

# Sniffing

## Module 07

Unmask the **Invisible Hacker**.



# Module Objectives

- Overview of Sniffing Concepts
- Understanding MAC Attacks
- Understanding DHCP Attacks
- Understanding ARP Poisoning
- Understanding MAC Spoofing Attacks



- Understanding DNS poisoning
- Sniffing Tools
- Sniffing Countermeasures
- Understanding Various Techniques to Detect Sniffing
- Overview of Sniffing Pen Testing



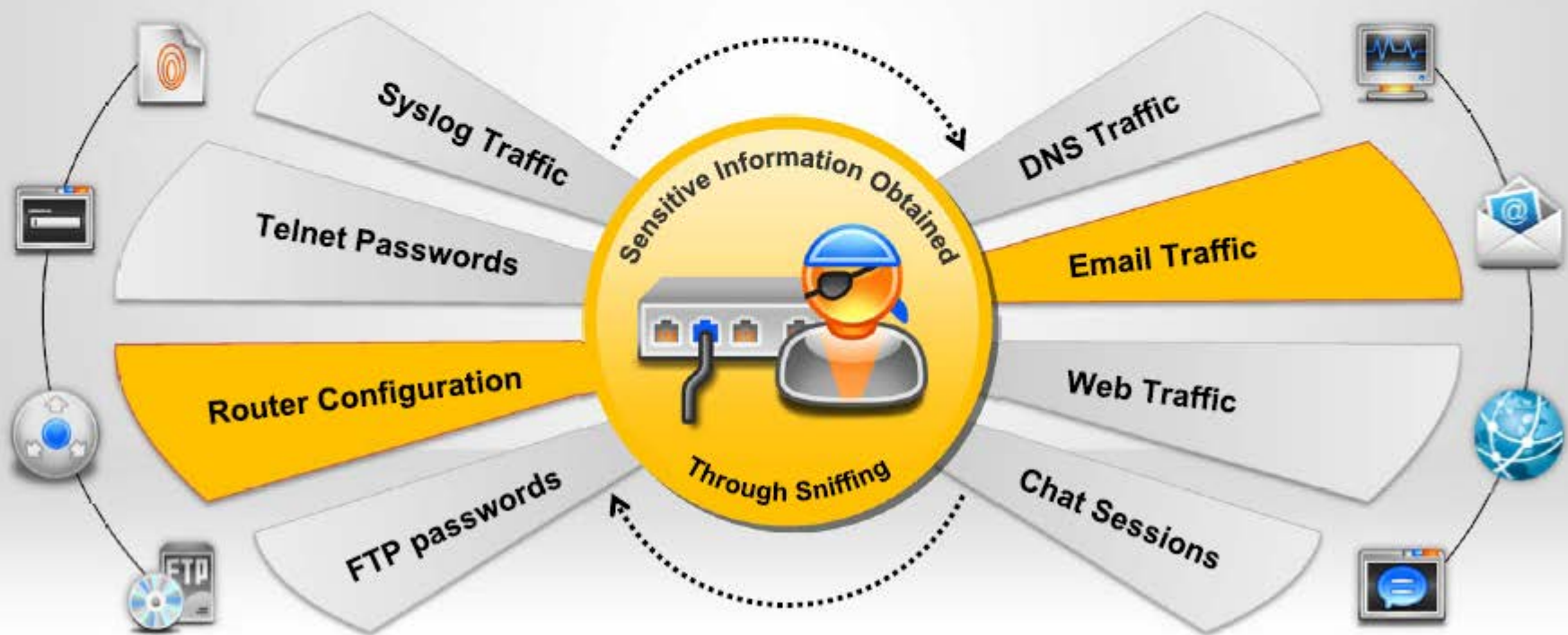
# Module Flow



# Network Sniffing and Threats

- Sniffing is a process of monitoring and **capturing all data packets** passing through a given network using sniffing tools
- It is a form of **wiretap** applied to computer networks

- Many enterprises' **switch ports** are open
- Anyone in the same physical location can plug into the network using an **Ethernet cable**



# How a Sniffer Works

## Promiscuous Mode

Sniffer turns the NIC of a system to the **promiscuous mode** so that it listens to all the data transmitted on its segment



A sniffer can constantly monitor all the network traffic to a computer through the NIC by **decoding the information** encapsulated in the data packet

## Decode Information

# Types of Sniffing: **Passive Sniffing**

01

**Passive sniffing** means sniffing through a **hub**, on a hub the traffic is sent to all ports

02

It involves only monitoring of the packets sent by others without sending **any additional data packets** in the network traffic

03

In a network that use hubs to connect systems, all **hosts on the network** can see all traffic therefore attacker can easily capture traffic going through the hub

04

Hub usage is out-dated today. Most modern networks use **switches**



**Note:** Passive sniffing provides significant stealth advantages over active sniffing

# Types of Sniffing: Active Sniffing

- Active sniffing is used to sniff a **switch-based network**
- Active sniffing involves **injecting address resolution packets (ARP)** into the network to flood the switch's Content Addressable Memory (CAM) table, CAM keeps track of which host is connected to which port



## Active Sniffing Techniques

**1** MAC Flooding



**4** DHCP Attacks

**2** DNS Poisoning



**5** Switch Port Stealing

**3** ARP Poisoning



**6** Spoofing Attack

# How an Attacker Hacks the Network Using Sniffers

An attacker connects his laptop to a switch port



1

He runs discovery tools to learn about network topology



2

He identifies victim's machine to target his attacks



3

He poisons the victim machine by using ARP spoofing techniques



4

The traffic destined for the victim machine is redirected to the attacker



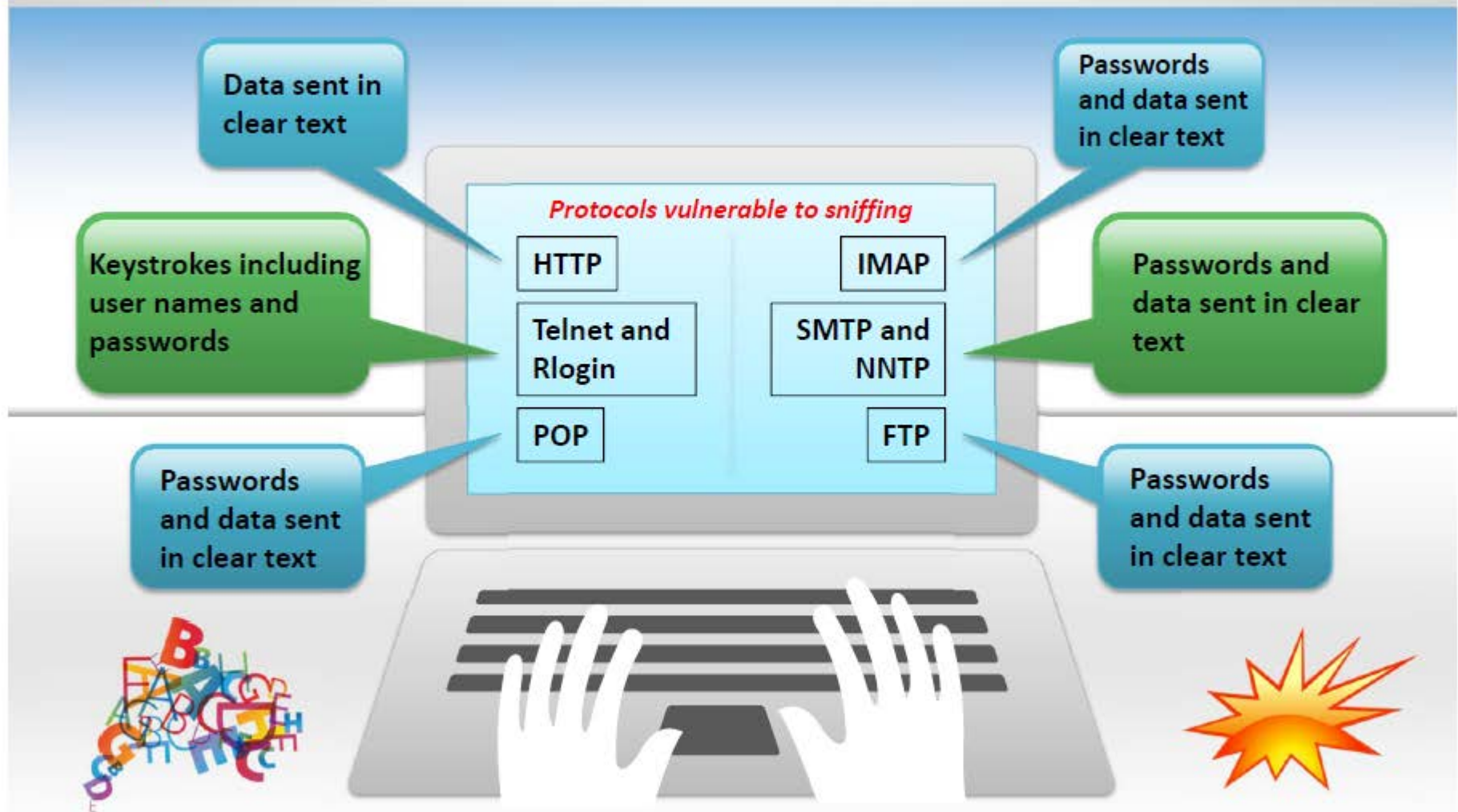
5

The hacker extracts passwords and sensitive data from the redirected traffic



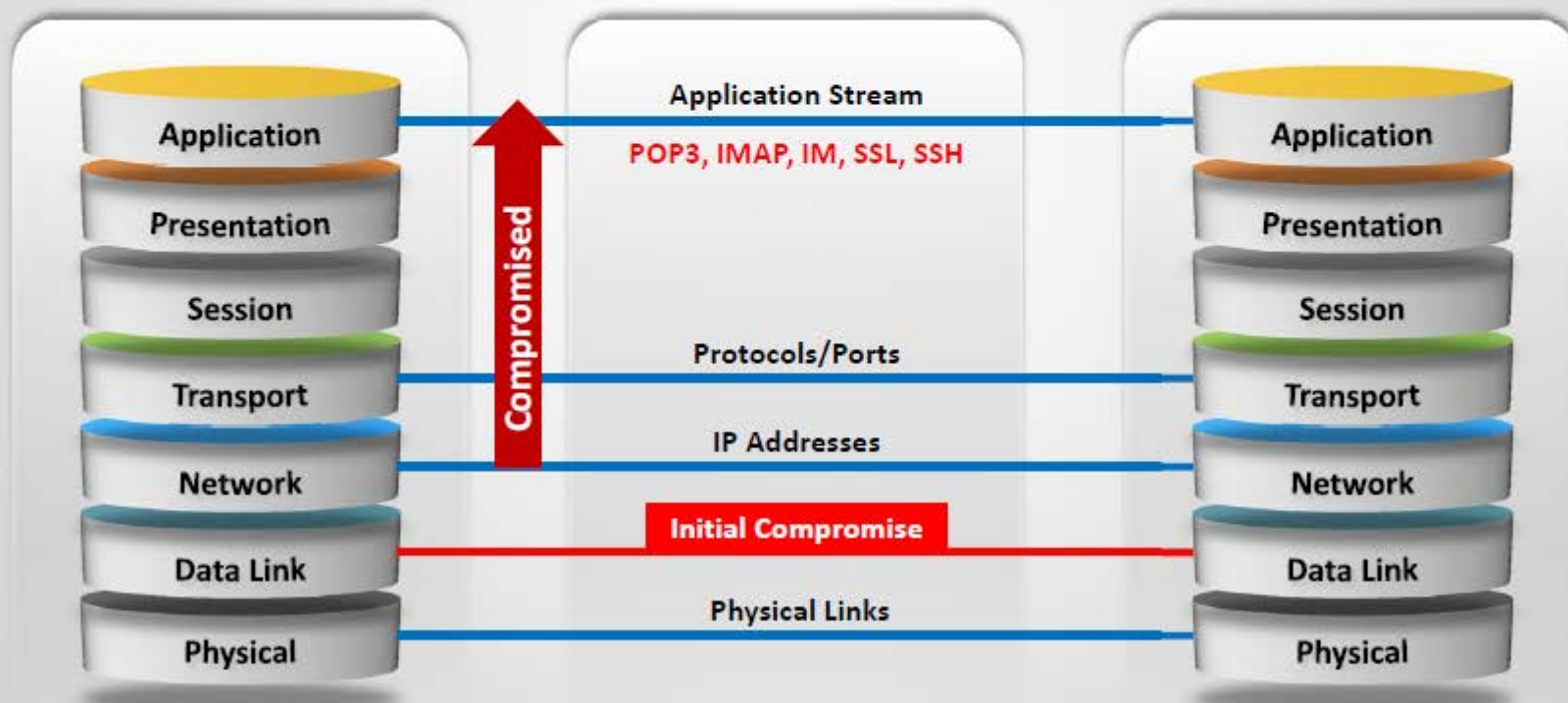
6

# Protocols Vulnerable to Sniffing



# Sniffing in the **Data Link Layer** of the OSI Model

- Sniffers operate at the **Data Link layer** of the OSI model
- Networking layers in the OSI model are designed to work **independently** of each other; if a sniffer sniffs data in the Data Link layer, the upper OSI layer will not be aware of the sniffing



# Hardware Protocol Analyzer



A hardware protocol analyzer is a piece of equipment that **captures signals** without altering the traffic in a cable segment



It can be used to monitor network usage and identify **malicious network traffic** generated by hacking software installed in the network



It captures a data packet, decodes it, and analyzes its content according to certain **predetermined rules**



It allows attacker to see individual **data bytes** of each packet passing through the cable

# Hardware Protocol Analyzers



Keysight N2X N5540A



Keysight E2960B



RADCOM PrismLite Protocol Analyzer



RADCOM Prism UltraLite  
Protocol Analyzer



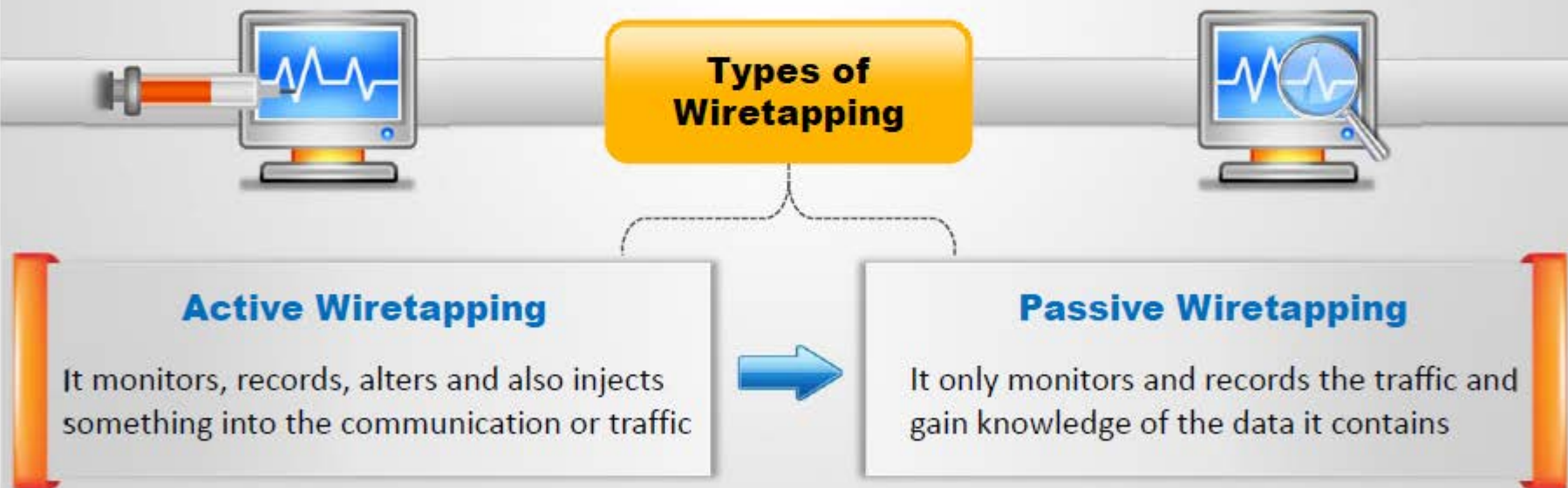
FLUKE Networks OptiView® XG  
Network Analyzer



FLUKE Networks OneTouch™  
AT Network Assistant

# Wiretapping

- 1 Wiretapping is the process of monitoring **telephone** and **Internet** conversations by a third party
- 2 Attackers **connect a listening device** (hardware, software, or a combination of both) to the circuit carrying information between two phones or hosts on the Internet
- 3 It allows an attacker to **monitor**, **intercept**, **access**, and **record information** contained in a data flow in a communication system

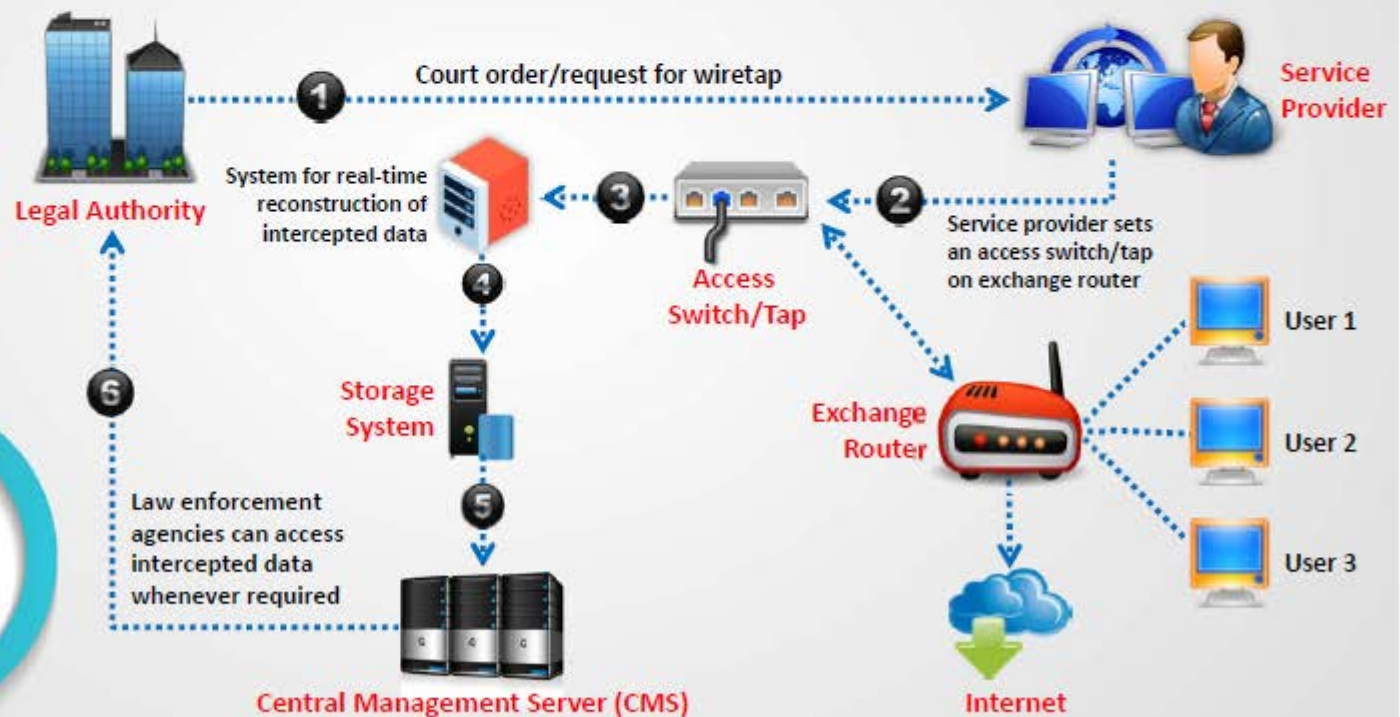


**Note:** Wiretapping without a warrant or the consent of the concerned person is a criminal offense in most countries

# Lawful Interception

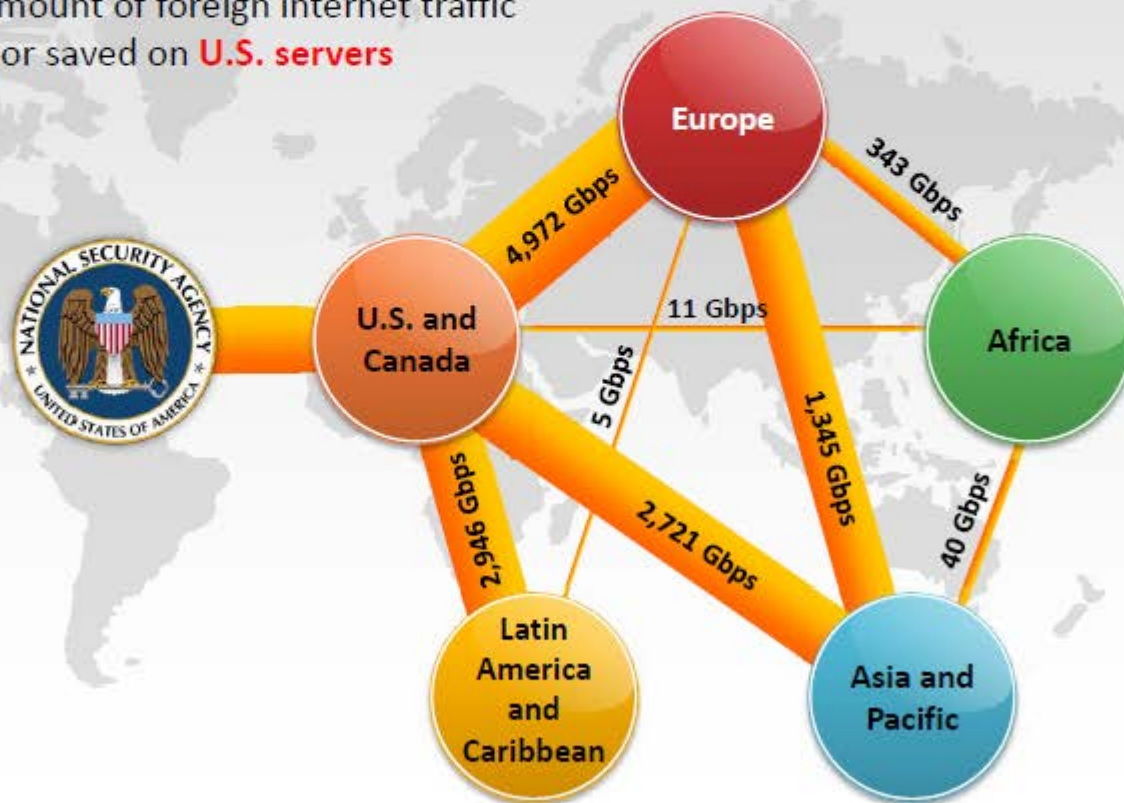


Lawful interception refers to legally **intercepting data communication** between two end points for surveillance on the traditional telecommunications, VoIP, data, and multiservice networks

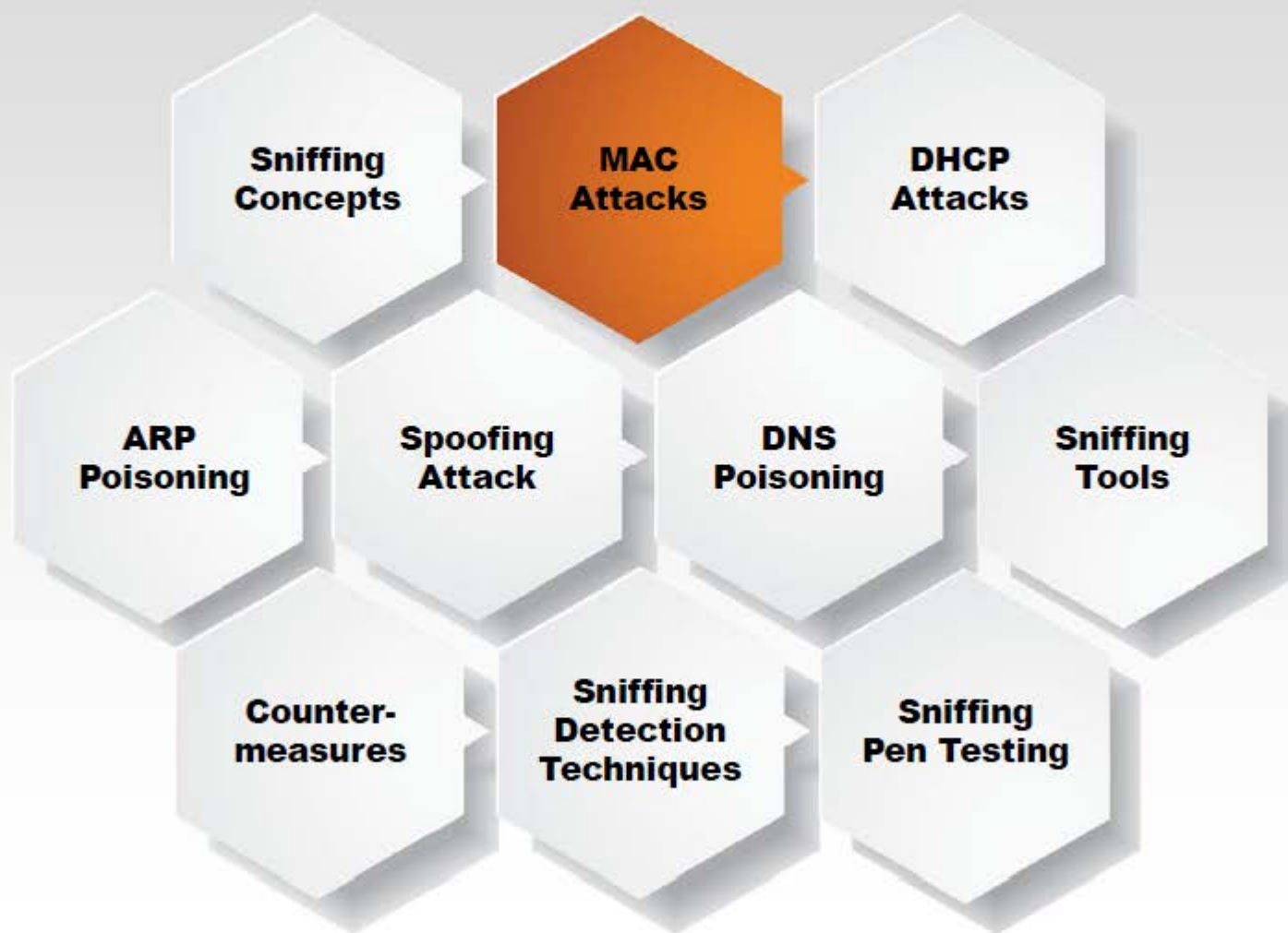


# Wiretapping Case Study: **PRISM**

- PRISM stands for "**P**lanning **T**ool for **R**esource **I**ntegration, **S**ynchronization, and **M**anagement," and is a "**data tool**" designed to collect and process "**foreign intelligence**" that passes through American servers
- NSA wiretaps a huge amount of foreign internet traffic that is routed through or saved on **U.S. servers**



