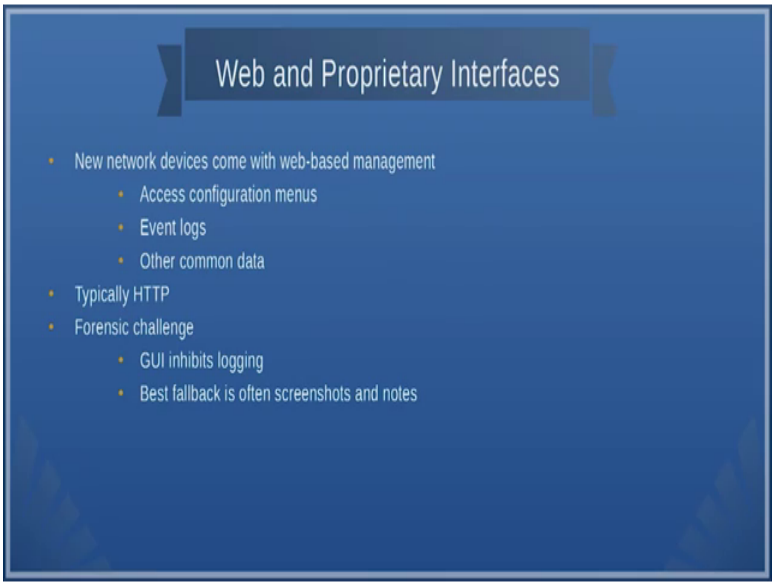
- Transfers files between remote systems
- Transfers without authentication
- Services are small and limited, but still widespread
- UDP on port 69
- VoIP
- Firewalls
- Network devices often communicate with central servers
  - Backup configurations on routers and switches
- Forensic investigators uses
  - Export files form network devices not supported by SCP or SFTP

A TFTP also tries to get login to I mean not login actually tries to get the data from the device ok. You can use TFTP to get transfer data from say firewalls and other network devises etcetera ok. So, usually forensic investigator use files that are not supported by SCP or SFTP suppose you do not have support for both of them you actually use TFTP to transfer the file. So, essentially all these tools to transfer the files from the devises.

Now, what are all the files that we are going to transfer? Actually we use those tools like t shark and TCP dump to dump the packets. So, once the packets are dumped you do not want analyze them in the same device where you are store the packets. So, you just take those packets out, bring it to server and then do your analysis. So, when you want to do that you have to use these kind of devises mechanisms for example, you can use SSH etcetera. So, TFTP is one mechanism where if you do not have SSH or SFTP you can use TFTP of course, the TFTP server should be running ok, or the client should server should be server and the client should be running in all these protocols for I mean FTP etcetera the server and client should be there in the usual it is there or you can use a SNMP even to transfer the data.

There are also some proprietary interfaces with typical use http to transfer the data ok. So, the problem is that GUI inhibits to login into the device etcetera and sometimes this proprietary interfaces you can get the screenshots etcetera from, so you log into the device you might have login to the device and the web server should be running.

(Refer Slide Time: 21:18)

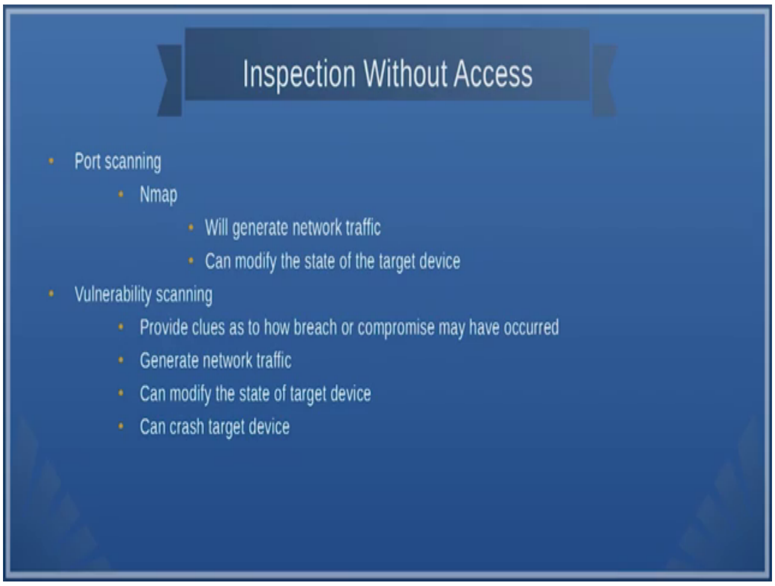


Web and Proprietary Interfaces

- New network devices come with web-based management
  - Access configuration menus
  - Event logs
  - Other common data
- Typically HTTP
- Forensic challenge
  - GUI inhibits logging
  - Best fallback is often screenshots and notes

Suppose someone close the web server you cannot access it etcetera then, and one other thing you should be careful is server closes down you should always be monitoring for all these process to be up in all these devices.

