



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

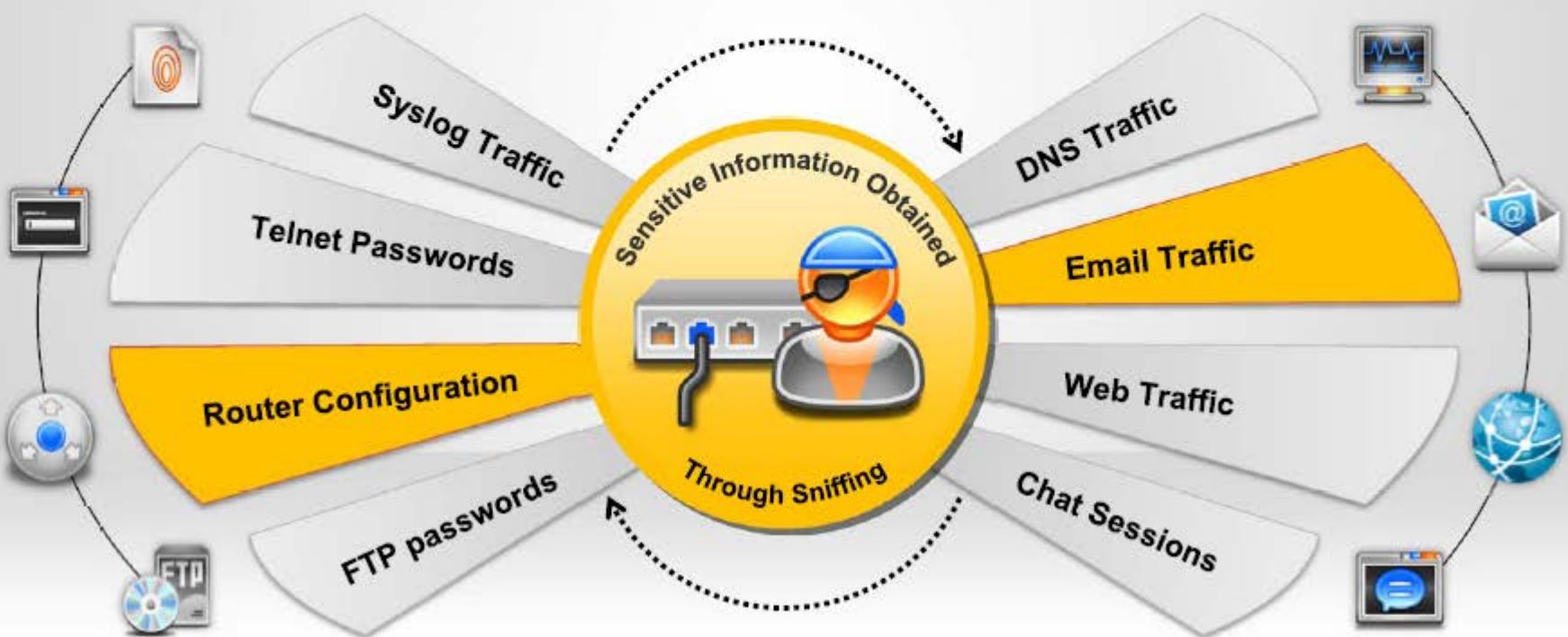


Network Sniffing and Threats

CEH
Certified Ethical Hacker

- 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



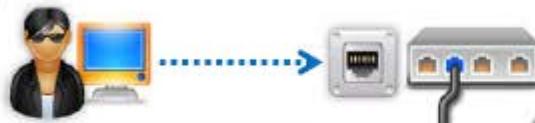
6

Spoofing Attack

How an Attacker Hacks the Network Using Sniffers