# Module Flow



# MAC Address/CAM Table

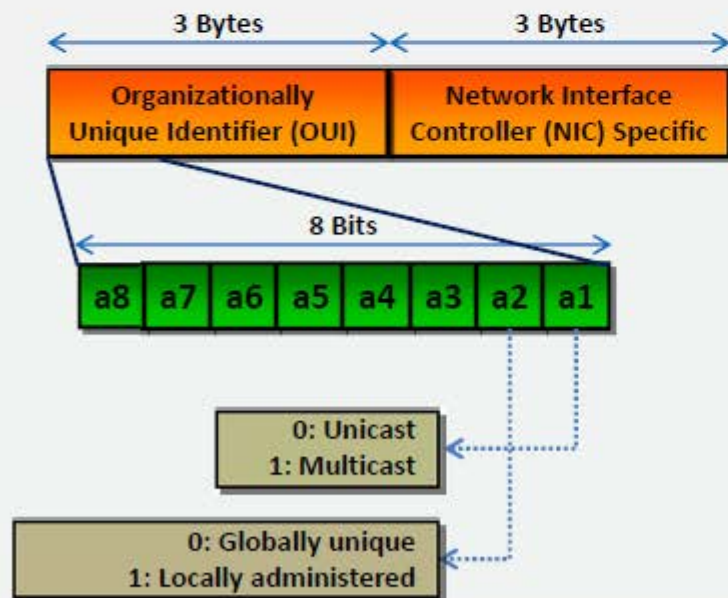


Each switch has a **fixed size dynamic Content Addressable Memory (CAM) table**



The CAM table **stores information** such as MAC addresses available on physical ports with their associated VLAN parameters

## MAC Address



## CAM Table

vlan	MAC Add	Type	Learn	Age	Ports
255	00d3.ad34.123g	Dyna mic	Yes	0	Gi5/2
5	as23.df45.45t6	Dyna mic	Yes	0	Gi2/5
5	er23.23er.t5e3	Dyna mic	Yes	0	Gi1/6

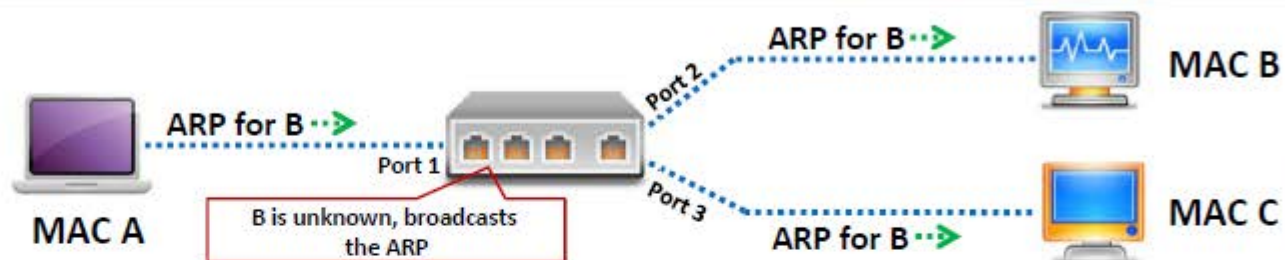


# How CAM Works

1

MAC	PORT
A	1
C	3

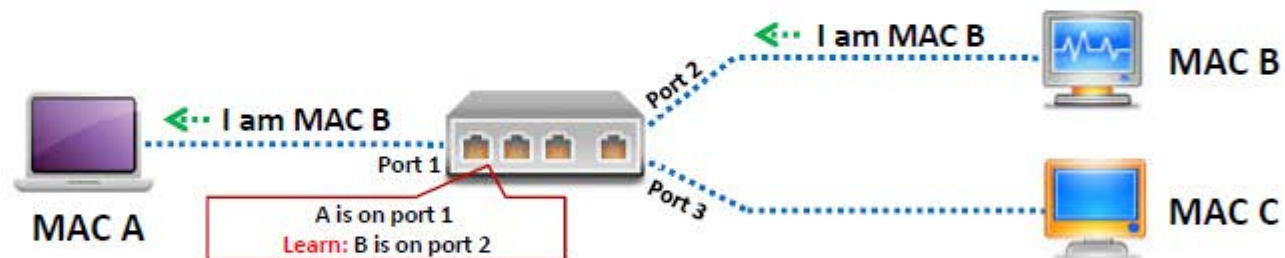
CAM Table



2

MAC	PORT
A	1
B	2
C	3

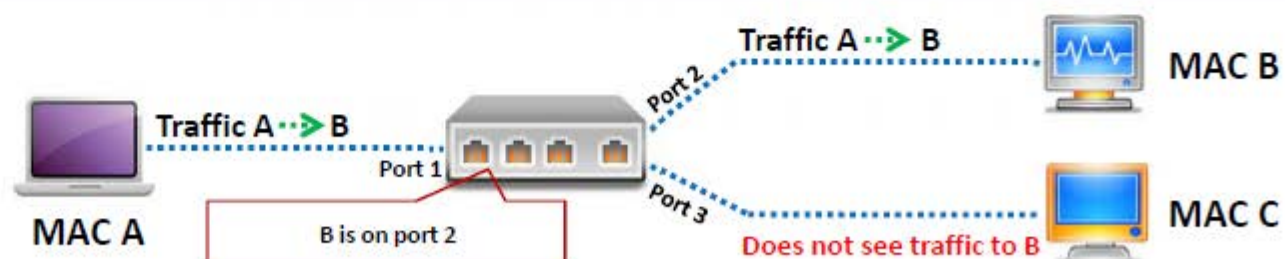
CAM Table



3

MAC	PORT
A	1
B	2
C	3

CAM Table



# What Happens When CAM Table Is Full?



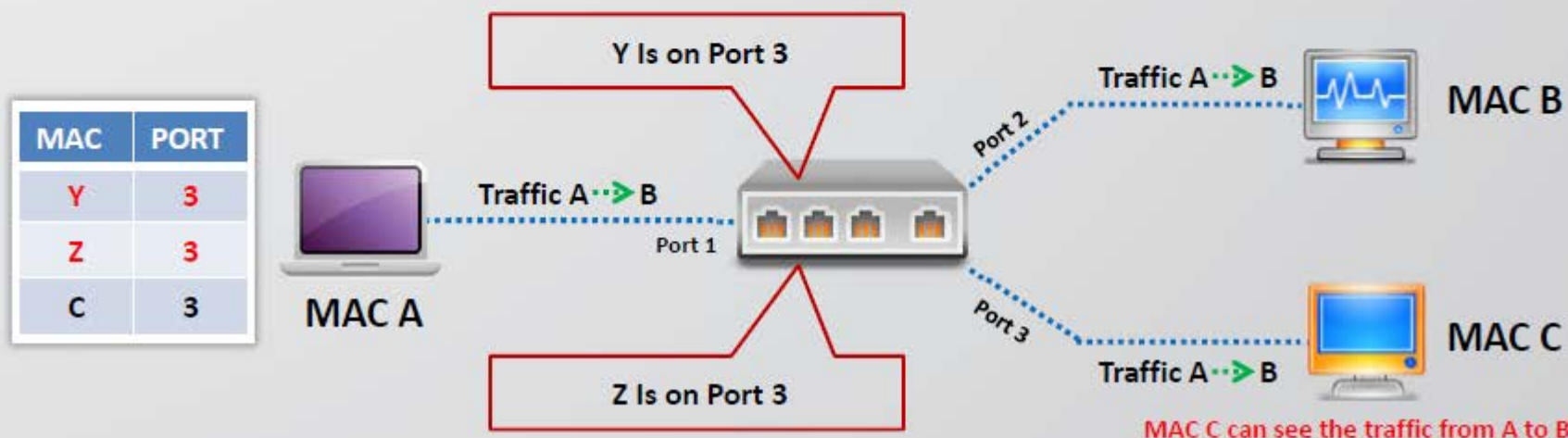
Once the CAM table on the switch is full, additional ARP request **traffic will flood every port on the switch**



This will **change the behavior of the switch** to reset to its learning mode, broadcasting on every port similar to a hub



This attack will also **fill the CAM tables of adjacent switches**

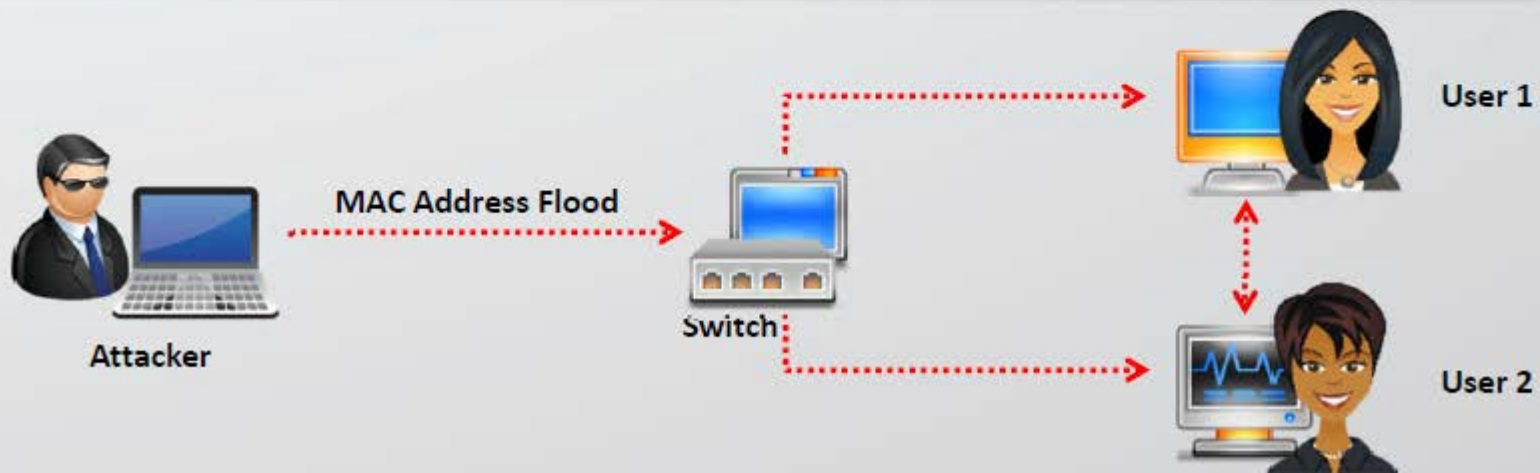


# MAC Flooding

MAC flooding involves **flooding of CAM table** with fake MAC address and IP pairs until it is full



Switch then **acts as a hub** by broadcasting packets to all machines on the network and attackers can sniff the traffic easily



# Mac Flooding Switches with **macof**



- **macof** is a Unix/Linux tool that is a part of dsniff collection
- Macof sends random **source MAC** and **IP addresses**
- This tool **floods the switch's CAM tables** (131,000 per min) by sending bogus MAC entries



## Command Prompt

```
macof -i eth1
```

```
18:b1:22:12:85:15 13:15:5a:6b:45:c4 0.0.0.0.25684 > 0.0.0.0.86254: S 2658741236:1235486715(0) win 512
12:a8:d8:15:4d:3b ab:4c:cd:5f:ad:cd 0.0.0.0.12387 > 0.0.0.0.78962: S 1238569742:782563145(0) win 512
13:3f:ab:14:25:95 66:ab:6d:4d:b2:85 0.0.0.0.45638 > 0.0.0.0.4568: S 123587152:456312589(0) win 512
a2:2f:85:12:ac:2f 12:85:2f:52:41:25 0.0.0.0.42358 > 0.0.0.0.35842: S 3256789512:3568742158(0) win 512
96:25:a3:5c:52:af 82:12:41:1d:ac:d6 0.0.0.0.45213 > 0.0.0.0.2358: S 3684125687:3256874125(0) win 512
a2:c2:b5:8c:6d:2a 5a:cc:f6:41:8d:df 0.0.0.0.12354 > 0.0.0.0.78521: S 1236542358:3698521475(0) win 512
55:42:ac:85:c5:96 a5:5f:ad:9d:12:aa 0.0.0.0.123 > 0.0.0.0.12369: S 8523695412:8523698742(0) win 512
a9:4d:4c:5a:5d:ad a4:ad:5f:4d:e9:ad 0.0.0.0.23685 > 0.0.0.0.45686: S 236854125:365145752(0) win 512
s3:e5:1a:25:2w:a3 25:35:a8:5d:af:fc 0.0.0.0.23685 > 0.0.0.0.85236: S 8623574125:3698521456(0) win 512
```

<http://monkey.org>

# Switch Port Stealing

Switch Port Stealing sniffing technique uses **MAC flooding** to sniff the packets

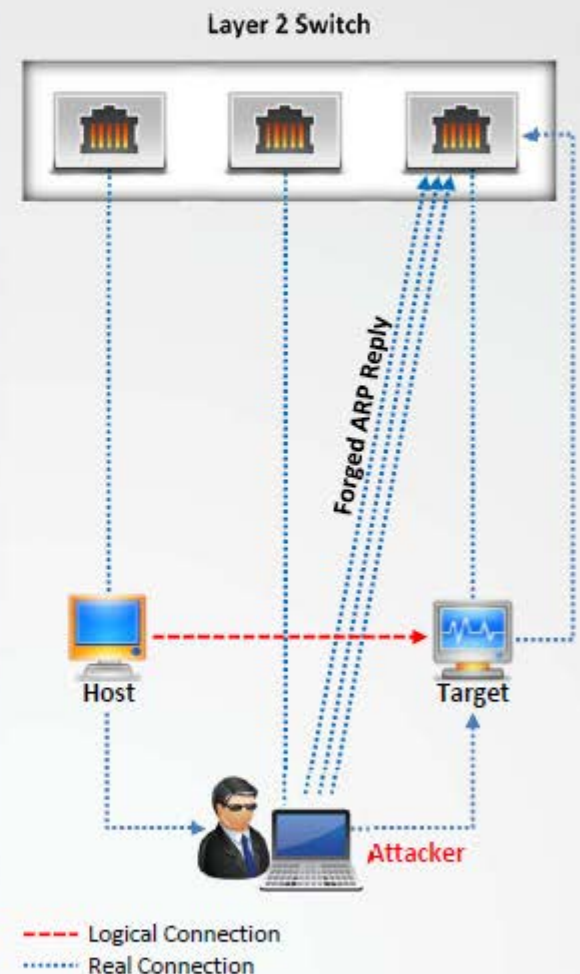
Attacker floods the switch with **forged gratuitous ARP packets** with target MAC address as source and his own MAC address as destination

A **race condition** of attacker's flooded packets and target host packets will occur and thus switch has to change his MAC address binding constantly between two different ports

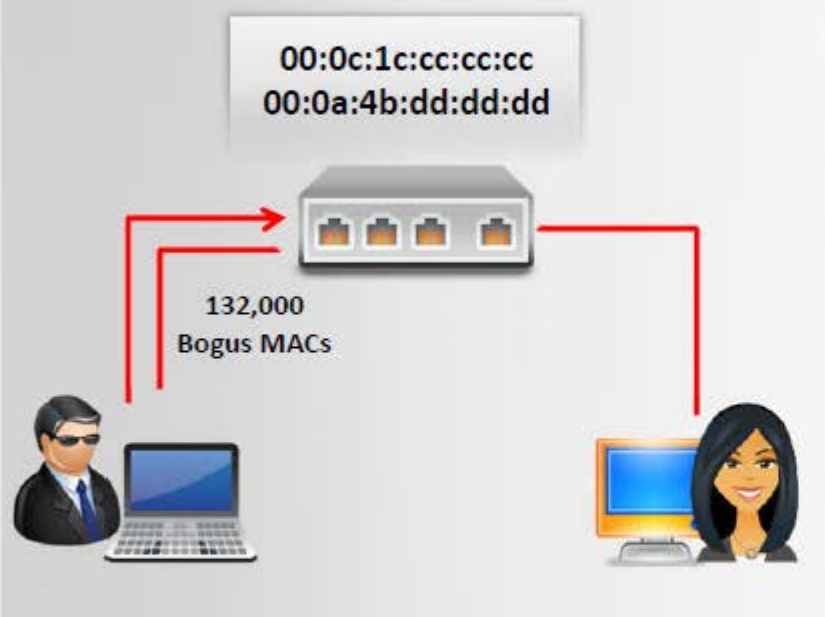
In such case if attacker is fast enough, he will be able to **direct the packets** intended for the target host toward his switch port

Attacker now manages to **steal the target host switch port** and sends ARP request to stolen switch port to discover target host's IP address

When attacker gets ARP reply, this indicates that **target host's switch port binding** has been restored and attacker can now be able to sniff the packets sent toward targeted host

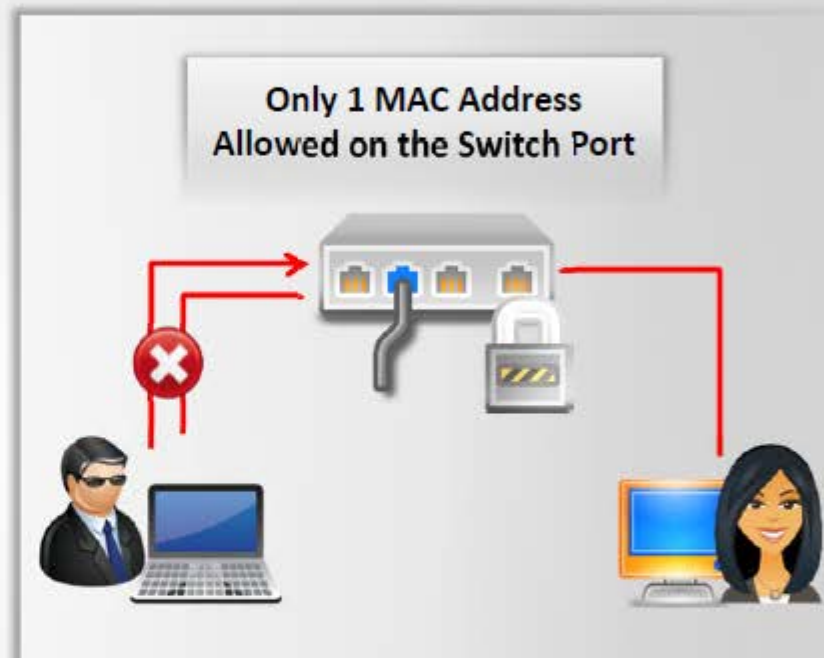


# How to Defend against **MAC Attacks**



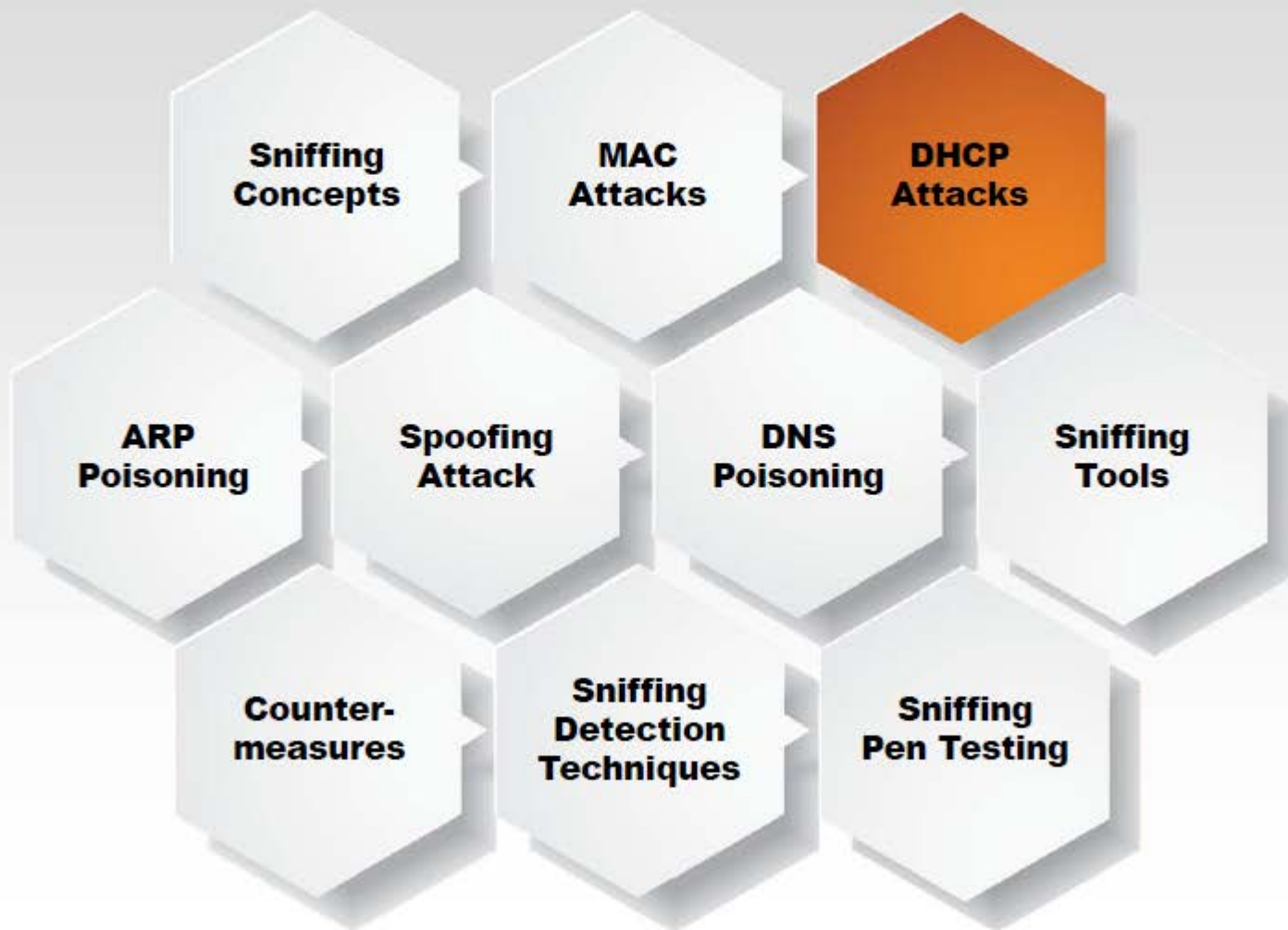
## Configuring Port Security on Cisco switch:

- switchport port-security
- switchport port-security maximum 1 vlan access
- switchport port-security violation restrict
- switchport port-security aging time 2
- switchport port-security aging type inactivity
- snmp-server enable traps port-security trap-rate 5



Port security can be used to **restrict inbound traffic** from only a selected set of MAC addresses and limit MAC flooding attack