(Refer Slide Time: 21:36)

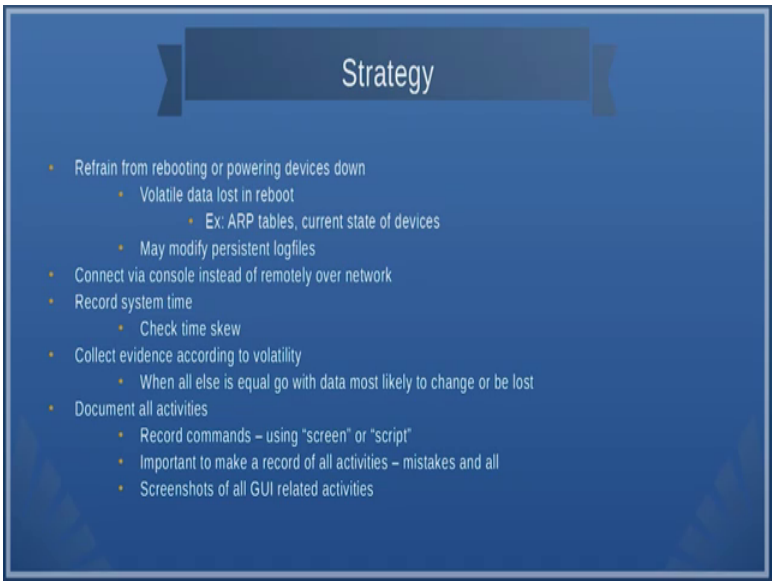


Inspection Without Access

- Port scanning
  - Nmap
    - Will generate network traffic
    - Can modify the state of the target device
- Vulnerability scanning
  - Provide clues as to how breach or compromise may have occurred
  - Generate network traffic
  - Can modify the state of target device
  - Can crash target device

Can we do inspection without access? Yes. We can we can do something known as port scanning. I think you should be very familiar with this (Refer Time: 21:41) talked about this port scanning, then you can also look at vulnerability scans and other thing like penetration testing etcetera which , but some of these vulnerability scanning can crash a device and, but these things you can do inspection without access. For example, SSH and all you need access to other side here you do not access to other side you just send the data and get those values ok.

(Refer Slide Time: 22:13)



Strategy

- Refrain from rebooting or powering devices down
  - Volatile data lost in reboot
    - Ex: ARP tables, current state of devices
  - May modify persistent logfiles
- Connect via console instead of remotely over network
- Record system time
  - Check time skew
- Collect evidence according to volatility
  - When all else is equal go with data most likely to change or be lost
- Document all activities
  - Record commands – using "screen" or "script"
  - Important to make a record of all activities – mistakes and all
  - Screenshots of all GUI related activities

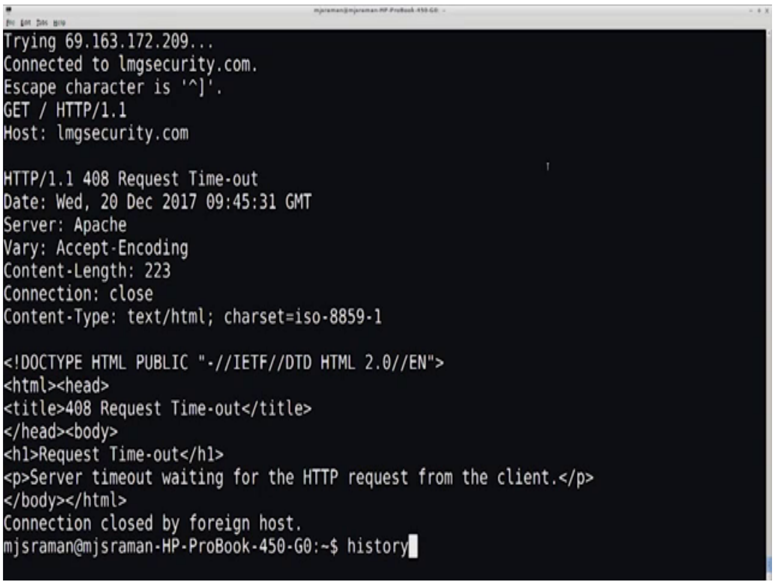
So, what should be your strategy as a network forensic specialist ok? First refrain from rebooting or powering devices down because once you do that you lose the data especially the volatile data, for example, the ARP tables, the currents state of the devices, all these things go off ok. In some devices if you reboot the log files goes off one of the example is that temp in the Linux machines the slash tmp directory actually is cleaned up during every boot and its only a scratch pad kind of directories.