An attacker connects his laptop to a switch port



He runs discovery tools to learn about network topology



He identifies victim's machine to target his attacks



He poisons the victim machine by using ARP spoofing techniques



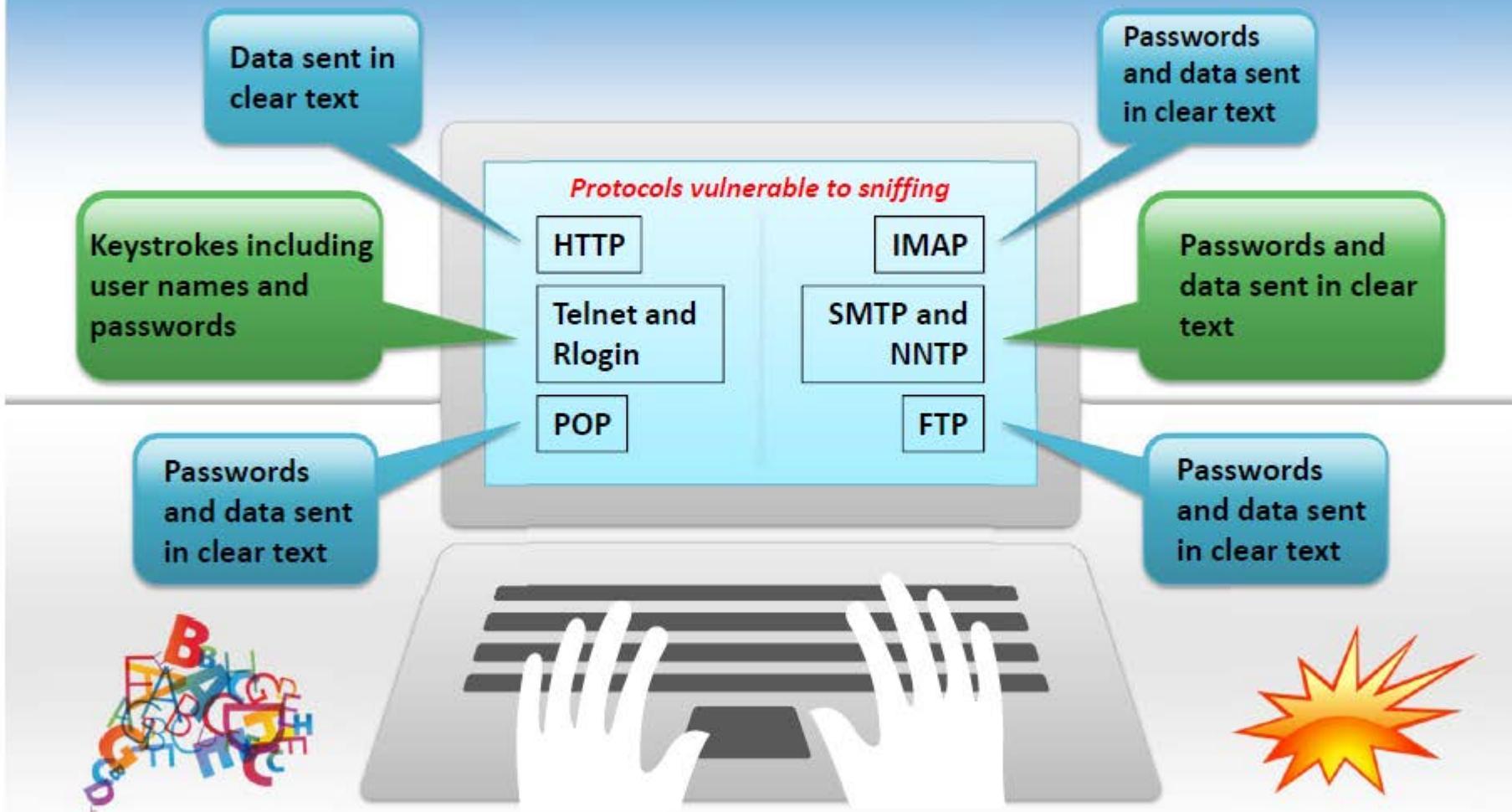
The traffic destined for the victim machine is redirected to the attacker