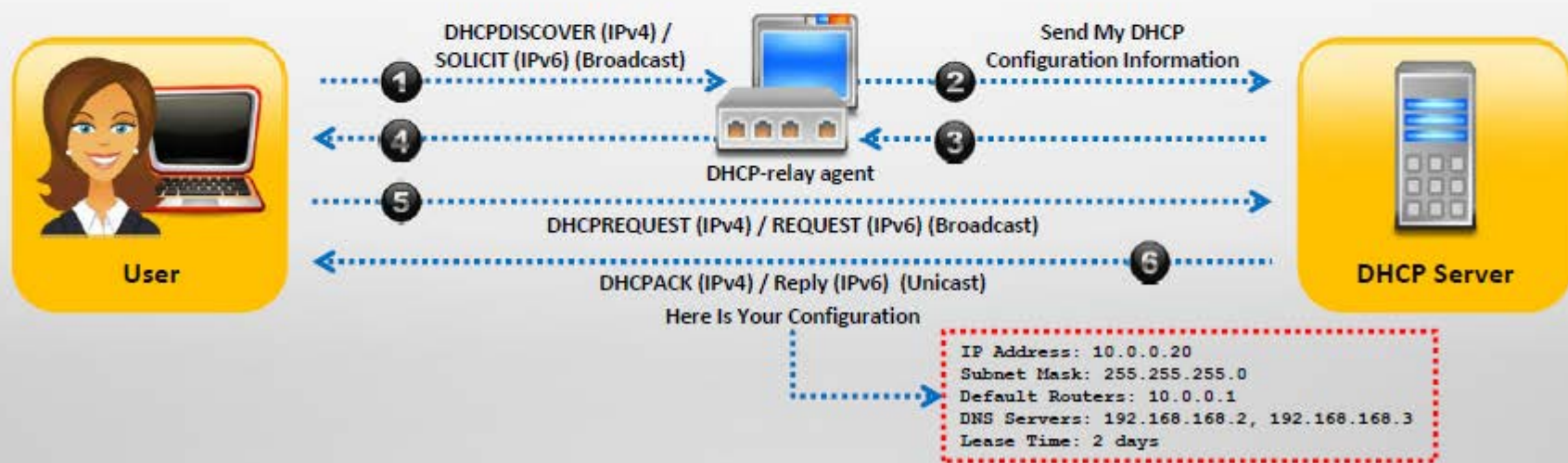
# Module Flow



# How DHCP Works

- DHCP servers maintain **TCP/IP configuration information** in a database such as valid TCP/IP configuration parameters, valid IP addresses, and duration of the lease offered by the server
- It provides address configurations to DHCP-enabled clients in the form of a **lease offer**

1. Client broadcasts **DHCPDISCOVER/SOLICIT** request asking for DHCP Configuration Information
2. DHCP-relay agent captures the client request and **unicasts** it to the DHCP servers available in the network
3. DHCP server unicasts **DHCPOFFER/ADVERTISE**, which contains client and server's MAC address
4. Relay agent broadcasts **DHCPOFFER/ADVERTISE** in the client's subnet
5. Client broadcasts **DHCPREQUEST/REQUEST** asking DHCP server to provide the DHCP configuration information
6. DHCP server sends unicast **DHCPACK/REPLY** message to the client with the IP config and information



# DHCP Request/Reply Messages

DHCPv4 Message	DHCPv6 Message	Description
DHCPDiscover	Solicit	Client broadcast to locate available DHCP servers
DHCPOffer	Advertise	Server to client in response to DHCPDISCOVER with offer of configuration parameters
DHCPRequest	Request, Confirm, Renew, Rebind	Client message to servers either (a) Requesting offered parameters, (b) Confirming correctness of previously allocated address, or (c) Extending the lease period
DHCPAck	Reply	Server to client with configuration parameters, including committed network address
DHCPRelease	Release	Client to server relinquishing network address and canceling remaining lease
DHCPDecline	Decline	Client to server indicating network address is already in use
N/A	Reconfigure	Server tells the client that it has new or updated configuration settings. The client then sends either a renew/reply or Information-request/Reply transaction to get the updated information
DHCPInform	Information Request	Client to server, asking only for local configuration parameters; client already has externally configured network address
N/A	Relay-Forward	A relay agent sends a relay-forward message to relay messages to servers, either directly or through another relay agent
N/A	Relay-Reply	A server sends a relay-reply message to a relay agent containing a message that the relay agent delivers to a client
DHCPNAK	N/A	Server to client indicating client's notion of network address is incorrect (e.g., Client has moved to new subnet) or client's lease as expired

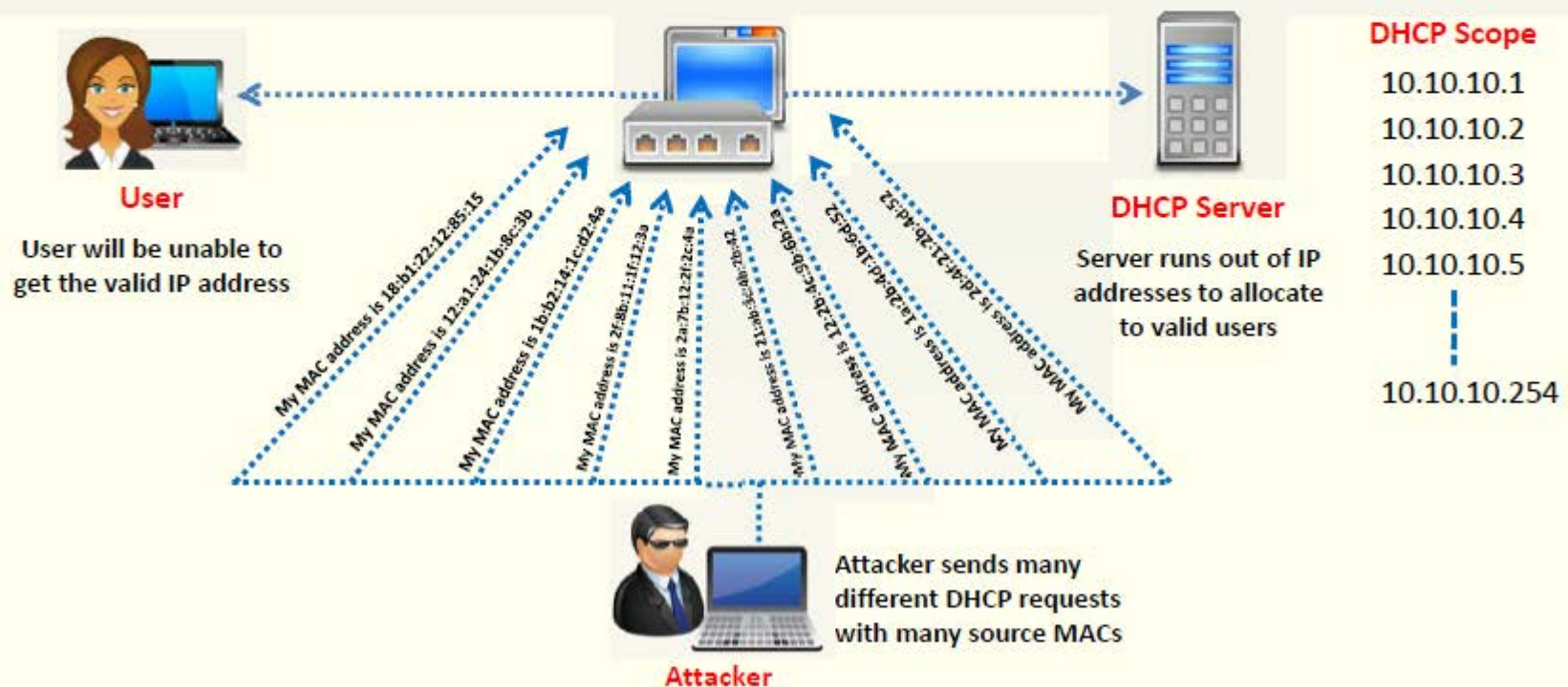
# IPv4 DHCP Packet Format



OP Code	Hardware Type	Hardware Length	HOPS
Transaction ID (XID)			
Seconds		Flags	
Client IP Address (CIADDR)			
Your IP Address (YIADDR)			
Server IP Address (SIADDR)			
Gateway IP Address (GIADDR)			
Client Hardware Address (CHADDR)—16 bytes			
Server Name (SNAME)—64 bytes			
Filename—128 bytes			
DHCP Options			

# DHCP Starvation Attack

- This is a denial-of-service (DoS) attack on the DHCP servers where attacker broadcasts **forged DHCP requests** and tries to lease all of the DHCP addresses available in the DHCP scope
- As a result legitimate user is **unable to obtain or renew an IP address** requested via DHCP, failing access to the network access



# DHCP Starvation Attack Tools

## Dhcpstarv

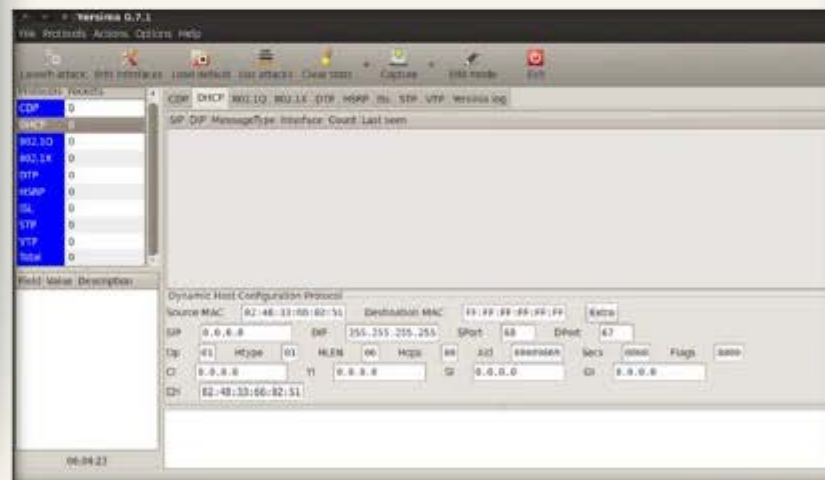
- dhcstarv implements DHCP starvation attack. It requests **DHCP leases** on specified interface, saves them, and renews on regular basis



<http://dhcstarv.sourceforge.net>

## Yersinia

- Yersinia is a network tool designed to take advantage of some **weakness** in different network protocols
- It pretends to be a solid framework for analyzing and testing the **deployed networks and systems**

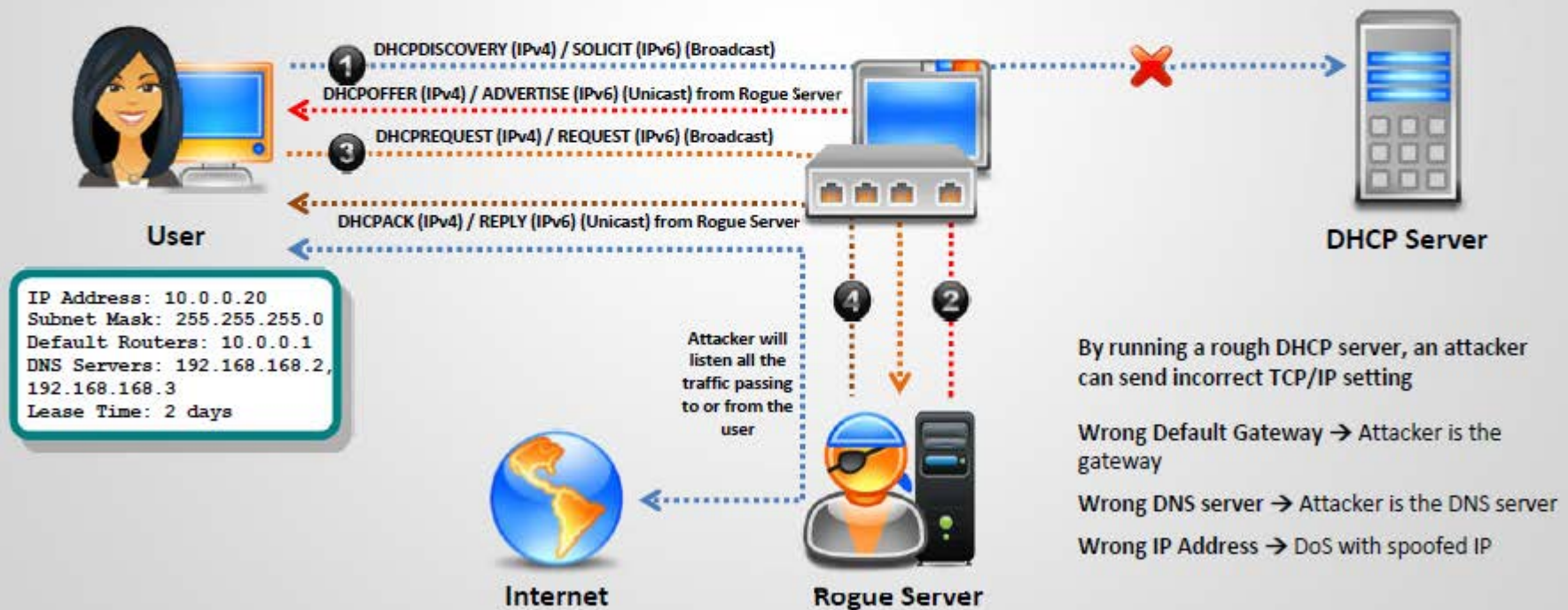


<http://www.yersinia.net>

# Rogue DHCP Server Attack

Attacker sets **rogue DHCP server** in the network and responds to DHCP requests with bogus IP addresses; this results in compromised network access

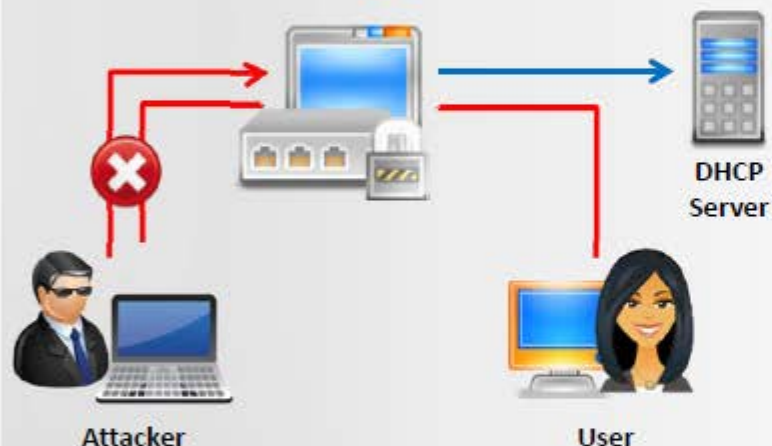
This attack works in conjunction with the DHCP Starvation attack; attacker sends **TCP/IP setting** to the user after knocking him/her out from the genuine DHCP server



# How to Defend Against DHCP Starvation and Rogue Server Attack

Enable **port security** to defend against DHCP starvation attack

- Configuring MAC limit on switch's edge ports drops the packets from further MACs once the limit is reached



## IOS Switch Commands

- `switchport port-security`
- `switchport port-security maximum 1`
- `switchport port-security violation restrict`
- `switchport port-security aging time 2`
- `switchport port-security aging type inactivity`

Enable **DHCP snooping** that allows switch to accept DHCP transaction coming only from a trusted port

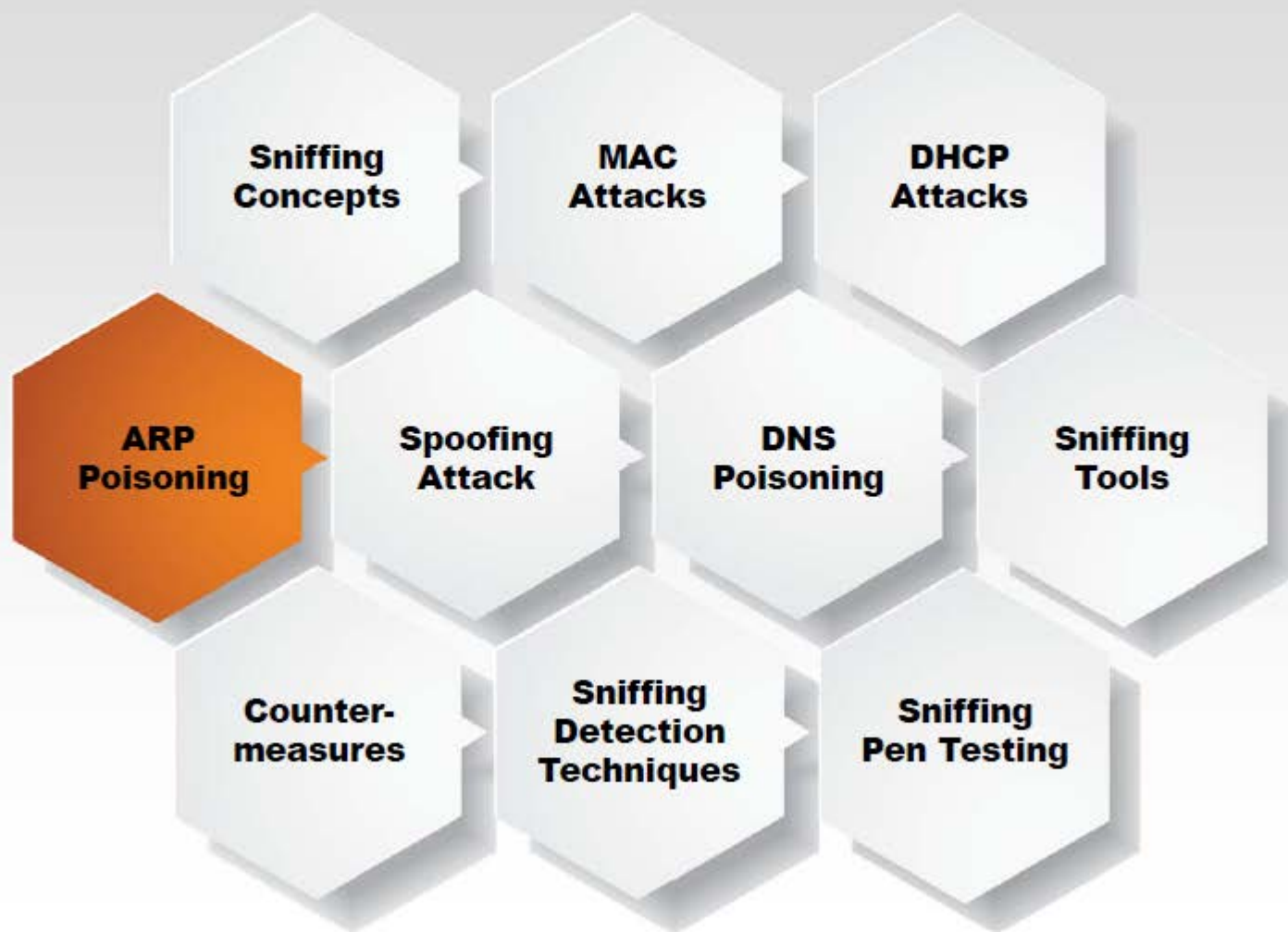


## IOS Global Commands

- `ip dhcp snooping vlan 4,104` → this is what VLANs to snoop
- `no ip dhcp snooping information option` → this allows some DHCP options
- `ip dhcp snooping` → this turns on DHCP snooping

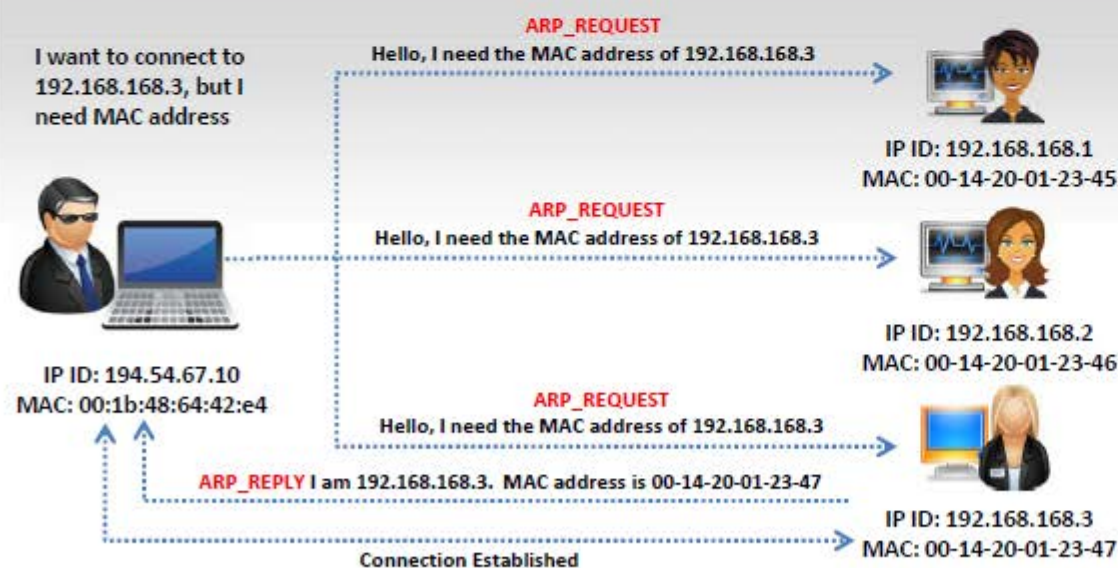
**Note:** All ports in the VLAN are not trusted by default

# Module Flow



# What Is Address Resolution Protocol (ARP)?

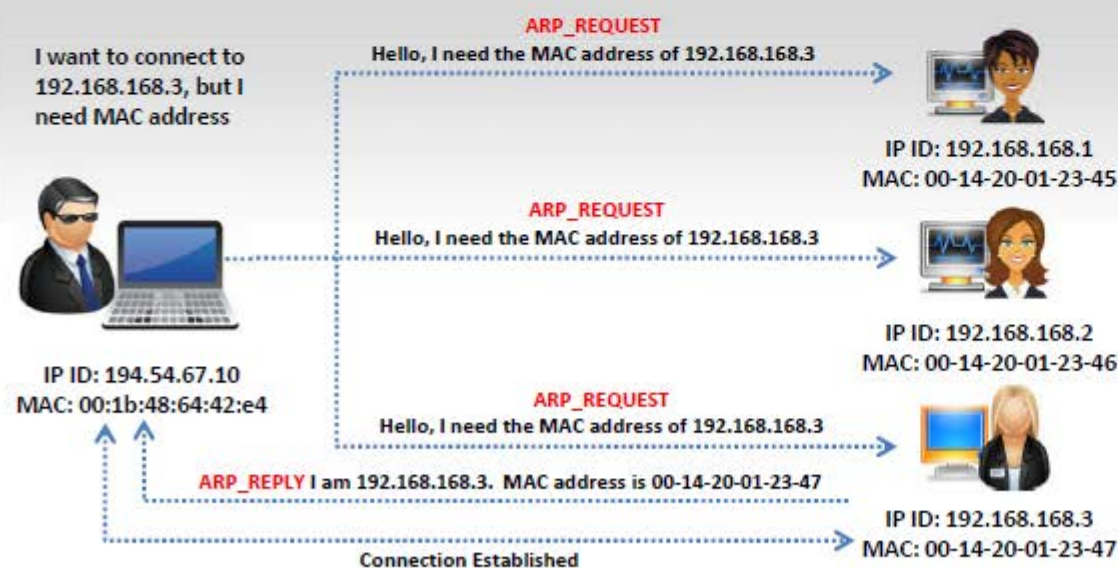
- Address Resolution Protocol (ARP) is a stateless protocol used for **resolving IP addresses to machine (MAC) addresses**
- All network devices (that needs to communicate on the network) broadcasts ARP queries in the network to find out other **machines' MAC addresses**
- When one machine needs to communicate with another, it looks up its ARP table. If the MAC address is not found in the table, the **ARP\_REQUEST** is broadcasted over the network.
- All machines on the network will compare this IP address to their MAC address
- If one of the machine in the network identifies with this address, it will respond to ARP\_REQUEST with its IP and MAC address. The requesting machine will store the address pair in the ARP table and communication will take place



```
C:\Windows\system32\cmd.exe
C:\Users>arp -a
Interface: 192.168.168.188 --- 0x3
Internet Address      Physical Address      Type
192.168.168.1         00-14-20-01-23-45    dynamic
192.168.168.2         00-14-20-01-23-46    dynamic
192.168.168.9         00-14-20-01-23-47    dynamic
192.168.168.11        00-14-20-01-23-48    dynamic
192.168.168.14        00-14-20-01-23-49    dynamic
192.168.168.15        00-14-20-01-23-4A    dynamic
192.168.168.16        00-14-20-01-23-4B    dynamic
192.168.168.17        00-14-20-01-23-4C    dynamic
192.168.168.19        00-14-20-01-23-4D    dynamic
192.168.168.26        00-14-20-01-23-4E    dynamic
192.168.168.30        00-14-20-01-23-4F    dynamic
```

# What Is Address Resolution Protocol (ARP)?

- Address Resolution Protocol (ARP) is a stateless protocol used for **resolving IP addresses to machine (MAC) addresses**
- All network devices (that needs to communicate on the network) broadcasts ARP queries in the network to find out other **machines' MAC addresses**
- When one machine needs to communicate with another, it looks up its ARP table. If the MAC address is not found in the table, the **ARP\_REQUEST** is broadcasted over the network.
- All machines on the network will compare this IP address to their MAC address
- If one of the machine in the network identifies with this address, it will respond to ARP\_REQUEST with its IP and MAC address. The requesting machine will store the address pair in the ARP table and communication will take place



```
C:\Windows\system32\cmd.exe
C:\Users>arp -a
Interface: 192.168.168.188 --- 0x3
Internet Address      Physical Address      Type
192.168.168.1         00-14-20-01-23-45    dynamic
192.168.168.2         00-14-20-01-23-46    dynamic
192.168.168.9         00-14-20-01-23-47    dynamic
192.168.168.11        00-14-20-01-23-48    dynamic
192.168.168.14        00-14-20-01-23-49    dynamic
192.168.168.15        00-14-20-01-23-4A    dynamic
192.168.168.16        00-14-20-01-23-4B    dynamic
192.168.168.17        00-14-20-01-23-4C    dynamic
192.168.168.19        00-14-20-01-23-4D    dynamic
192.168.168.26        00-14-20-01-23-4E    dynamic
192.168.168.30        00-14-20-01-23-4F    dynamic
```