So, similarly all these devices use some kind of scratch pad locations and these are all the location that might go off if you switch off and switch on the device therefore, you have to collect these scratchpad memories or scratch pad locations and then try to do the analysis ok. If you are network is remote then you have to you have know other option, but to connect a console or a SSH etcetera ok. One of the things you should look at is the record the systems the system time.

See these devices could be a different time zones one they may not be sink in time unless a you something like a NTP network time protocol and then sink their timings. So, you should also look at the times q ok. So, you cannot have some event that are happen before and some event that are happened after because we do the then the correlation becomes extremely difficult ok. And you should able to collect evidence according to the volatility ok. One of the things is when all else is equal go with the data that is more likely to change or be lost ok, that should be collected as early as possible ok.

And most important of all the activity that you should do is that you could document all the activities ok, record the commands there are lot of tools that you can use record the commands for example, one of the tools that you produce is history. So, if you look at this I type history I will be able to see what are all the commands that I type before ok.

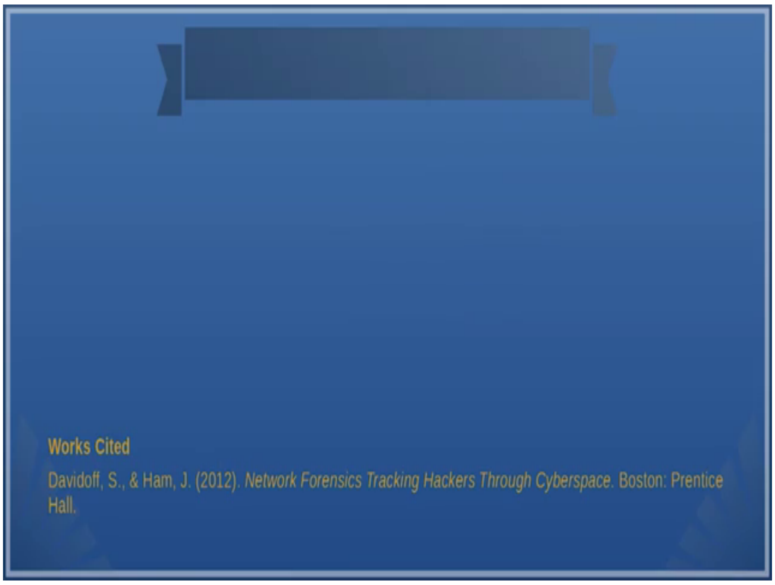
(Refer Slide Time: 24:10)



```
Trying 69.163.172.209...
Connected to lmgsecurity.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: lmgsecurity.com

HTTP/1.1 408 Request Time-out
Date: Wed, 20 Dec 2017 09:45:31 GMT
Server: Apache
Vary: Accept-Encoding
Content-Length: 223
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>408 Request Time-out</title>
</head><body>
<h1>Request Time-out</h1>
<p>Server timeout waiting for the HTTP request from the client.</p>
</body></html>
Connection closed by foreign host.
mjsraman@mjsraman-HP-ProBook-450-G0:~$ history
```

So, these kind of tools or and could be very important. One more thing with this history sometimes this it only stores it actually rolls back. So, it may be able to store everything. So, you need to be very careful and these kind of volatile data you should be able to take out very easily and rest of the data I think you can collect at your own (Refer Time: 24:35), but that does not mean that you collect it very late because an attacker if we gets accessed to these kind of data you can even modify those data, ok.

(Refer Slide Time: 24:48)



Works Cited

Davidoff, S., & Ham, J. (2012). *Network Forensics Tracking Hackers Through Cyberspace.* Boston: Prentice Hall.

Good reference for this is the Davidoff and Ham book which we recommend for you to read, ok. So, it covers two chapters, whatever we have discussed covers two chapters. And in the next session we will take a look at wireshark just to see how to apply packet filtering etcetera.

Thank you very much.