# ARP Spoofing Attack



ARP packets can be **forged** to send data to the attacker's machine



ARP Spoofing involves constructing a large number of **forged ARP request** and reply packets to overload a switch

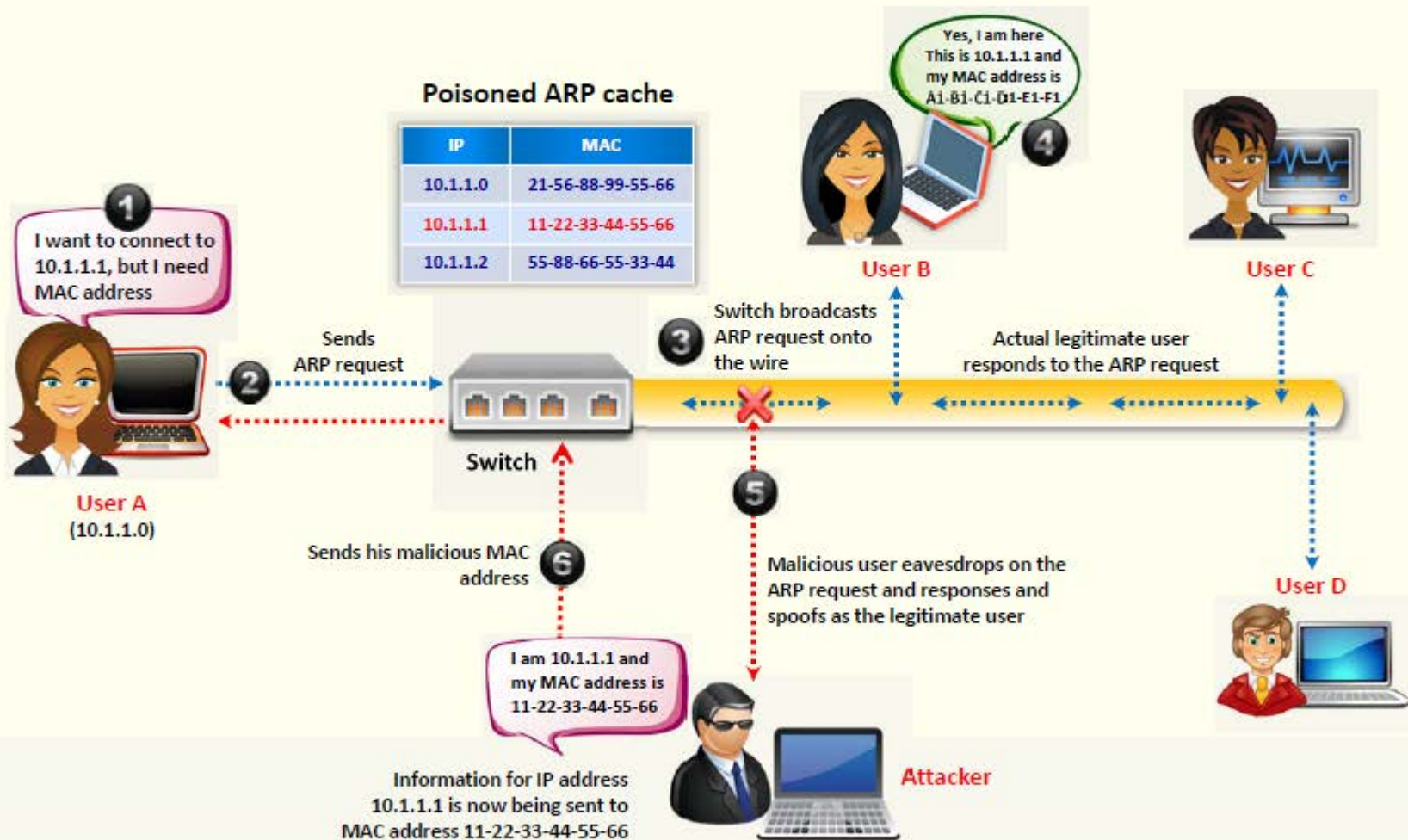


Switch is set in '**forwarding mode**' after ARP table is flooded with spoofed ARP replies and attackers can sniff all the network packets



Attackers flood a target computer's ARP cache with forged entries, which is also known as **poisoning**

# How Does ARP Spoofing Work



# Threats of ARP Poisoning

Using fake **ARP messages**, an attacker can divert all communications between two machines so that all traffic is exchanged via his/her PC



Packet Sniffing



Data Interception



Session Hijacking



Connection Hijacking



VoIP Call Tapping



Connection Resetting



Manipulating Data



Stealing Passwords



Man-in-the-Middle Attack



Denial-of-Service (DoS) Attack

# ARP Poisoning Tools: Cain & Abel and WinArpAttacker



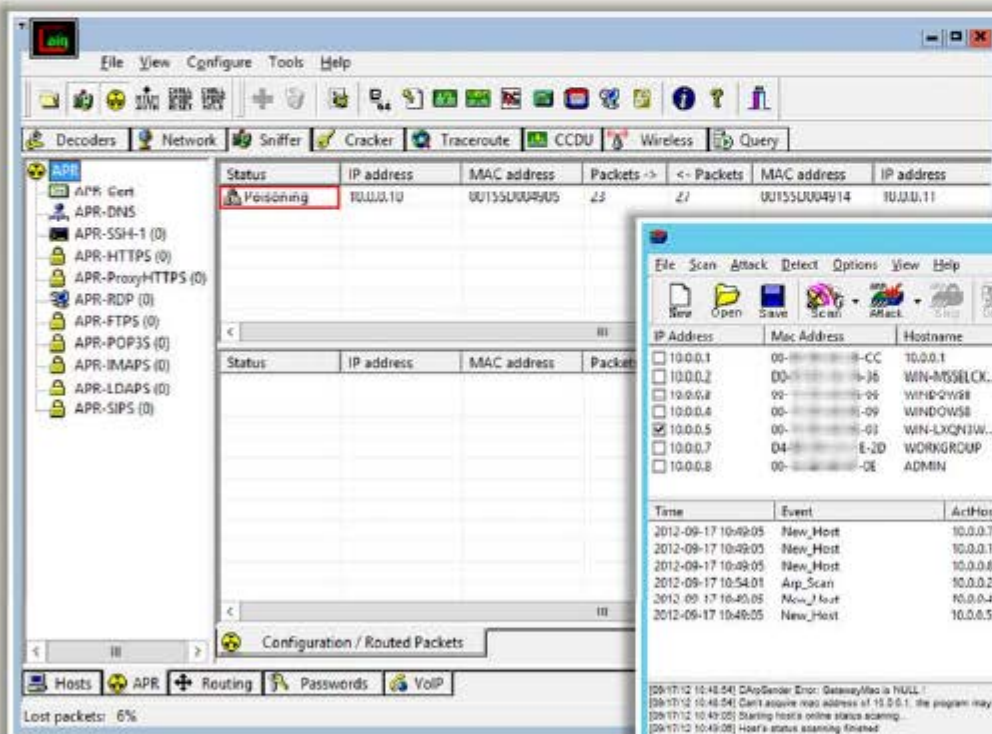
## Cain & Abel

- Cain & Abel allows sniffing packets of various protocols on **switched LANs** by hijacking IP traffic of multiple hosts concurrently

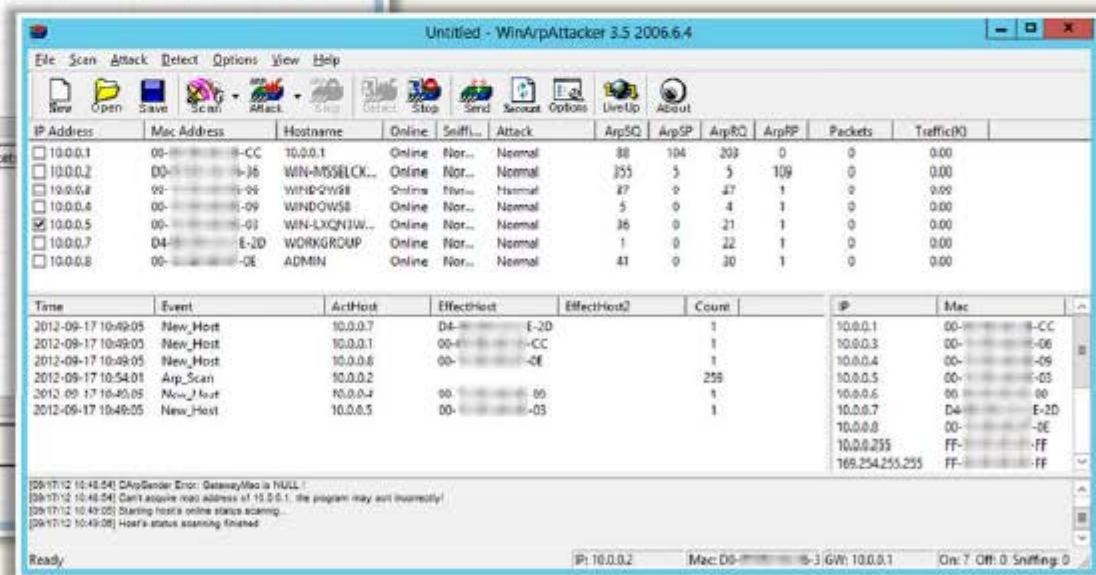


## WinArpAttacker

WinArpAttacker sends **IP conflict packets** to target computers as fast as possible and diverts all communications



<http://www.oxid.it>



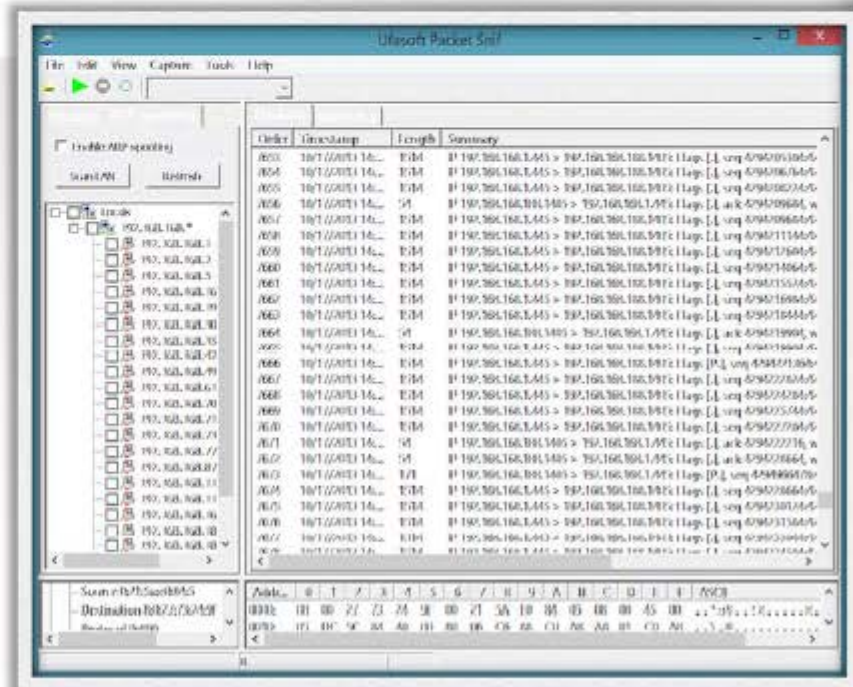
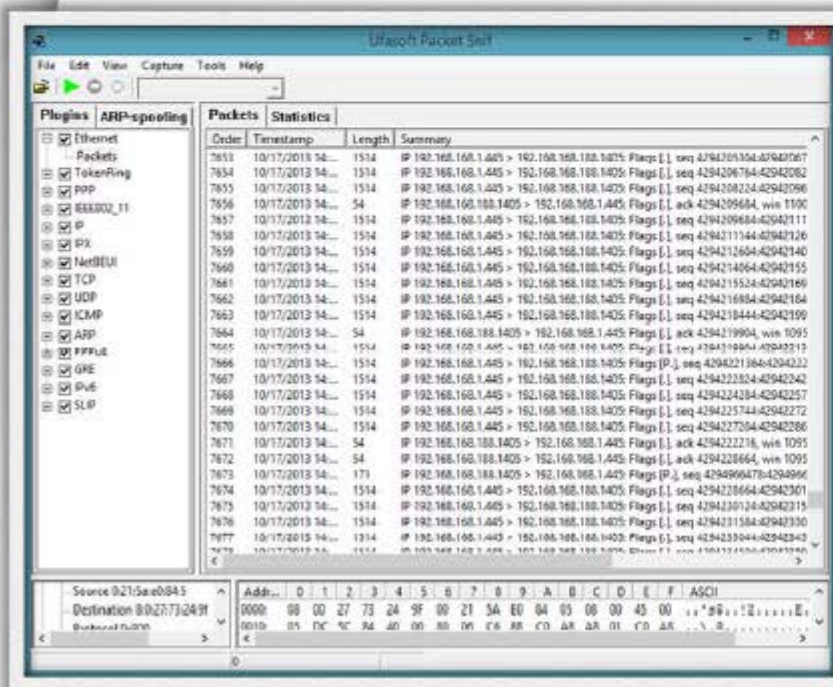
<http://www.xfocus.net>

# ARP Poisoning Tool: Ufasoft Snif

**CEH**  
Certified Ethical Hacker



Ufasoft Snif is an automated ARP poisoning tool that sniffs **passwords** and **email messages** on the network and works on **Wi-Fi network** as well



<http://ufasoft.com>

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

# How to Defend Against ARP Poisoning

Implement **Dynamic ARP Inspection** Using DHCP Snooping Binding Table

```
sh ip dhcp snooping binding
```

MacAddress	IpAddress	Lease	Type	VLAN	Interface
1a:12:3b:2f:df:1c	10.10.10.8	125864	dhcp-snooping	4	FastEthernet 3/18

DHCP Snooping Enabled  
Dynamic ARP Inspection Enabled



10.10.10.1

MAC A

No ARP entry in the binding table then discard the packet

ARP 10.10.10.1  
Saying 10.10.10.2  
is MAC C

ARP 10.10.10.2 Saying  
10.10.10.1 is MAC C

10.10.10.5  
MAC C

10.10.10.2  
MAC B

Check the **MAC** and **IP** fields to see if the ARP from the interface is in the binding; if not, **traffic is blocked**

# Configuring DHCP Snooping and Dynamic ARP Inspection on Cisco Switches

1

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config)# ^Z
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs: 10
DHCP snooping is operational on following VLANs: 10
DHCP snooping is configured on the following L3
Interfaces:
```

DHCP snooping trust/rate is configured on the following Interfaces:

Interface	Trusted	Rate limit (pps)
-----------	---------	------------------

2

```
Switch# show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease	Type	VLAN	Interface
1a:12:3b:2f:df:1c	10.10.10.8	125864	dhcp-snooping	4	FastEthernet 0/3

Total number of bindings: 1

3

```
Switch(config)# ip arp inspection vlan 10
Switch(config)# ^Z
Switch# show ip arp inspection
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan Configuration        Operation ACL Match Static ACL
10 Enabled Active
Vlan ACL Logging DHCP Logging Probe Logging
10 Deny Deny Off
Vlan Forwarded Dropped DHCP Drops ACL Drops
10 0 0 0 0
Vlan DHCP Permits ACL Permits Probe Permits Source MAC Failures
10 0 0 0 0
Vlan Dest MAC Failures IP Validation Failures Invalid Protocol Data
10 0 0 0
```

4

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1
Invalid ARPs (Res) on Fa0/5, vlan
10. ([0013.6050.acf4/192.168.10.1/ffff.
ffff.ffff/192.168.10.1/05:37:31 UTC
Mon Mar 1 2012])
```

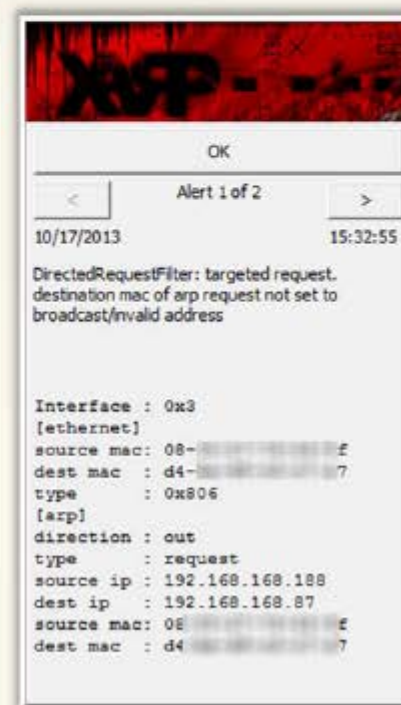
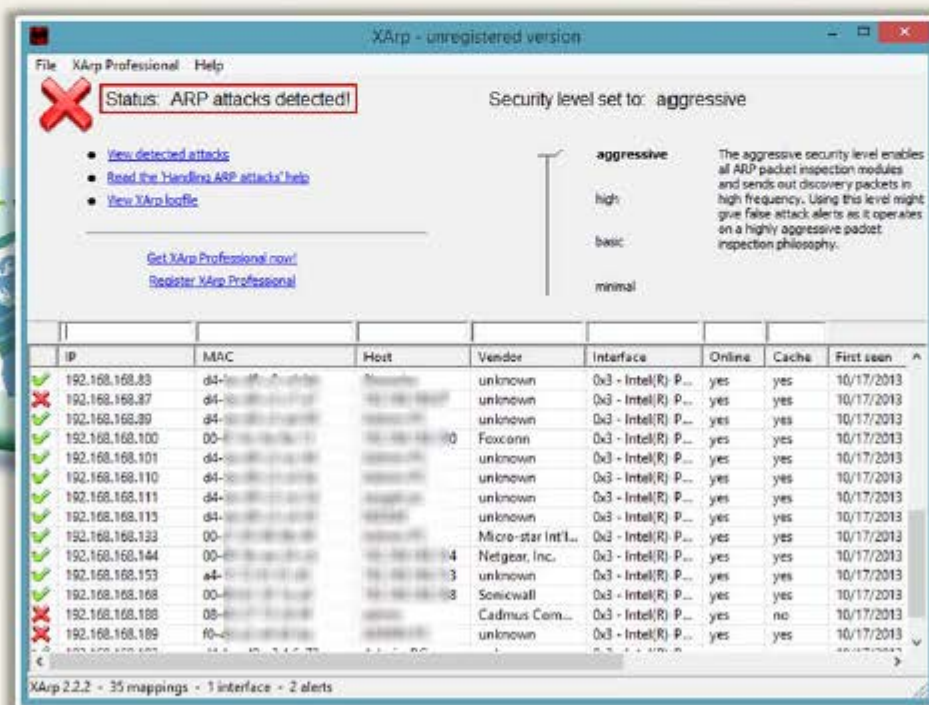


# ARP Spoofing Detection: XArp

**CEH**  
Certified Ethical Hacker



- XArp helps users to detect **ARP attacks** and keep their data private
- It allows administrators to **monitor whole subnets** for ARP attacks
- Different **security levels** and fine tuning possibilities allow normal and power users to efficiently use XArp to detect ARP attacks



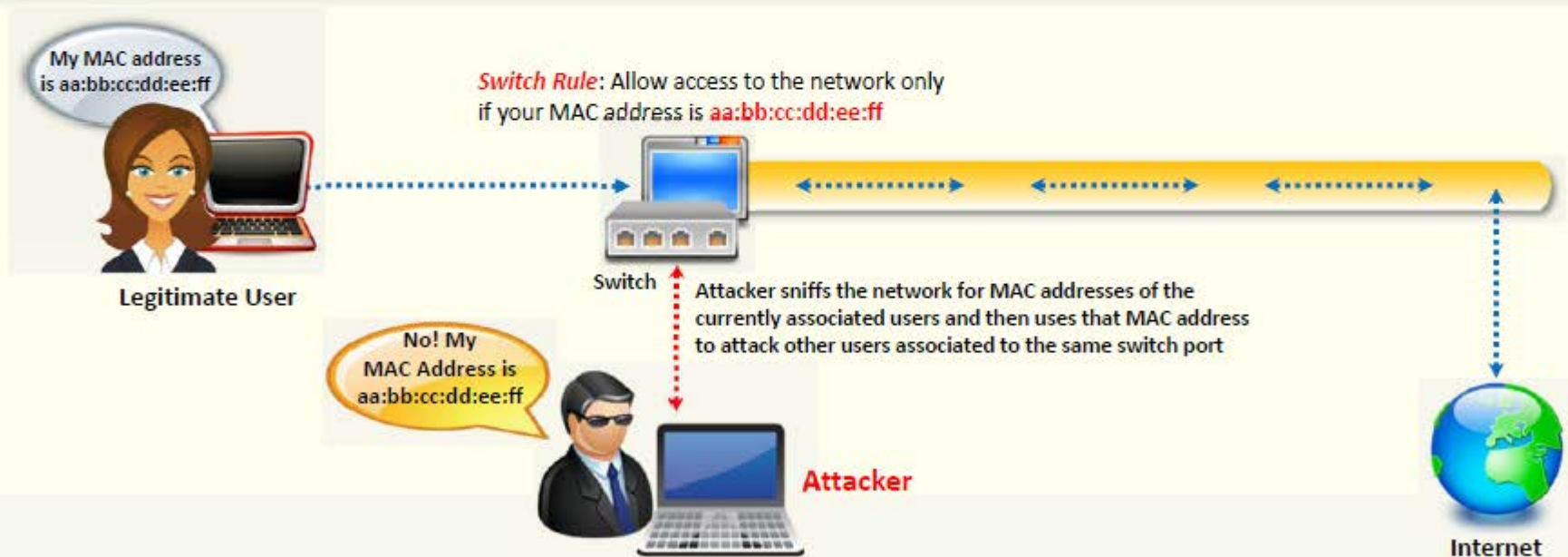
<http://www.chrismc.de>

# Module Flow



# MAC Spoofing/Duplicating

- MAC duplicating attack is launched by **sniffing a network for MAC addresses** of clients who are actively associated with a switch port and re-using one of those addresses
- By listening to the traffic on the network, a malicious user can **intercept and use a legitimate user's MAC address** to receive all the traffic destined for the user
- This attack allows an attacker to **gain access to the network** and take over someone's identity already on the network

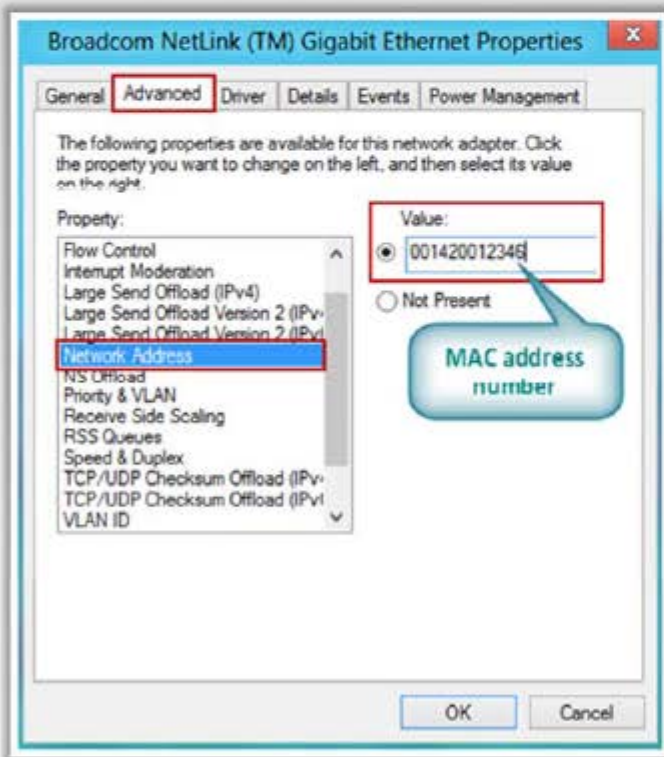


**Note:** This technique can be used to bypass Wireless Access Points' MAC filtering

# MAC Spoofing Technique: Windows

## In Windows 8 OS

**Method 1:** If the network interface card supports clone MAC address then follow the steps:



1

Go to **Right bottom** of the screen → **Settings** → **Control Panel** → **Network and Internet** → **Networking and Sharing Center**

2

Click on the **Ethernet** and then click on the **Properties** in the Ethernet Status window

3

In the Ethernet properties window click on the **Configure** button and then on the **Advanced** tab

4

Under the "**Property:**" section, browse for **Network Address** and click on it

5

On the right side, under "**Value:**", type in the new MAC address you would like to assign and click **OK**

**Note:** Enter the MAC address number without "-" in between

6

Type "**ipconfig/all**" or "**net config rdr**" in command prompt to verify the changes

7

If the changes are visible then **reboot** the system, else try method 2 (change MAC address in the registry)

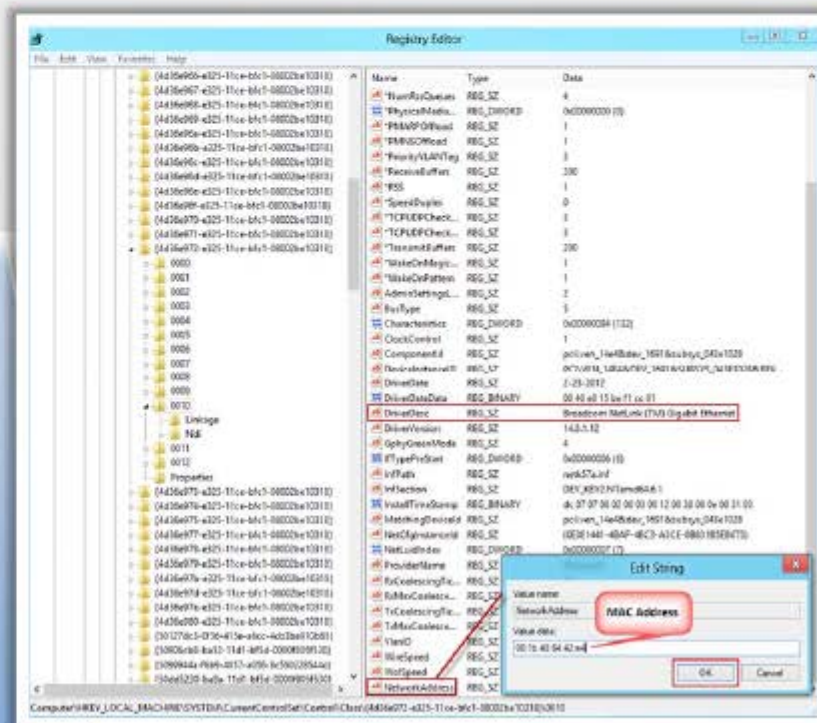
# MAC Spoofing Technique: Windows (Cont'd)

## Method 2: Steps to change MAC address in Registry

- Go to **Start** → **Run**, type **regedt32** to start registry editor

**Note:** Do not type **Regedit** to start registry editor

- Go to  
"**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}**" and double click on it to expand the tree
- 4-digit sub keys representing network adapters will be found (starting with 0000, 0001, 0002, etc.)
- Search for the proper "**DriverDesc**" key to find the desired interface
- Edit, or add, the string key "**NetworkAddress**" (data type "**REG\_SZ**") to contain the new MAC address
- Disable** and then **re-enable** the network interface that was changed or reboot the system



# MAC Spoofing Tool: **SMAC**

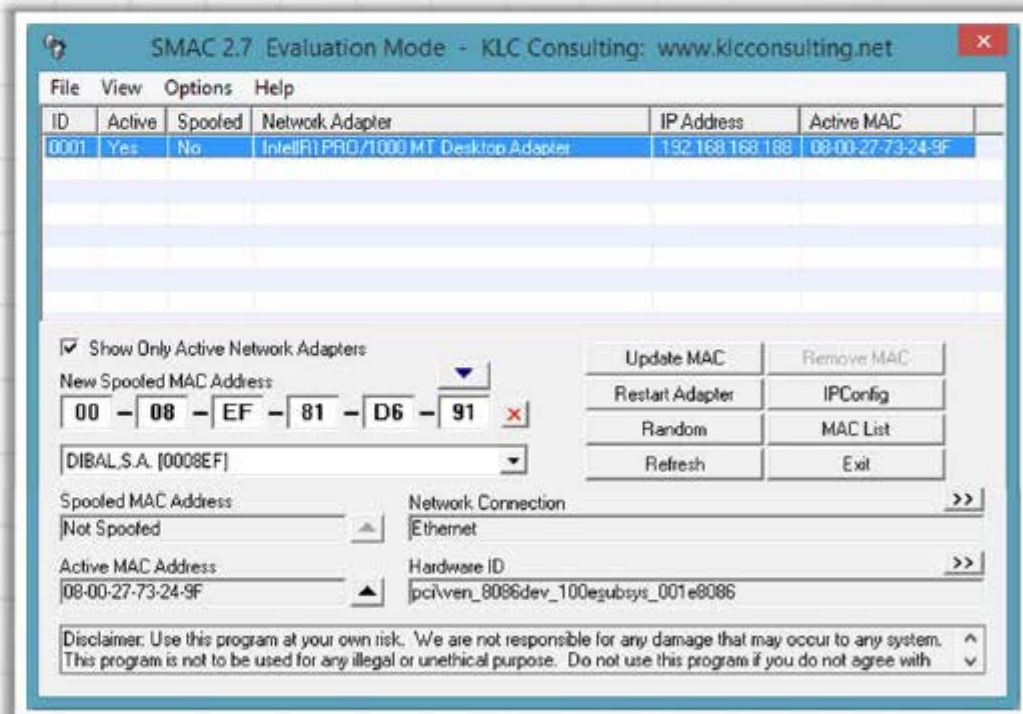
**CEH**  
Certified Ethical Hacker

SMAC is a MAC Address Changer (Spoofers) that allows users to **change MAC address** for any network interface cards (NIC) on the Windows systems



## Features

- Automatically activates new **MAC address** right after changing it
- Shows the **manufacturer** of the MAC address
- Randomly **generates any New MAC address** or based on a selected manufacturer



<http://www.klcconsulting.net>

# How to Defend Against MAC Spoofing

Use DHCP Snooping Binding Table, Dynamic ARP Inspection, and IP Source Guard

```
sh ip dhcp snooping binding
```

MacAddress	IpAddress	Lease	Type	VLAN	Interface
2a:33:4c:2f:4a:1c	10.10.10.9	185235	dhcp-snooping	4	FastEthernet3/18



10.10.10.1  
MAC A

DHCP Snooping Enabled  
Dynamic ARP Inspection Enabled  
IP Source Guard Enabled

IP and MAC entry in the binding table does not match then discard the packet

Traffic Sent with IP 10.10.10.5 Mac B

10.10.10.2  
MAC B

Traffic Sent with IP 10.10.10.2 Mac C

Received Traffic Source IP 10.10.10.2 Mac B

10.10.10.5  
MAC C

Check the **MAC** and **IP** fields to see if the traffic from the interface is in the binding table; if not, **traffic is blocked**

# How to Defend Against MAC Spoofing

Use DHCP Snooping Binding Table, Dynamic ARP Inspection, and IP Source Guard

```
sh ip dhcp snooping binding
```

MacAddress	IpAddress	Lease	Type	VLAN	Interface
2a:33:4c:2f:4a:1c	10.10.10.9	185235	dhcp-snooping	4	FastEthernet3/18



10.10.10.1  
MAC A

DHCP Snooping Enabled  
Dynamic ARP Inspection Enabled  
IP Source Guard Enabled

IP and MAC entry in the binding table does not match then discard the packet

Traffic Sent with IP 10.10.10.5 Mac B

10.10.10.2  
MAC B

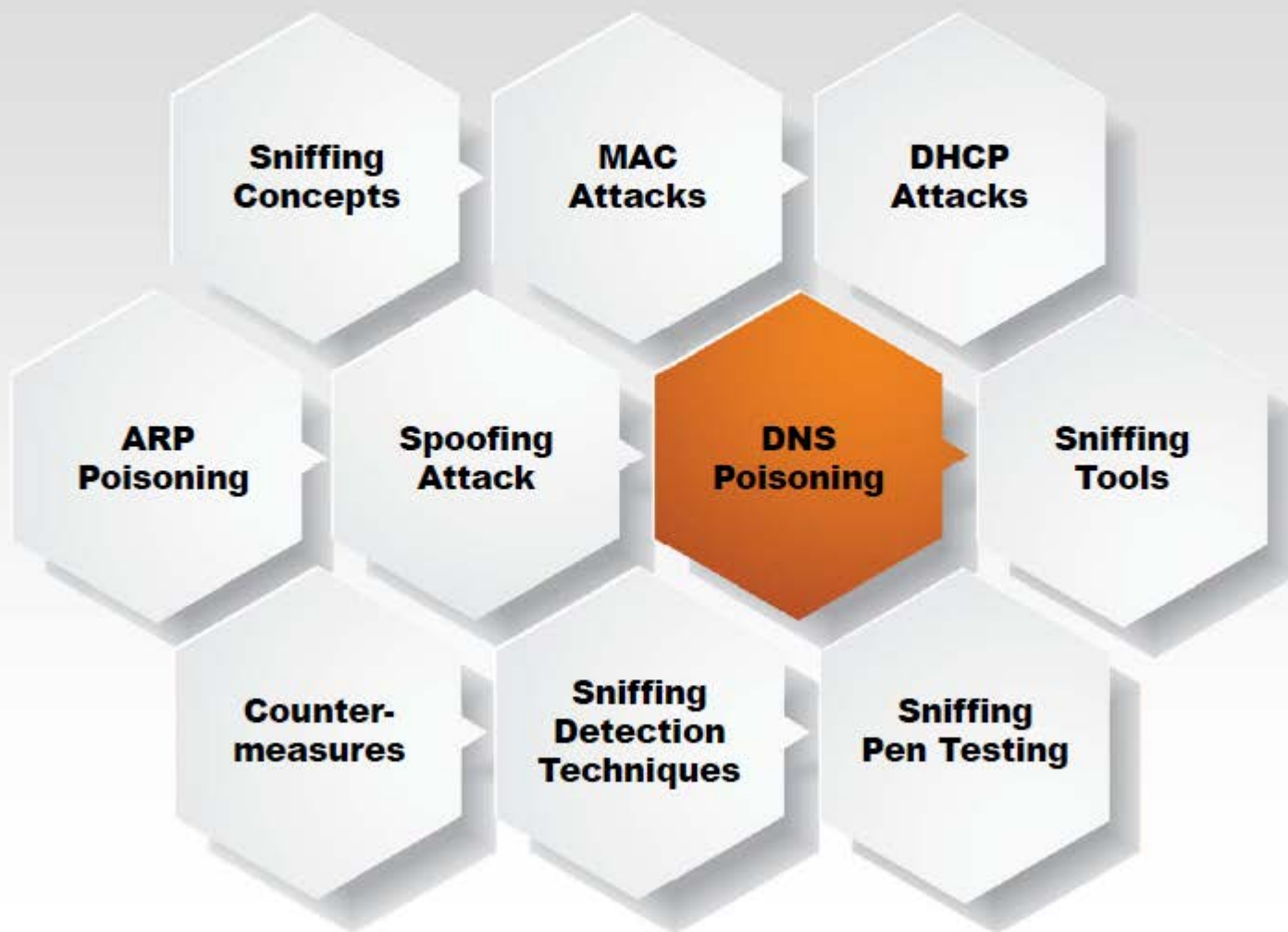
Traffic Sent with IP 10.10.10.2 Mac C

Received Traffic Source IP 10.10.10.2 Mac B

10.10.10.5  
MAC C

Check the **MAC** and **IP** fields to see if the traffic from the interface is in the binding table; if not, **traffic is blocked**

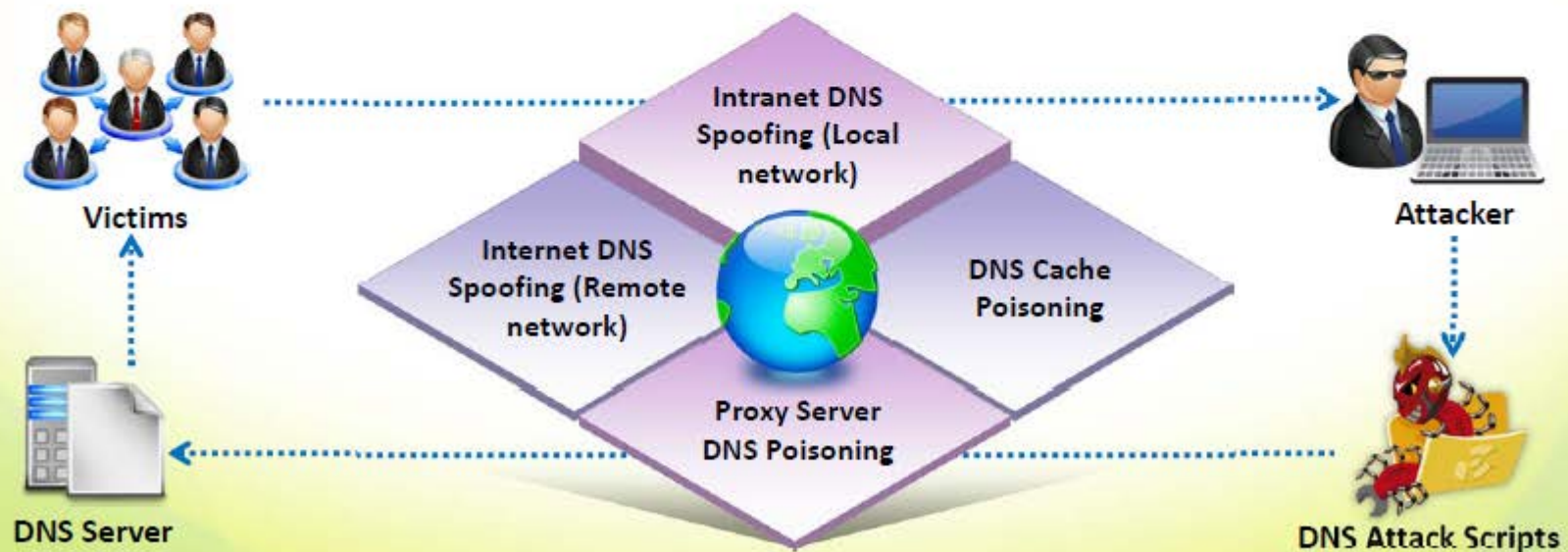
# Module Flow



# DNS Poisoning Techniques

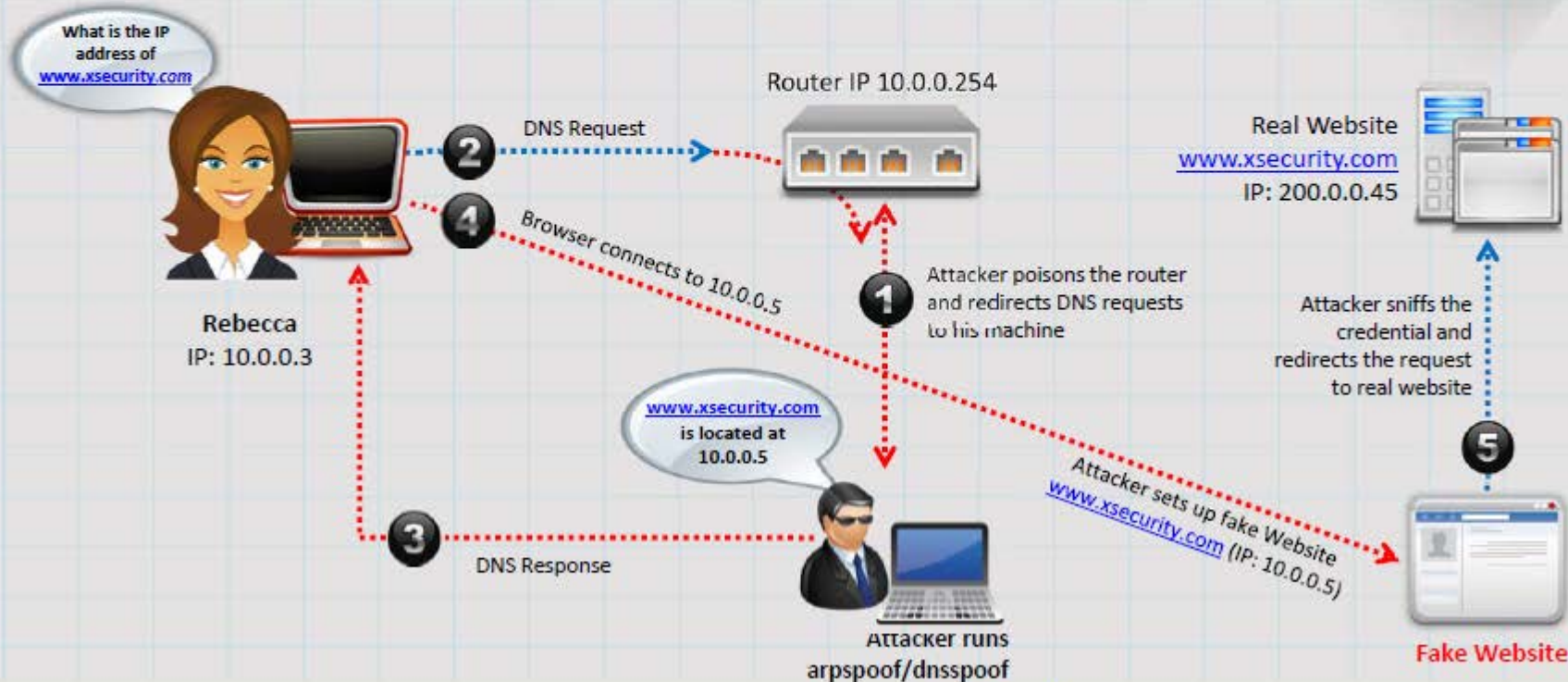
- DNS poisoning is a technique that **tricks a DNS server** into believing that it has received authentic information when, in reality, it has not
- It results in **substitution of a false IP address** at the DNS level where web addresses are converted into numeric IP addresses

- It allows attacker to replace **IP address entries** for a target site on a given DNS server with IP address of the server he/she controls
- Attacker can create **fake DNS entries** for the server (containing malicious content) with same names as that of the target server



# Intranet DNS Spoofing

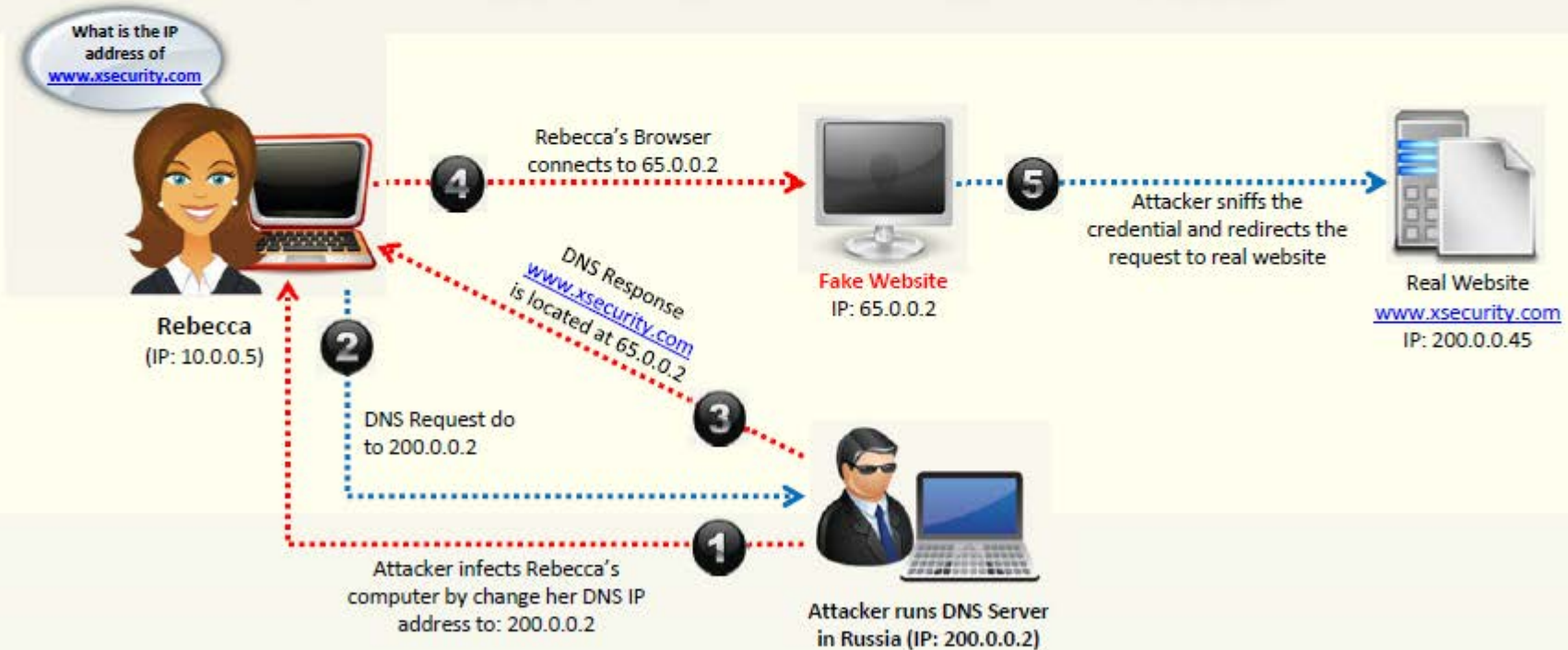
- For this technique, you must be connected to the **local area network (LAN)** and be able to sniff packets
- It works well against **switches** with ARP poisoning the router



# Internet DNS Spoofing



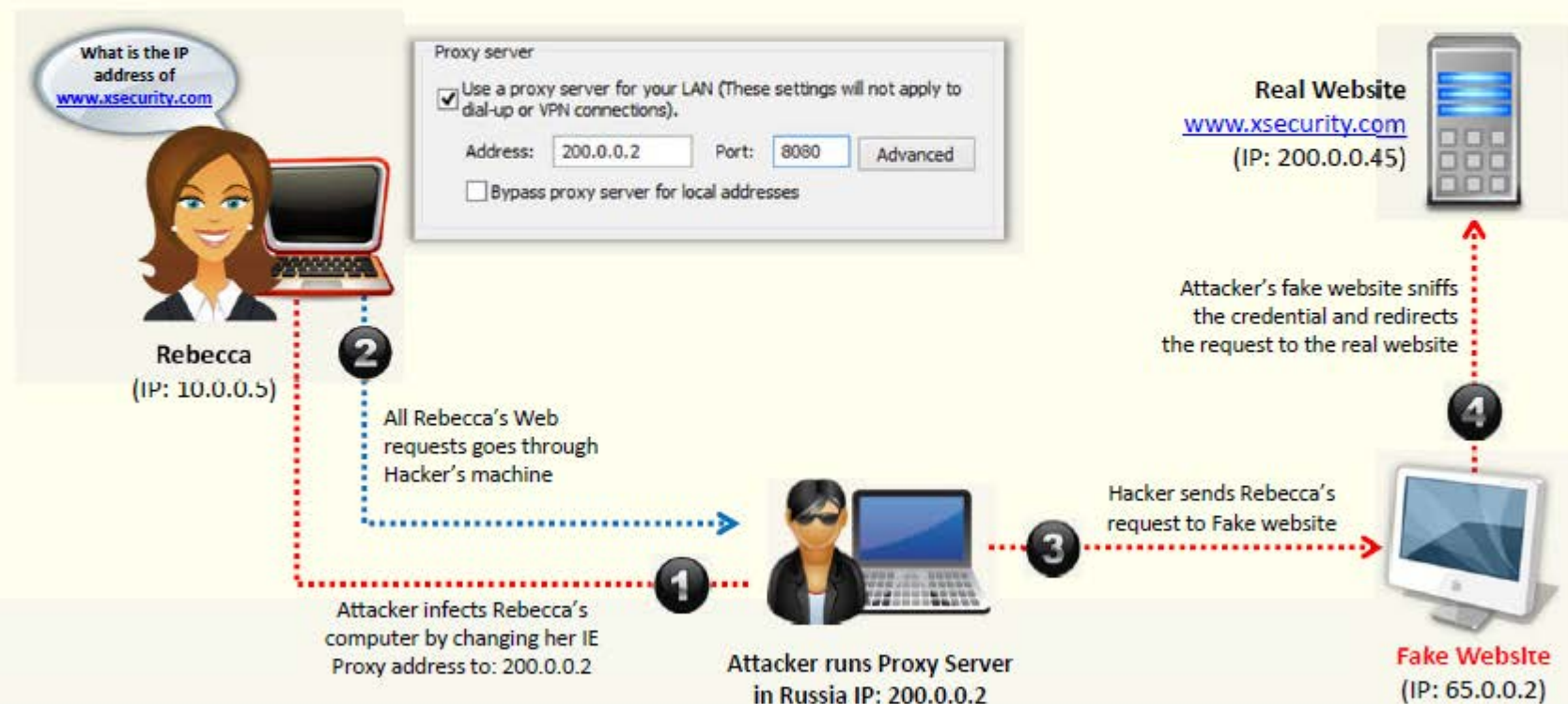
Internet DNS Spoofing, attacker **infects Rebecca's machine** with a Trojan and **changes her DNS IP address** to that of the attacker's



# Proxy Server DNS Poisoning



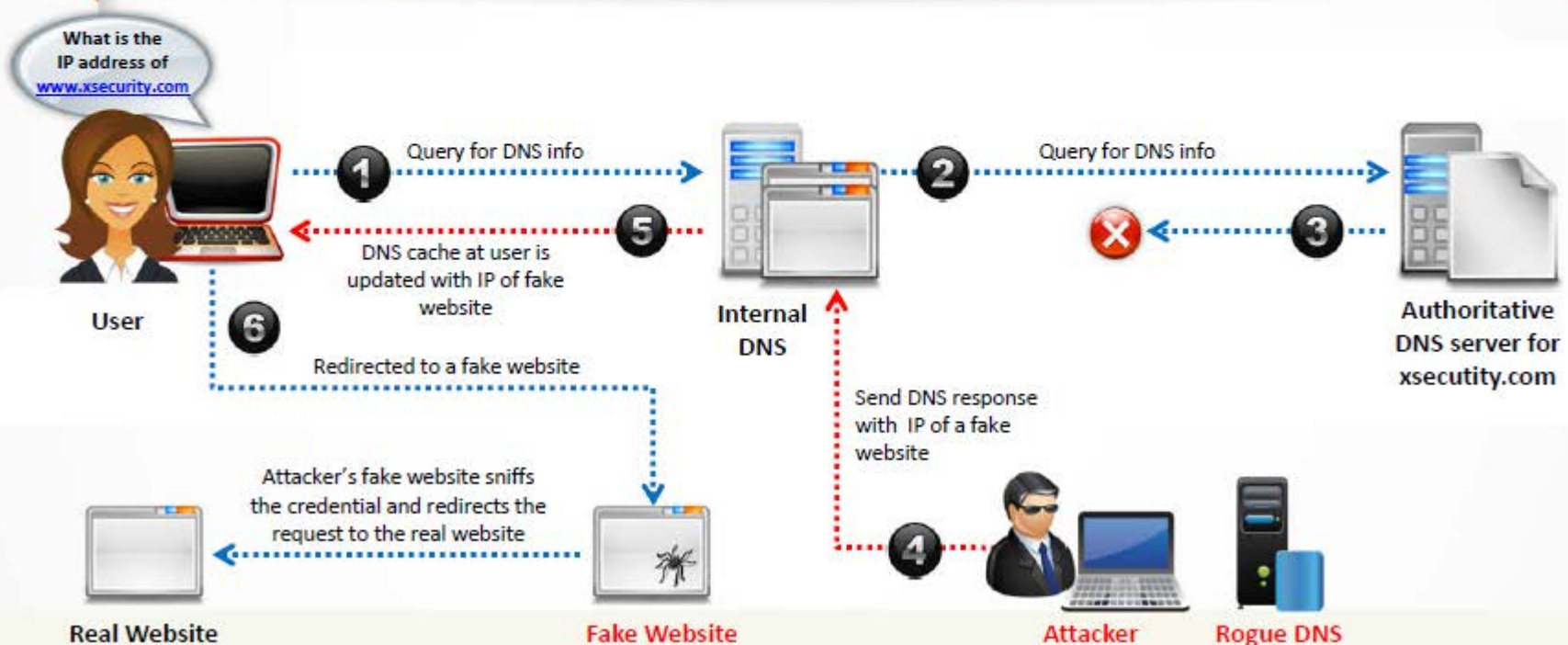
Attacker sends a Trojan to Rebecca's machine that changes her **proxy server settings** in Internet Explorer to that of the attacker's and redirects to fake website



# DNS Cache Poisoning

**01** DNS cache poisoning refers to **altering** or **adding forged DNS records** into the DNS resolver cache so that a DNS query is redirected to a malicious site

**02** If the DNS resolver cannot validate that the DNS responses have come from an **authoritative source**, it will cache the **incorrect entries** locally and serve them to users who make the same request



# How to **Defend** Against **DNS Spoofing**



Resolve all **DNS queries** to local DNS server



Block **DNS requests** from going to external servers



Configure **firewall** to restrict external DNS lookup



Implement **IDS** and deploy it correctly



Implement **DNSSEC**



Configure **DNS resolver** to use a new random source port for each outgoing query



Restrict **DNS recusing service**, either full or partial, to authorized users

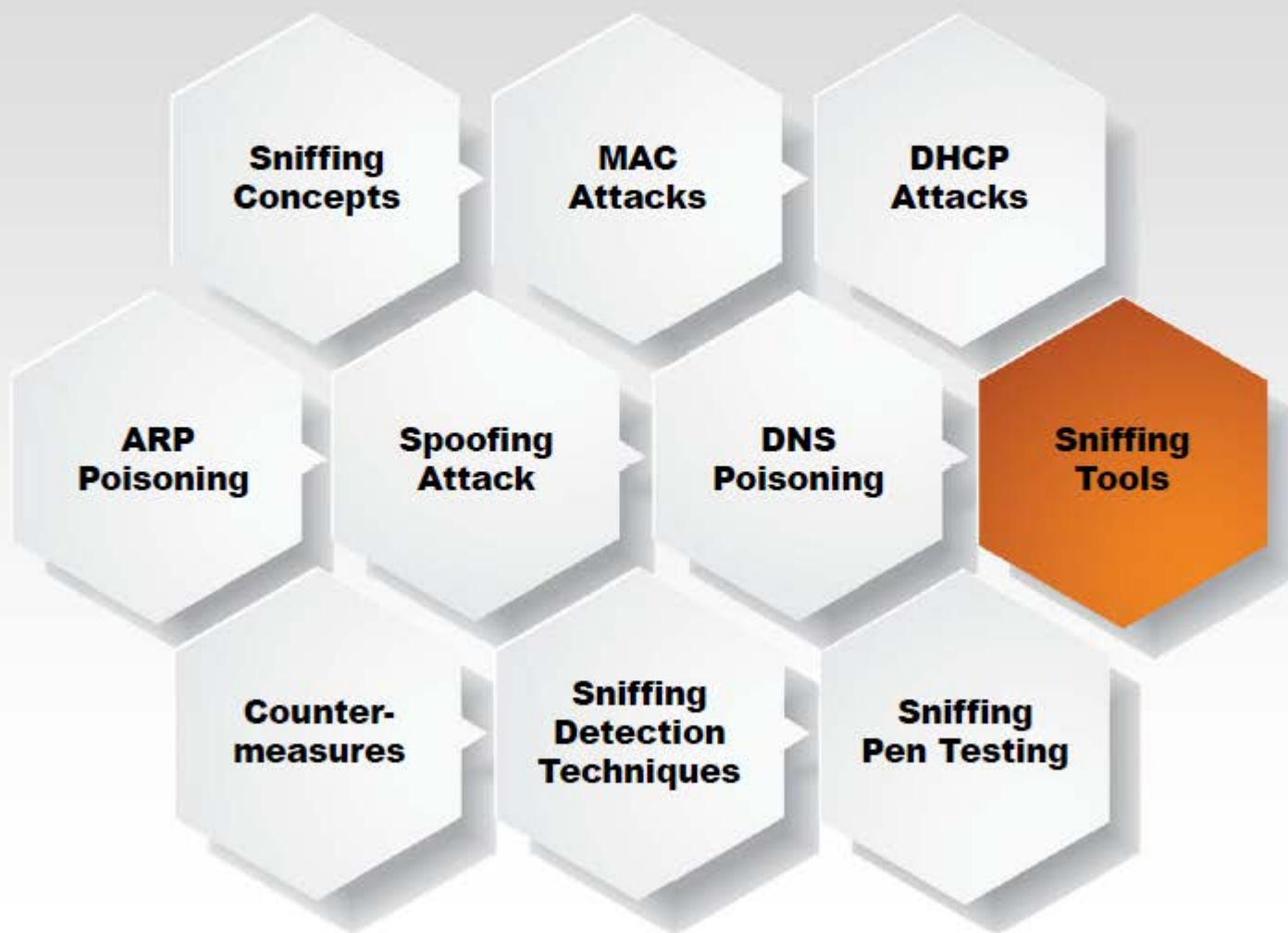


Use **DNS Non-Existent Domain (NXDOMAIN)** Rate Limiting



Secure your **internal machines**

# Module Flow



# Sniffing Tool: Wireshark

It lets you **capture and interactively browse the traffic** running on a computer network

01

Wireshark uses **Winpcap** to capture packets, so it can only capture the packets on the networks supported by Winpcap

02

It **captures live network traffic** from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI networks

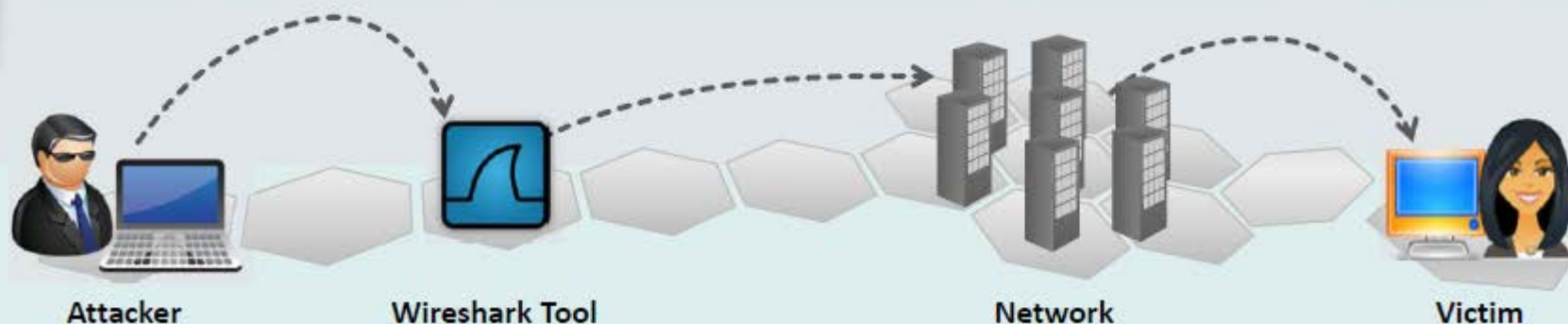
03

Captured files can be programmatically edited via **command-line**

04

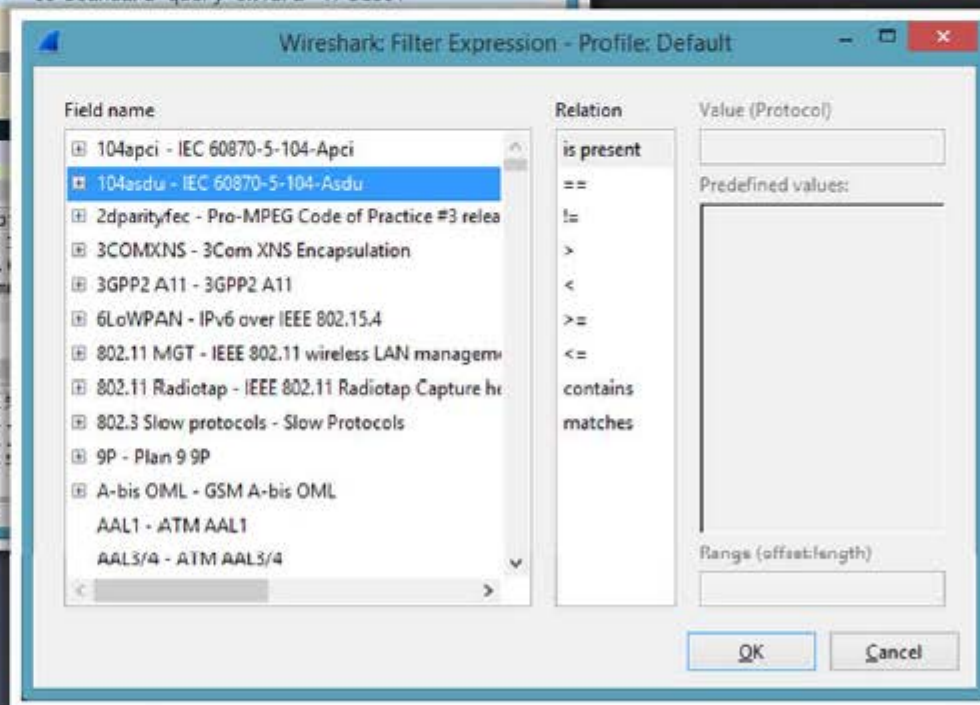
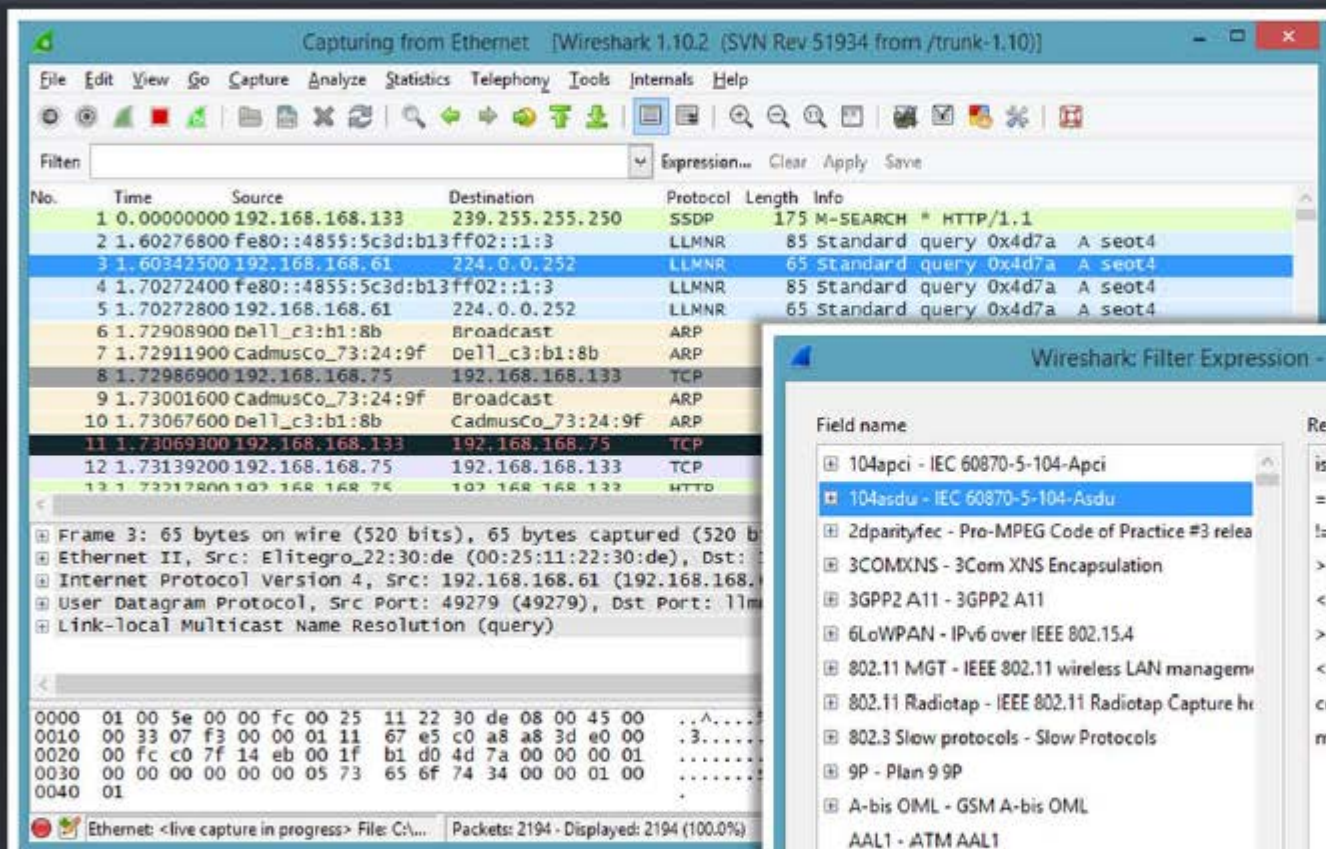
A **set of filters** for customized data display can be refined using a display filter

05



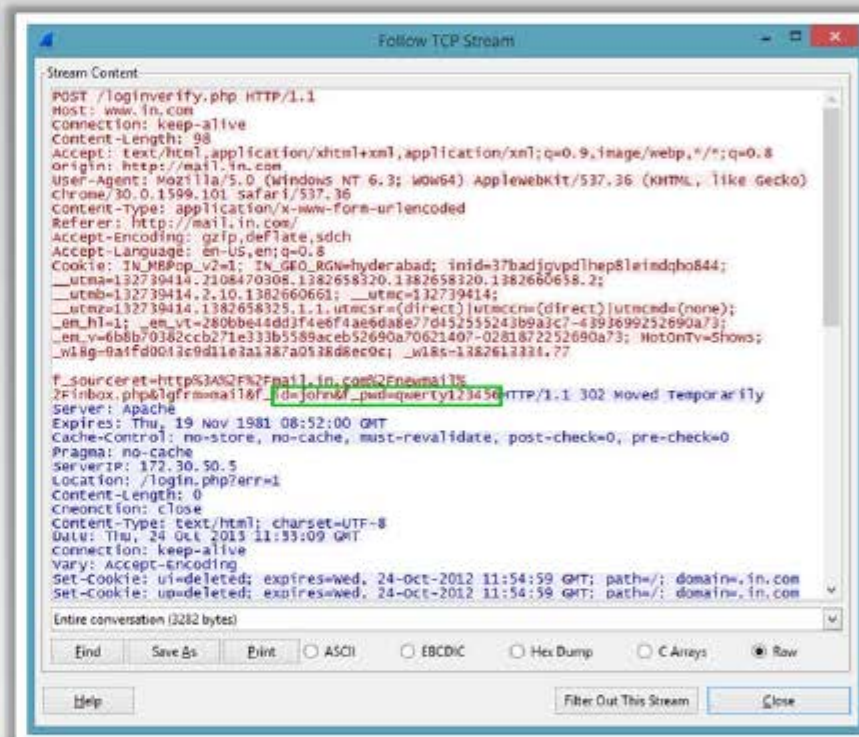
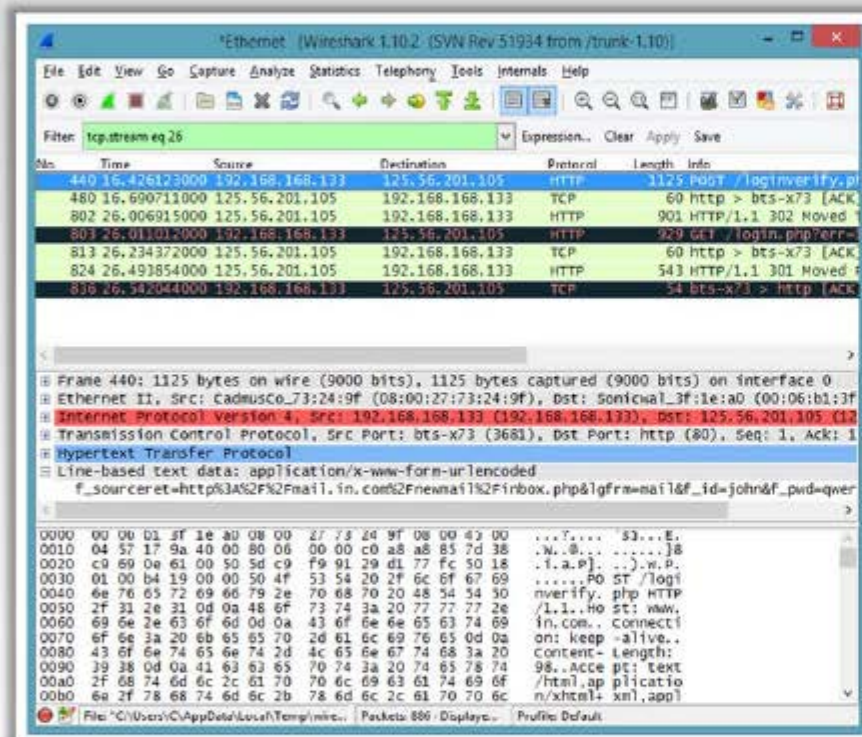
# Sniffing Tool: Wireshark

(Cont'd)



<http://www.wireshark.org>

# Follow TCP Stream in Wireshark



**Password revealed  
in TCP Stream**

# Display Filters in Wireshark

Display filters are used to **change the view of packets** in the captured files

1

## Display Filtering by Protocol

Example: Type the protocol in the filter box; arp, http, tcp, udp, dns, ip



2

## Monitoring the Specific Ports

- tcp.port==23
- ip.addr==192.168.1.100 machine  
ip.addr==192.168.1.100 && tcp.port=23

3

## Filtering by Multiple IP Addresses

- ip.addr == 10.0.0.4 or  
ip.addr == 10.0.0.5

4

## Filtering by IP Address

- ip.addr == 10.0.0.4

5

## Other Filters

- ip.dst == 10.0.1.50 && frame.pkt\_len > 400
- ip.addr == 10.0.1.12 && icmp && frame.number > 15 && frame.number < 30
- ip.src==205.153.63.30 or ip.dst==205.153.63.30

# Additional Wireshark Filters

01

```
tcp.flags.reset==1
```

Displays all TCP resets



02

```
udp contains 33:27:58
```

Sets a filter for the HEX values of 0x33 0x27 0x58 at any offset



03

```
http.request
```

Displays all HTTP GET requests



04

```
tcp.analysis.retransmission
```

Displays all retransmissions in the trace



05

```
tcp contains traffic
```

Displays all TCP packets that contain the word 'traffic'



06

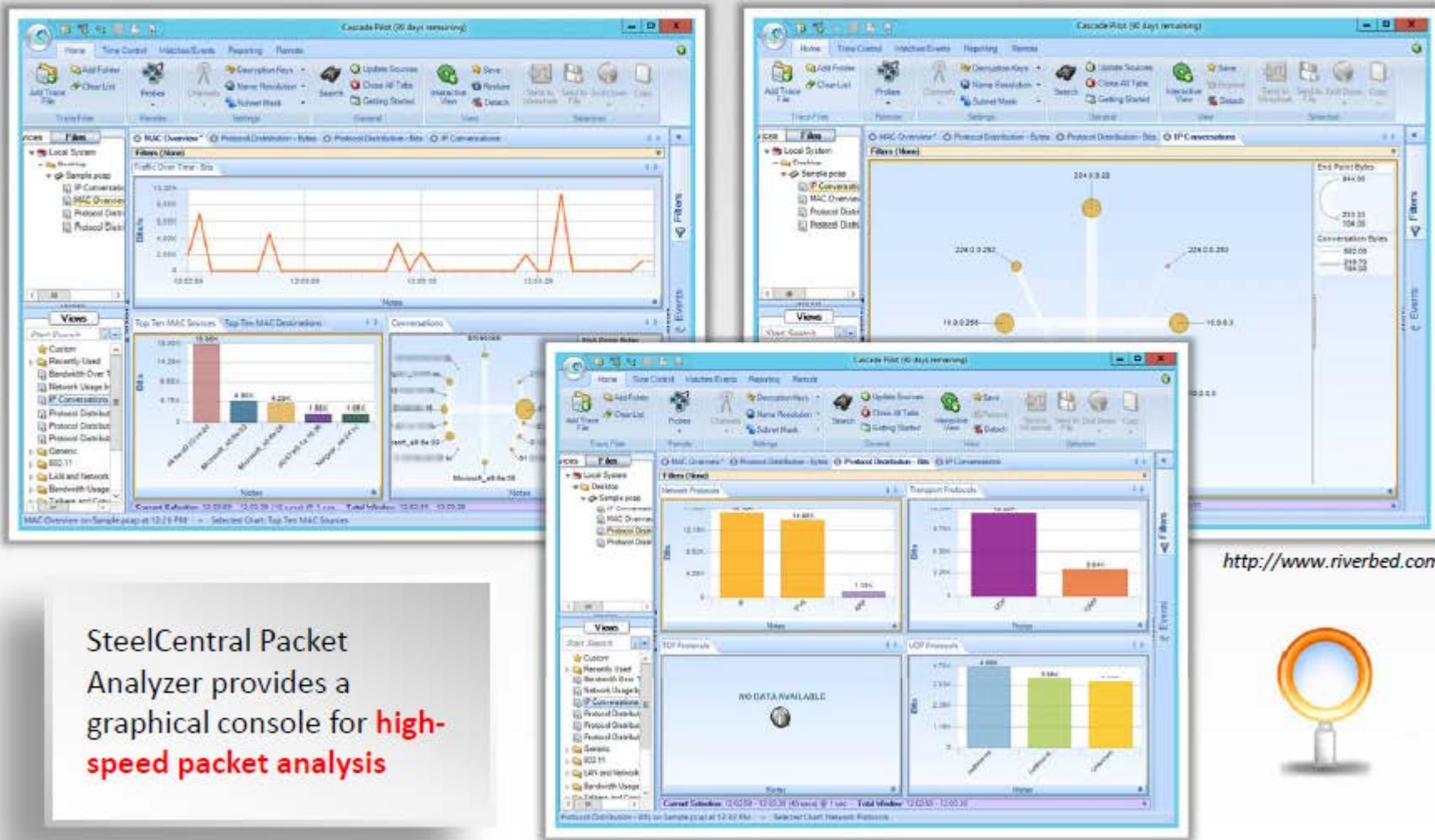
```
!(arp or icmp or dns)
```

Masks out arp, icmp, dns, or other protocols and allows you to view traffic of you interest



# Sniffing Tool: SteelCentral Packet Analyzer

**CEH**  
Certified Ethical Hacker



<http://www.riverbed.com>

SteelCentral Packet Analyzer provides a graphical console for **high-speed packet analysis**



# Sniffing Tool: Tcpdump/Windump

TCPdump is a **command line interface packet sniffer** which runs on Linux and Windows



## TCPDump

Runs on Linux and UNIX systems

```
C:\> tcpdump -i eth0
13:13:48.437836 10.20.21.03.router > RIP2-
ROUTERS.MCAST.NET.router: RIPv2
13:13:48.438364 10.20.21.23 > 10.20.21.55: icmp: RIP2-
ROUTERS.MCAST.NET udp
13:13:54.947195 vmtl.endicott.juggyboy.com.router > RIP2-
ROUTERS.MCAST.NET.rou
13:13:58.313192 :: > ff02::1:ff00:11: icmp6: neighbor sol: who has
fe80::
13:13:59.313573 fe80::26f:5a00:100:11 > ipv6-allrouters: icmp6:
router so
13:14:05.179268 :: > ff02::1:ff00:14: icmp6: neighbor sol: who has
fe80::
13:14:06.179453 fe80::26f:5a00:100:14 > ipv6-allrouters: icmp6:
router so
13:14:18.473315 10.20.21.55.router > RIP2-
ROUTERS.MCAST.NET.router: RIPv2
13:14:18.473950 10.20.21.23 > 10.20.21.55: icmp: RIP2-
ROUTERS.MCAST.NET udp
13:14:20.628769 10.20.21.64.file.net-tms >
btvds01.srv.juggyboy.com.domain: 49
13:14:24.982405 vmtl.endicott.juggyboy.com.router > RIP2-
ROUTERS.MCAST.NET.rou
```

<http://www.tcpdump.org>

## WinDump

Runs on Windows systems

```
C:\Users\C\Desktop\WinDump\WinDump.exe: listening on \Device\NPF_{644A8-B883-...}
15:10:35.004005 IP ... .137 > 192.168.168.255.137: UDP, length
15:10:35.372362 IP6 WIN-F2JB969755.546 > ff02::1:2.547: dhcp6 sol
15:10:35.669372 IP6 admin.50347 > ff02::1:3.5355: UDP, length 46
15:10:35.669718 IP admin.50347 > 224.0.0.252.5355: UDP, length 46
15:10:35.854857 IP6 ... .61220 > ff02::1:3.5355: UDP, length 23
15:10:35.855677 IP6 ... .63168 > 224.0.0.252.5355: UDP, length 23
15:10:35.954878 IP6 ... .61220 > ff02::1:3.5355: UDP, length 23
15:10:35.955385 IP6 ... .63168 > 224.0.0.252.5355: UDP, length 23
15:10:36.082704 IP6 admin.50347 > ff02::1:3.5355: UDP, length 46
15:10:36.083864 IP admin.50347 > 224.0.0.252.5355: UDP, length 46
15:10:36.154879 IP ... .137 > 192.168.168.255.137: UDP, length
15:10:36.459859 IP ... -PC.137 > 192.168.168.255.137: UDP, length
15:10:36.494136 IP admin.137 > ... .137: UDP, length 50
15:10:36.494641 IP6 admin.64799 > ff02::1:3.5355: UDP, length 45
15:10:36.494898 IP admin.64799 > 224.0.0.252.5355: UDP, length 45
15:10:36.495848 IP ... .137 > admin.137: UDP, length 175
15:10:36.496685 IP ... .5355 > admin.64799: UDP, length 94
15:10:36.496743 IP admin > ... : ICMP admin udp port 64799 unre
th 130
15:10:36.497512 IP6 admin.49395 > ff02::1:3.5355: UDP, length 98
15:10:36.497750 IP admin.49395 > 224.0.0.252.5355: UDP, length 98
15:10:36.904606 IP ... .137 > 192.168.168.255.137: UDP, length
15:10:36.908276 IP6 admin.49395 > ff02::1:3.5355: UDP, length 98
15:10:36.908503 IP admin.49395 > 224.0.0.252.5355: UDP, length 98
15:10:37.218104 IP ... -PC.137 > 192.168.168.255.137: UDP, length
15:10:37.252106 IP
```

<http://www.winpcap.org>

# Network Packet Analyzer: OmniPeek Network Analyzer



- OmniPeek sniffer displays a Google Map in the OmniPeek capture window showing the **locations of all the public IP addresses of captured packets**
- This feature is a great way to monitor the network in real time, and show from where in the world that **traffic is coming**



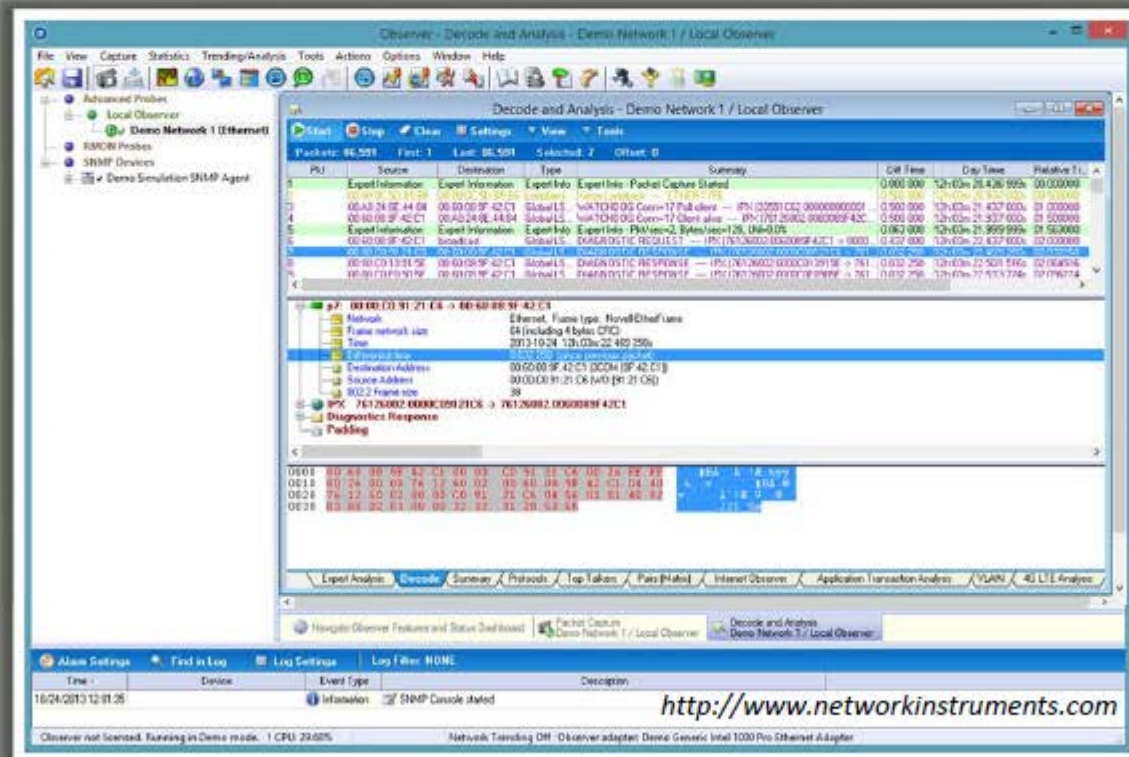
Packet	Source	Destination	Flags	Size	Relative Time	Protocol	Summary
1	97.65.218.170	10.0.0.2		1510	0.000000000	TCP	Src=47681, Dst= 4192, A....S=0
2	10.0.0.2	97.65.218.170		64	0.000000000	TCP	Src= 4192, Dst=47681, A....S=0
3	10.0.0.2	69.90.118.89		842	0.104863000	HTTP	C FOR=4192 GET /Search/... S= 10, Dst= 4192, A....S=0
4	69.90.118.89	10.0.0.2		64	0.135950000	HTTP	Src= 47681, Dst= 4192, A....S=0
5	97.65.218.170	10.0.0.2		1510	0.142946000	TCP	Src=47681, Dst= 4192, A....S=0
6	97.65.218.170	10.0.0.2		1510	0.145923000	TCP	Src=47681, Dst= 4192, A....S=0
7	10.0.0.2	97.65.218.170		64	0.145946000	TCP	Src= 4192, Dst=47681, A....S=0
8	97.65.218.170	10.0.0.2		1510	0.154869000	TCP	Src=47681, Dst= 4192, A....S=0
9	97.65.218.170	10.0.0.2		1510	0.160947000	TCP	Src=47681, Dst= 4192, A....S=0
10	10.0.0.2	97.65.218.170		64	0.160971000	TCP	Src= 4192, Dst=47681, A....S=0
11	97.65.218.170	10.0.0.2		1510	0.170626000	TCP	Src=47681, Dst= 4192, A....S=0
12	97.65.218.170	10.0.0.2		1510	0.176998000	TCP	Src=47681, Dst= 4192, A....S=0
13	10.0.0.2	97.65.218.170		64	0.177032000	TCP	Src= 4192, Dst=47681, A....S=0
14	97.65.218.170	10.0.0.2		1510	0.190613000	TCP	Src=47681, Dst= 4192, A....S=0
15	97.65.218.170	10.0.0.2		1510	0.205018000	TCP	Src=47681, Dst= 4192, A....S=0
16	10.0.0.2	97.65.218.170		64	0.205035000	TCP	Src= 4192, Dst=47681, A....S=0
17	97.65.218.170	10.0.0.2		1510	0.211977000	TCP	Src=47681, Dst= 4192, A....S=0
18	97.65.218.170	10.0.0.2		1510	0.217799000	TCP	Src=47681, Dst= 4192, A....S=0
19	10.0.0.2	97.65.218.170		64	0.217799000	TCP	Src= 4192, Dst=47681, A....S=0
20	97.65.218.170	10.0.0.2		1510	0.225063000	TCP	Src=47681, Dst= 4192, A....S=0
21	97.65.218.170	10.0.0.2		1510	0.231264000	TCP	Src=47681, Dst= 4192, A....S=0
22	10.0.0.2	97.65.218.170		64	0.231274000	TCP	Src= 4192, Dst=47681, A....S=0
23	97.65.218.170	10.0.0.2		1510	0.237170000	TCP	Src=47681, Dst= 4192, A....S=0
24	97.65.218.170	10.0.0.2		1510	0.243422000	TCP	Src=47681, Dst= 4192, A....S=0
25	10.0.0.2	97.65.218.170		64	0.243442000	TCP	Src= 4192, Dst=47681, A....S=0
26	97.65.218.170	10.0.0.2		1510	0.248407000	TCP	Src=47681, Dst= 4192, A....S=0
27	97.65.218.170	10.0.0.2		1510	0.255429000	TCP	Src=47681, Dst= 4192, A....S=0
28	10.0.0.2	97.65.218.170		64	0.255445000	TCP	Src= 4192, Dst=47681, A....S=0
29	97.65.218.170	10.0.0.2		1510	0.261348000	TCP	Src=47681, Dst= 4192, A....S=0



# Network Packet Analyzer: Observer



Observer provides a comprehensive drill-down into network traffic and provides **back-in-time analysis**, reporting, trending, alarms, application tools, and **route monitoring capabilities**



# Network Packet Analyzer: Sniff-O-Matic

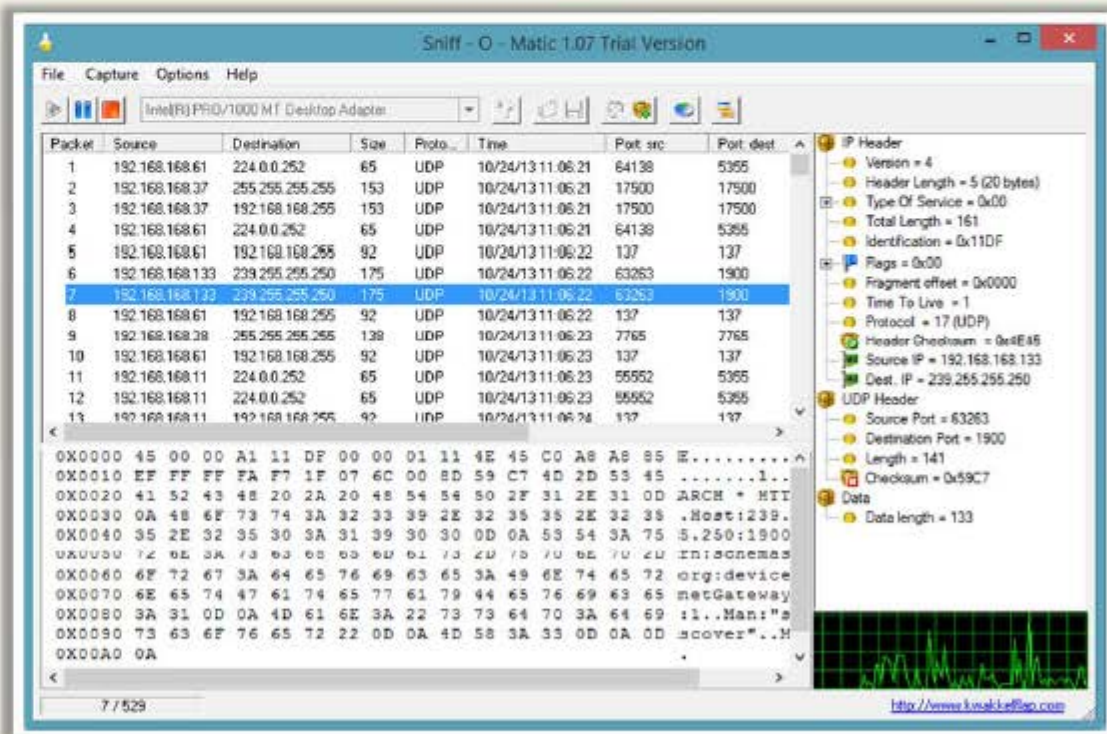


Sniff-O-Matic is a network protocol analyzer and packet sniffer that **captures network traffic** and enables you to **analyze the data**



## Features

- Capture IP packets on your LAN without packet loss
- Monitor network activity in real time
- Filters to show only the packets you want
- Realtime checksum calculation
- Save and load captured packets
- Traffic charts with filter info



<http://www.kwakkelflap.com>

# TCP/IP Packet Crafter: Colasoft Packet Builder



Colasoft Packet Builder allows user to select one from the provided templates: **Ethernet Packet**, **ARP Packet**, **IP Packet**, **TCP Packet** and **UDP Packet**, and **change the parameters** in the decoder editor, hexadecimal editor, or ASCII editor to create a packet



No.	Delta Time	Source	Destination
1	0.100000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF
2	0.100000	0.0.0.0	0.0.0.0
3	0.100000	0.0.0.0:0	0.0.0.0:0
4	0.100000	0.0.0.0:0	0.0.0.0:0

Hex Editor: Total 60 bytes

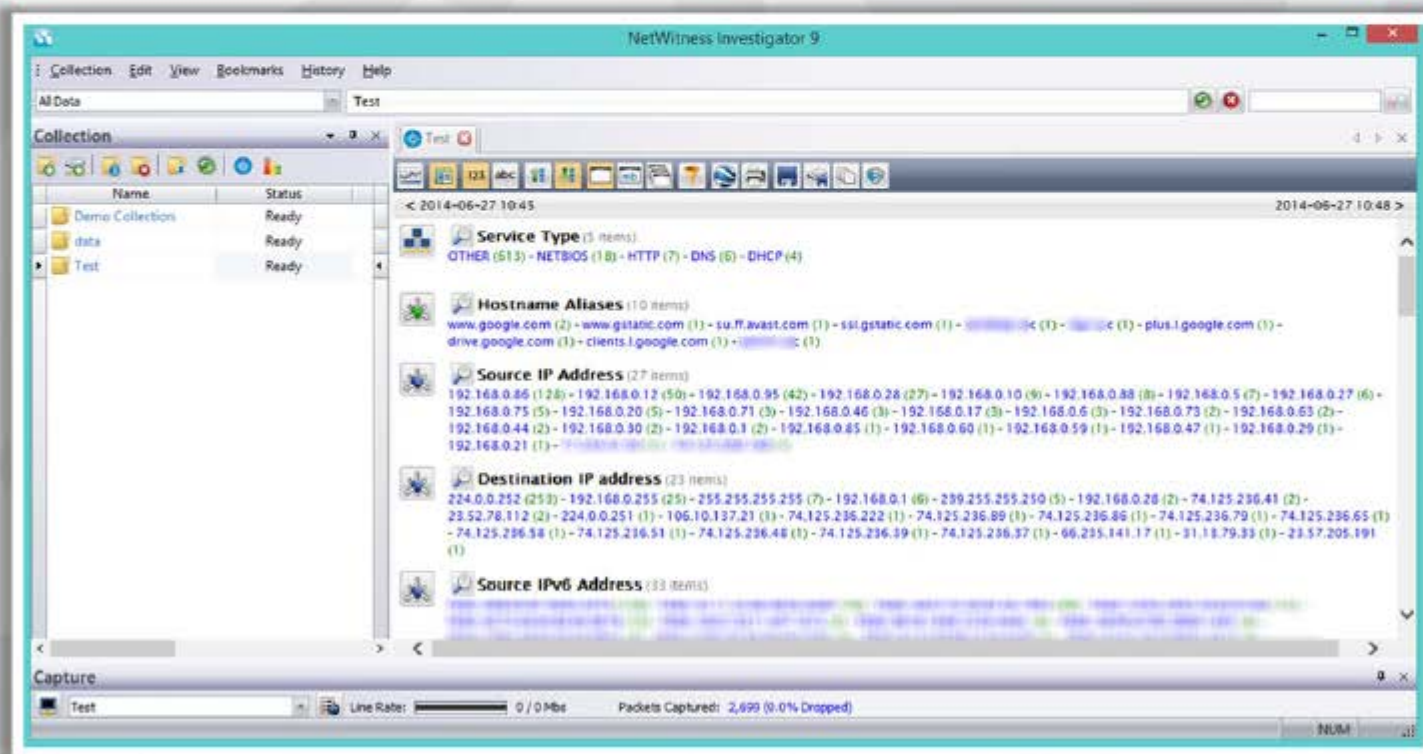
```
0000 FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00
000C 00 06 00 01 00 00 06 04 00 01 00 00 00 00 00 00
0018 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0024 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

<http://www.colasoft.com>

# Network Packet Analyzer: RSA NetWitness Investigator



RSA NetWitness Investigator captures live traffic and process packet files from virtually any existing network collection devices



<http://www.emc.com>

# Additional Sniffing Tools



## Ace Password Sniffer

<http://www.efeotech.com>



## EffeTech HTTP Sniffer

<http://www.efeotech.com>



## IPgrab

<http://ipgrab.sourceforge.net>



## ntopng

<http://www.ntop.org>



## Big-Mother

<http://www.tupsoft.com>



## Ettercap

<http://ettercap.sourceforge.net>



## EtherDetect Packet Sniffer

<http://www.etherdetect.com>



## SmartSniff

<http://www.nirsoft.net>



## dsniff

<http://monkey.org>



## EtherApe

<http://etherape.sourceforge.net>

# Additional Sniffing Tools

(Cont'd)



## Network Probe

<http://www.objectplanet.com>



## CommView

<http://www.tamos.com>



## WebSiteSniffer

<http://www.nirsoft.net>



## NetResident

<http://www.tamos.com>



## ICQ Sniffer

<http://www.etherboss.com>



## Kismet

<http://www.kismetwireless.net>



## MaaTec Network Analyzer

<http://www.maatec.com>



## AIM Sniffer

<http://www.elfetech.com>



## Alchemy Network Monitor

<http://www.mishelpers.com>



## Netstumbler

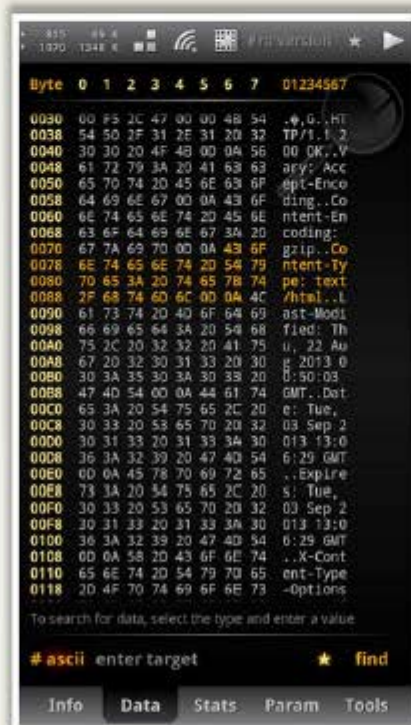
<http://www.netstumbler.com>

# Packet Sniffing Tools for Mobile: **Wi.cap.** **Network Sniffer Pro** and **FaceNiff**



## Wi.cap. Network Sniffer Pro

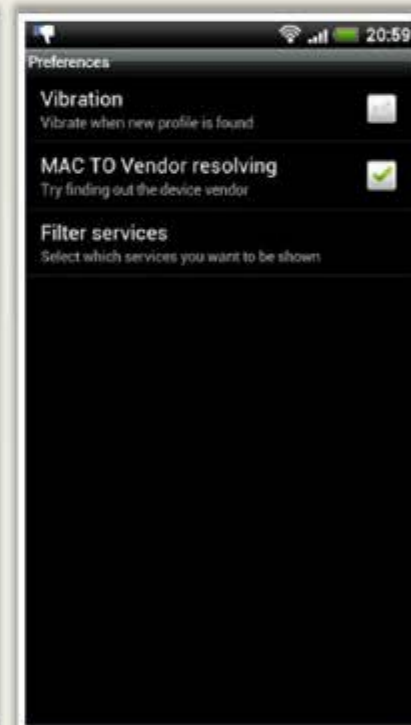
Mobile network packet sniffer for **ROOT ARM droids**



<https://play.google.com>

## FaceNiff

FaceNiff is an Android app that allows you to **sniff** and **intercept web session profiles** over the Wi-Fi



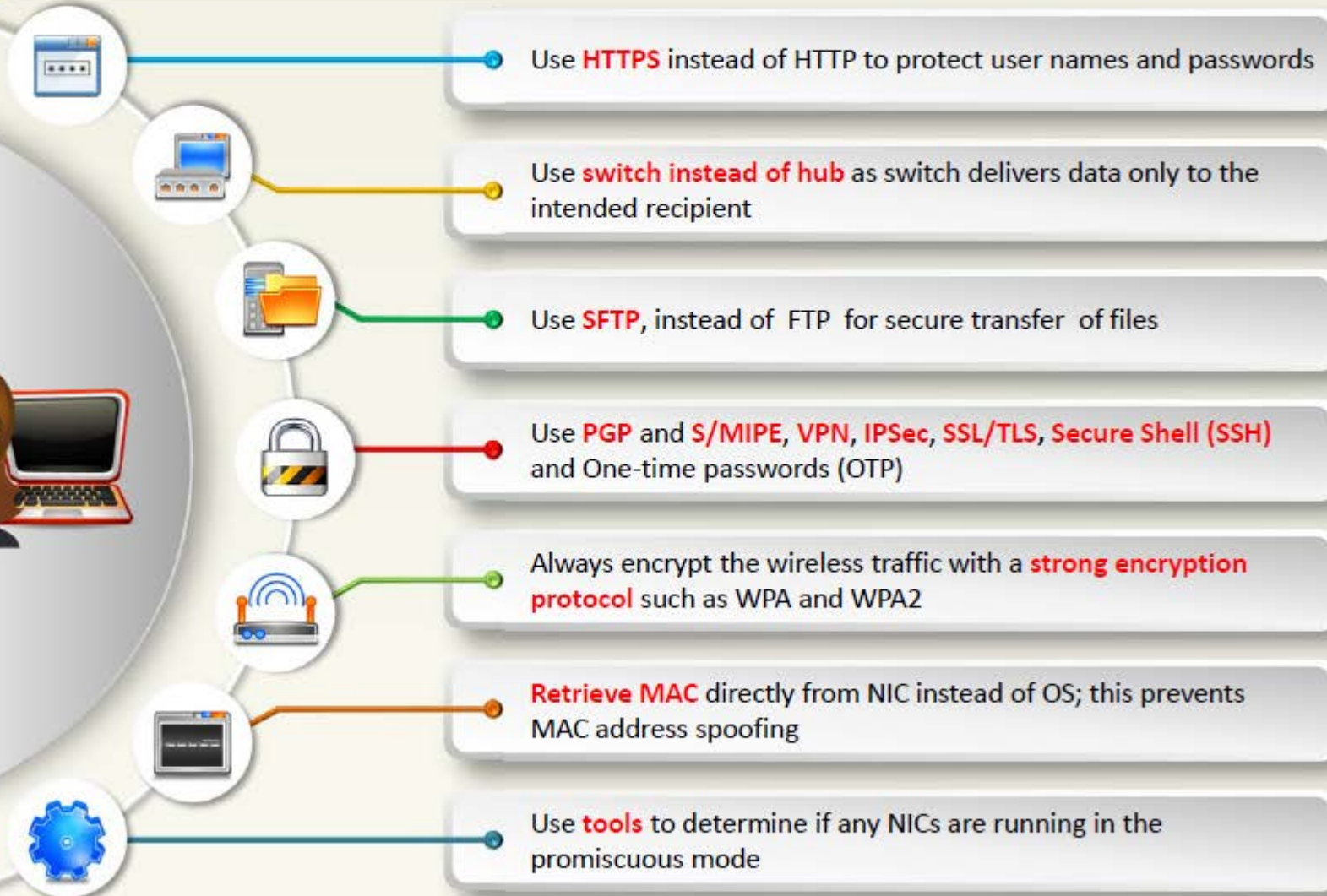
<http://faceniff.ponury.net>

# Module Flow



# How to Defend Against Sniffing

(Cont'd)



# Module Flow



# How to Detect Sniffing



## Promiscuous Mode

- You will need to **check which machines are running** in the promiscuous mode
- Promiscuous mode allows a network device to **intercept and read each network packet** that arrives in its entirety



## IDS

- **Run IDS** and notice if the **MAC address** of certain machines has changed (Example: router's MAC address)
- IDS can alert the administrator about **suspicious activities**

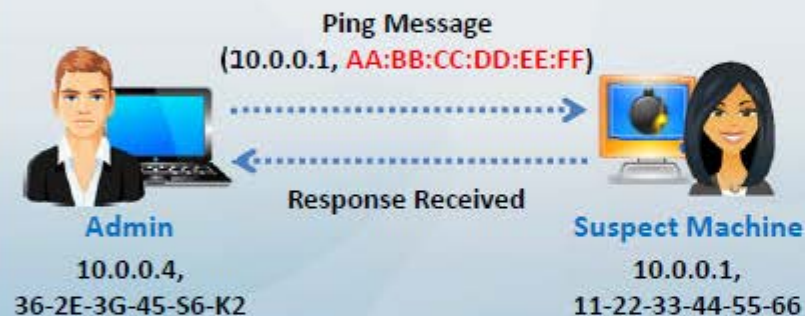


## Network Tools

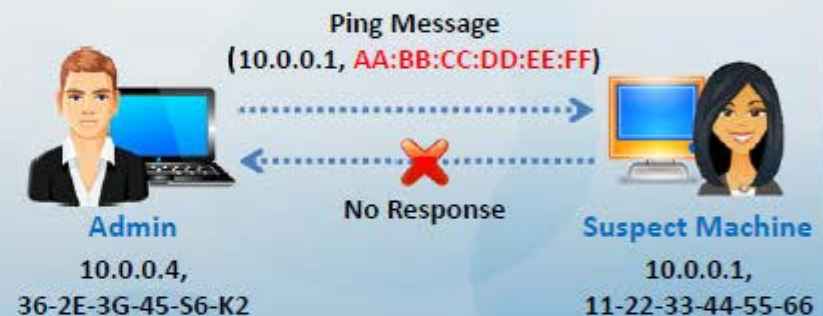
- Run network tools such as **Capsa Network Analyzer** to monitor the network for strange packets
- It enables you to **collect, consolidate, centralize** and **analyze traffic data** across different network resources and technologies

# Sniffer Detection Technique: Ping Method

## Promiscuous Mode

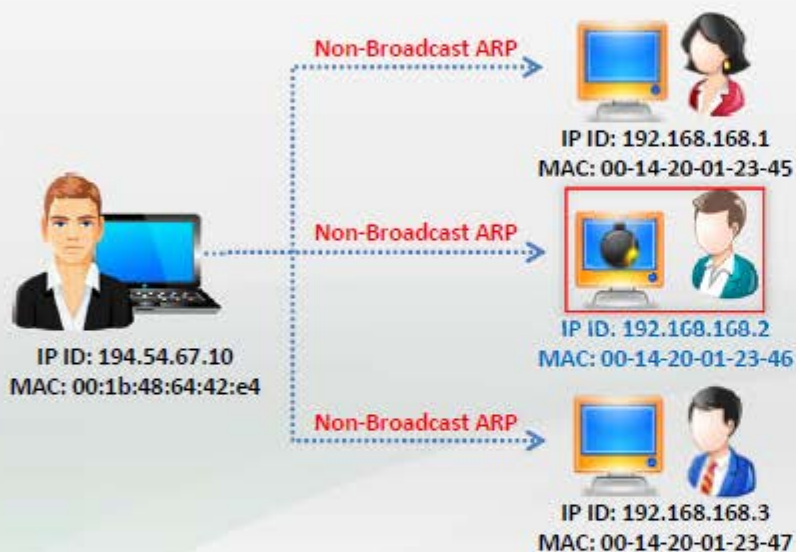


## Non-Promiscuous Mode

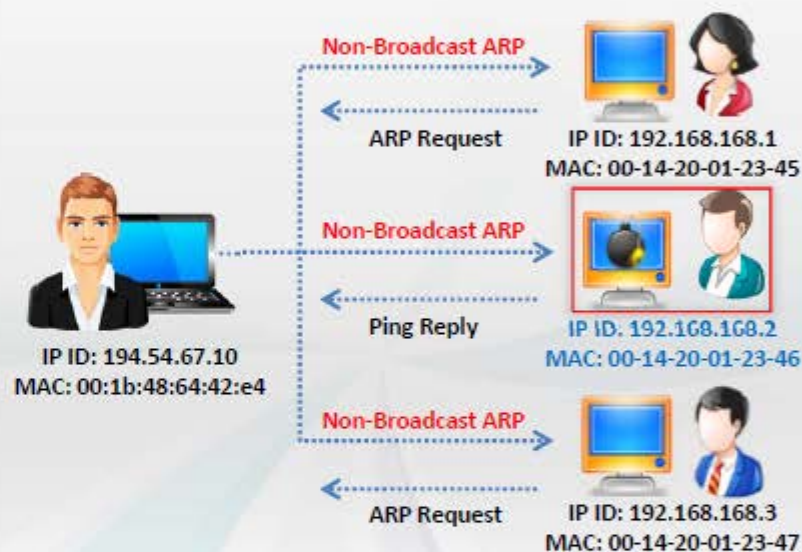


Send a ping request to the suspect machine with its IP address and **incorrect MAC address**. The Ethernet adapter rejects it, as the MAC address does not match, whereas the suspect machine running the **sniffer responds** to it as it does not reject packets with a different MAC address

# Sniffer Detection Technique: ARP Method



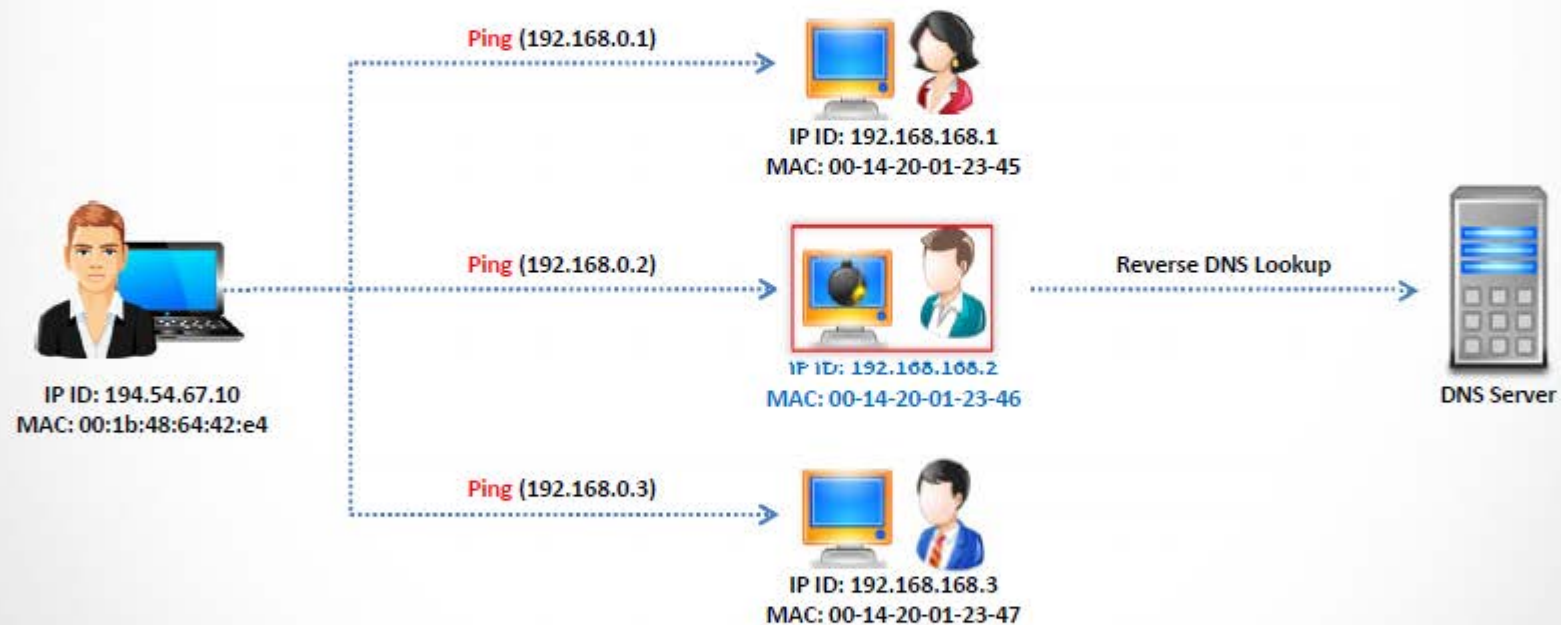
Only a machine in promiscuous mode (machine C) **caches the ARP information** (IP and MAC address mapping)



A machine in promiscuous mode **replies to the ping message** as it has correct information about the host sending **ping request** in its cache; rest of the machines will send ARP probe to identify the source of ping request

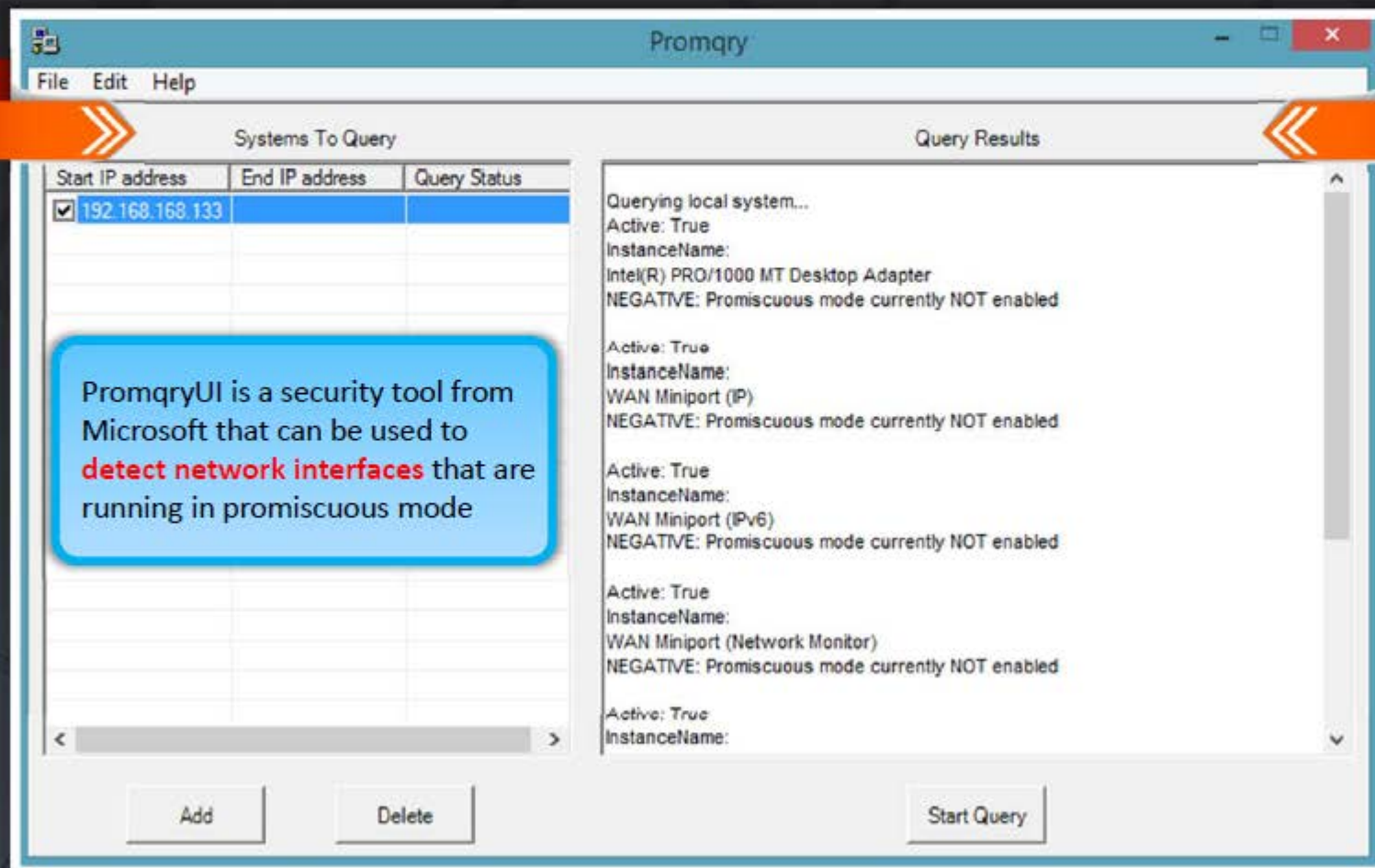
# Sniffer Detection Technique: DNS Method

Most of the sniffers perform **reverse DNS lookup** to identify the machine from the IP address



A machine generating **reverse DNS lookup traffic** will be most likely running a sniffer

# Promiscuous Detection Tool: PromqryUI



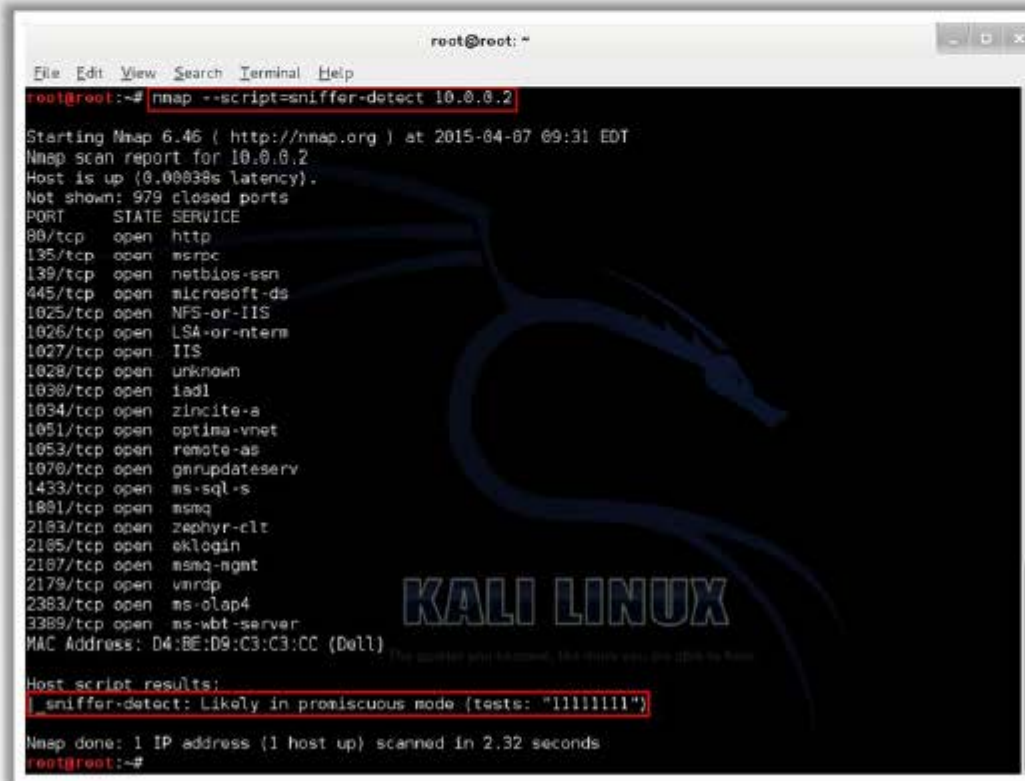
<http://www.microsoft.com>

# Promiscuous Detection Tool: Nmap



- Nmap's NSE script allows you to check if a target on a local Ethernet has its network card in **promiscuous** mode
- **Command to detect NIC in promiscuous mode:**

`nmap --script=sniffer-detect [Target IP Address/Range of IP addresses]`



```
root@root: ~  
File Edit View Search Terminal Help  
root@root:~# nmap --script=sniffer-detect 10.0.0.2  
  
Starting Nmap 6.46 ( http://nmap.org ) at 2015-04-07 09:31 EDT  
Nmap scan report for 10.0.0.2  
Host is up (0.00038s latency).  
Not shown: 979 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
135/tcp   open  msrpc  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
1025/tcp  open  NFS-or-IIS  
1026/tcp  open  LSA-or-nterm  
1027/tcp  open  IIS  
1028/tcp  open  unknown  
1030/tcp  open  iadl  
1034/tcp  open  zincite-a  
1051/tcp  open  optima-vnet  
1053/tcp  open  remote-as  
1070/tcp  open  gnupdateserv  
1433/tcp  open  ms-sql-s  
1801/tcp  open  asmq  
2103/tcp  open  zephyr-clt  
2105/tcp  open  eklogin  
2107/tcp  open  asmq-ngnt  
2179/tcp  open  vnrpd  
2383/tcp  open  ms-olap4  
3389/tcp  open  ms-wbt-server  
MAC Address: D4:BE:D9:C3:C3:CC (Dell)  
  
Host script results:  
_ sniffer-detect: Likely in promiscuous mode (tests: "11111111")  
  
Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds  
root@root:~#
```

# Module Flow



# Sniffing Pen Testing

- Sniffing pen test is used to check if the **data transmission** from an organization is **secure from sniffing and interception attacks**
- Sniffing pen test helps administrators to:



**Audit the network traffic** for malicious content



**Implement security mechanism** such as SSL and VPN to secure the network traffic



**Identify rogue sniffing application** in the network



**Discover rogue DHCP and DNS servers** in the network

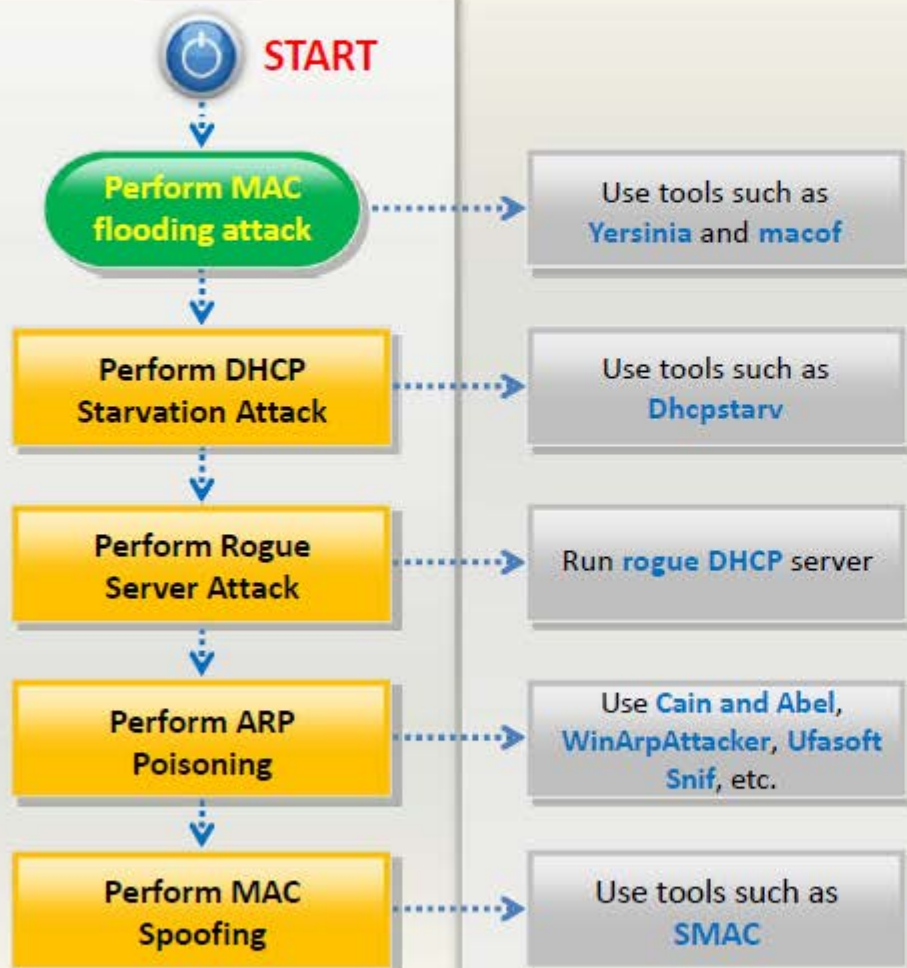


Discover the presence of **unauthorized networking devices**



# Sniffing Pen Testing

(Cont'd)

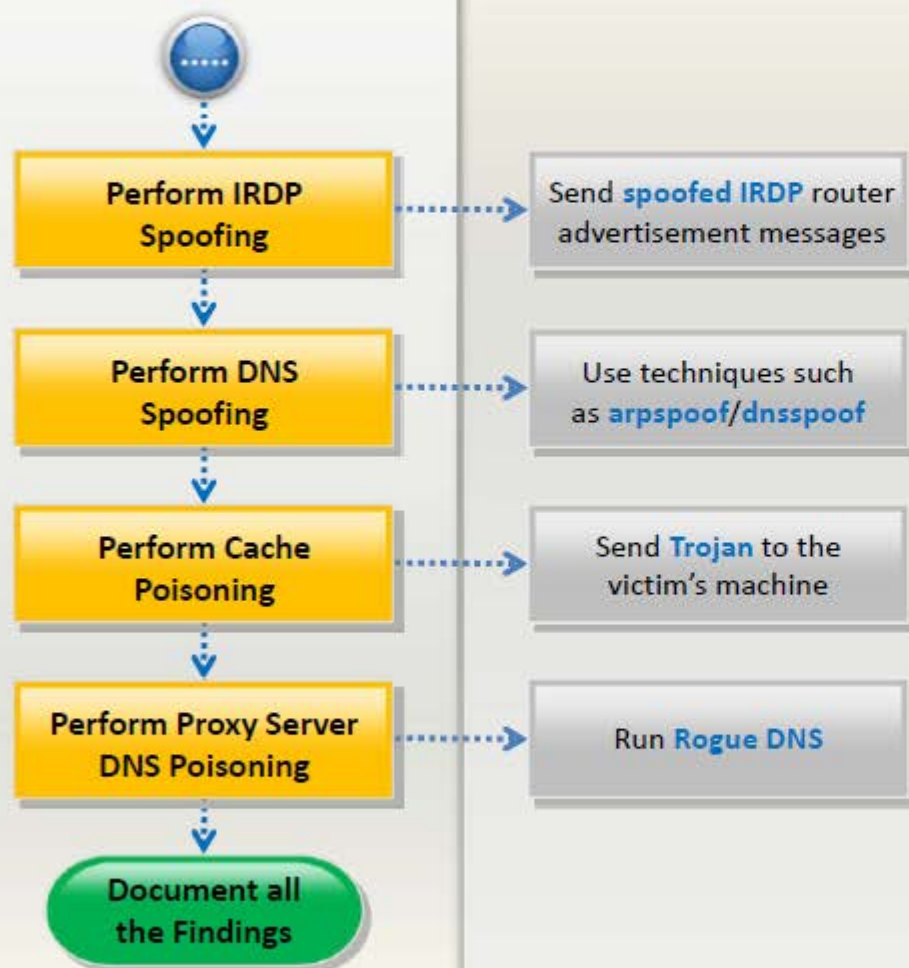


- Perform MAC flooding attack using tools such as **Yersinia** and **macof**
- Perform DHCP starvation attack using tools such as **Dhcpstarv** and **Yersinia**
- Perform rogue server attack by running **rogue DHCP server** in the network and responding to DHCP requests with **bogus IP addresses**
- Perform ARP poisoning using tools such as **Cain & Abel**, **WinArpAttacker**, **Ufasoft Snif**, etc.
- Perform MAC spoofing using tools such as **SMAC**



# Sniffing Pen Testing

(Cont'd)



- Perform IRDP spoofing by sending **spoofed IRDP router advertisement messages**
- Perform DNS spoofing using techniques such as **arpspoof/dnsspoof**
- Perform cache poisoning by sending **Trojan** to the victim's machine that changes proxy server settings in IE to that of attackers, thus redirecting to fake website
- Perform proxy server DNS poisoning by running **rogue DNS**

# Module Summary



- ☐ By placing a packet sniffer in a network, attackers can capture and analyze all the network traffic
- ☐ Attackers can sniff confidential information such as email and chat conversations, passwords, and web traffic
- ☐ Sniffing is broadly categorized as passive and active; passive sniffing refers to sniffing from a hub-based network, whereas active sniffing refers to sniffing from a switch-based network
- ☐ Networking layers in the OSI model are designed to work independently of each other; if a sniffer sniffs data in the Data Link layer, the upper OSI layer will not be aware of the problem
- ☐ Attackers use MAC attacks, DHCP attacks, ARP poisoning attacks, spoofing attacks, and DNS poisoning techniques to sniff network traffic
- ☐ Major countermeasures for sniffing include using static IP addresses and static ARP tables, and using encrypted sessions such as SSH instead of Telnet, Secure Copy (SCP) instead of FTP, SSL for data transmission