The hacker extracts passwords and sensitive data from the redirected traffic



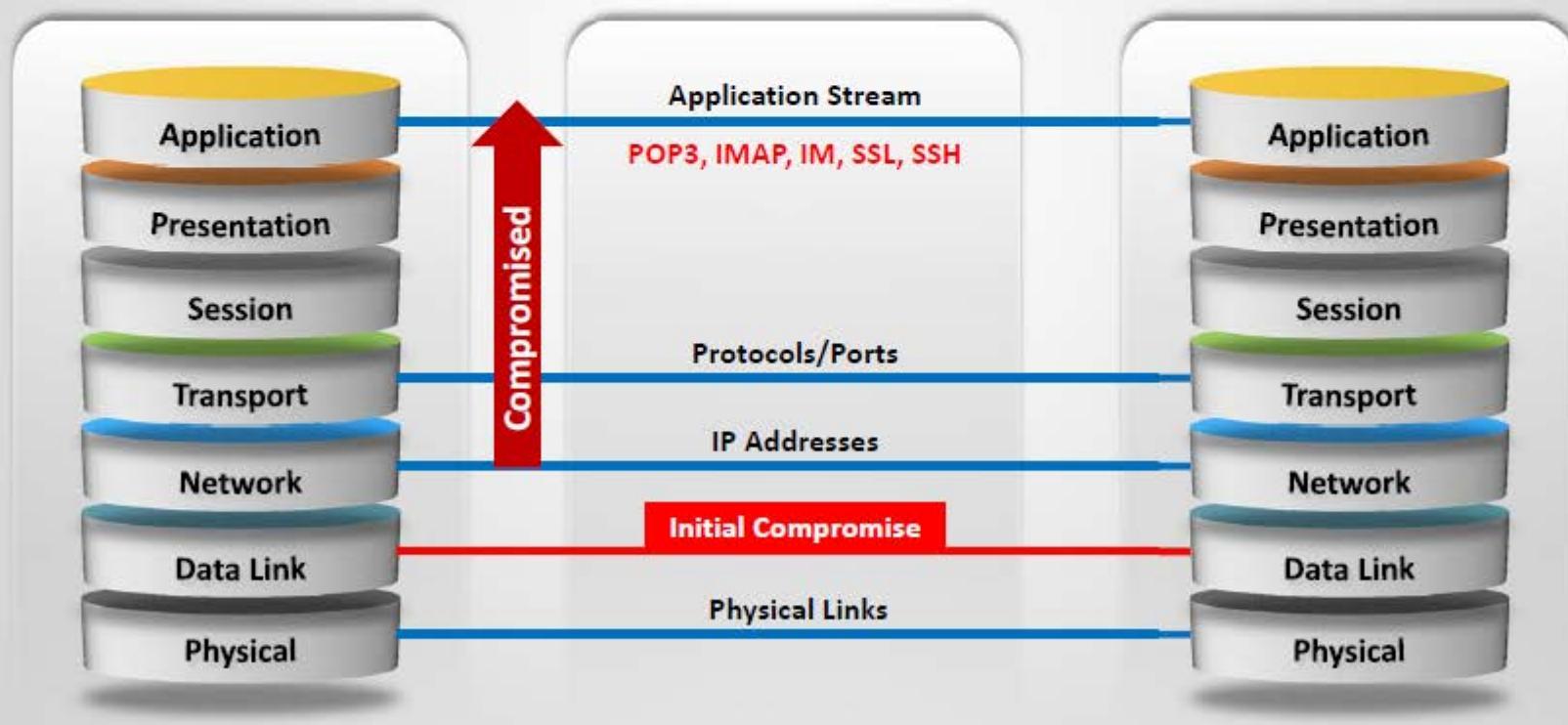
Protocols Vulnerable to Sniffing



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 AnalyzerFLUKE Networks OptiView® XG
Network AnalyzerFLUKE 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

Types of Wiretapping



Active Wiretapping

It monitors, records, alters and also injects something into the communication or traffic

Passive Wiretapping

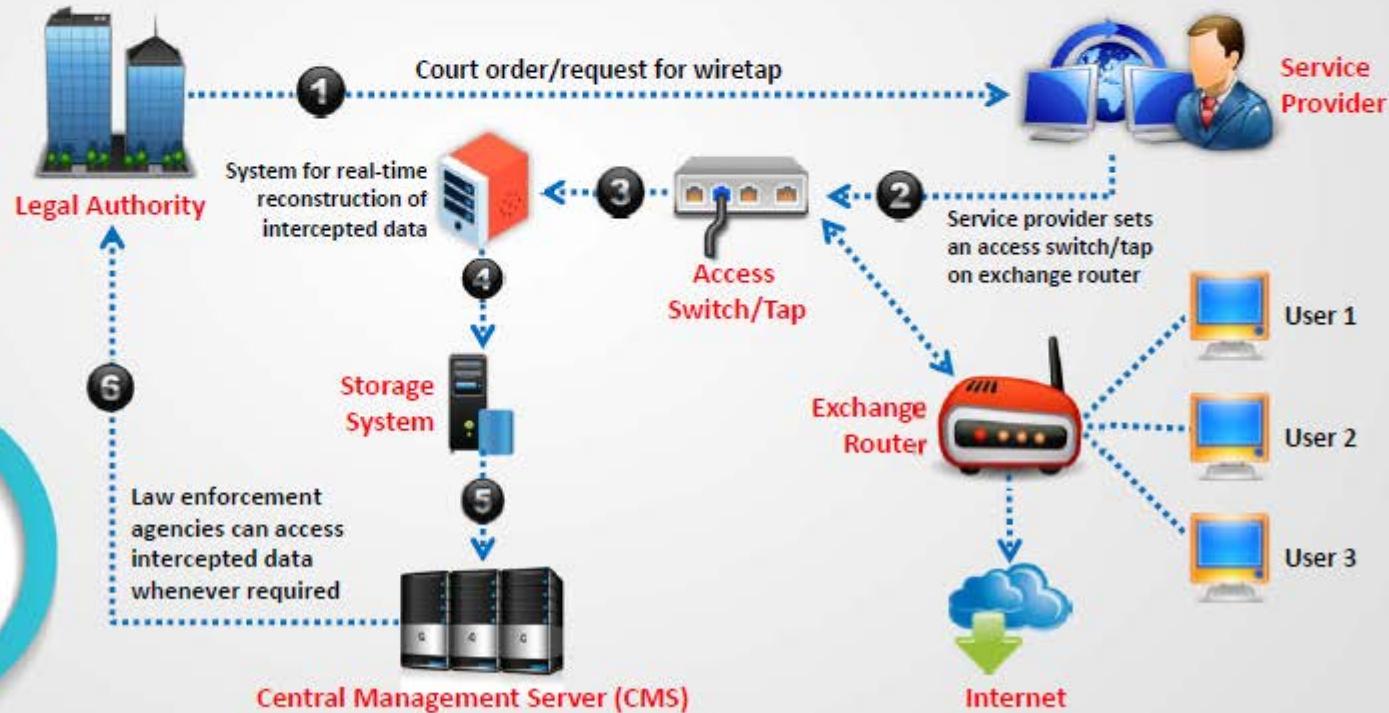
It only monitors and records the traffic and gain knowledge of the data it contains

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

Lawful Interception



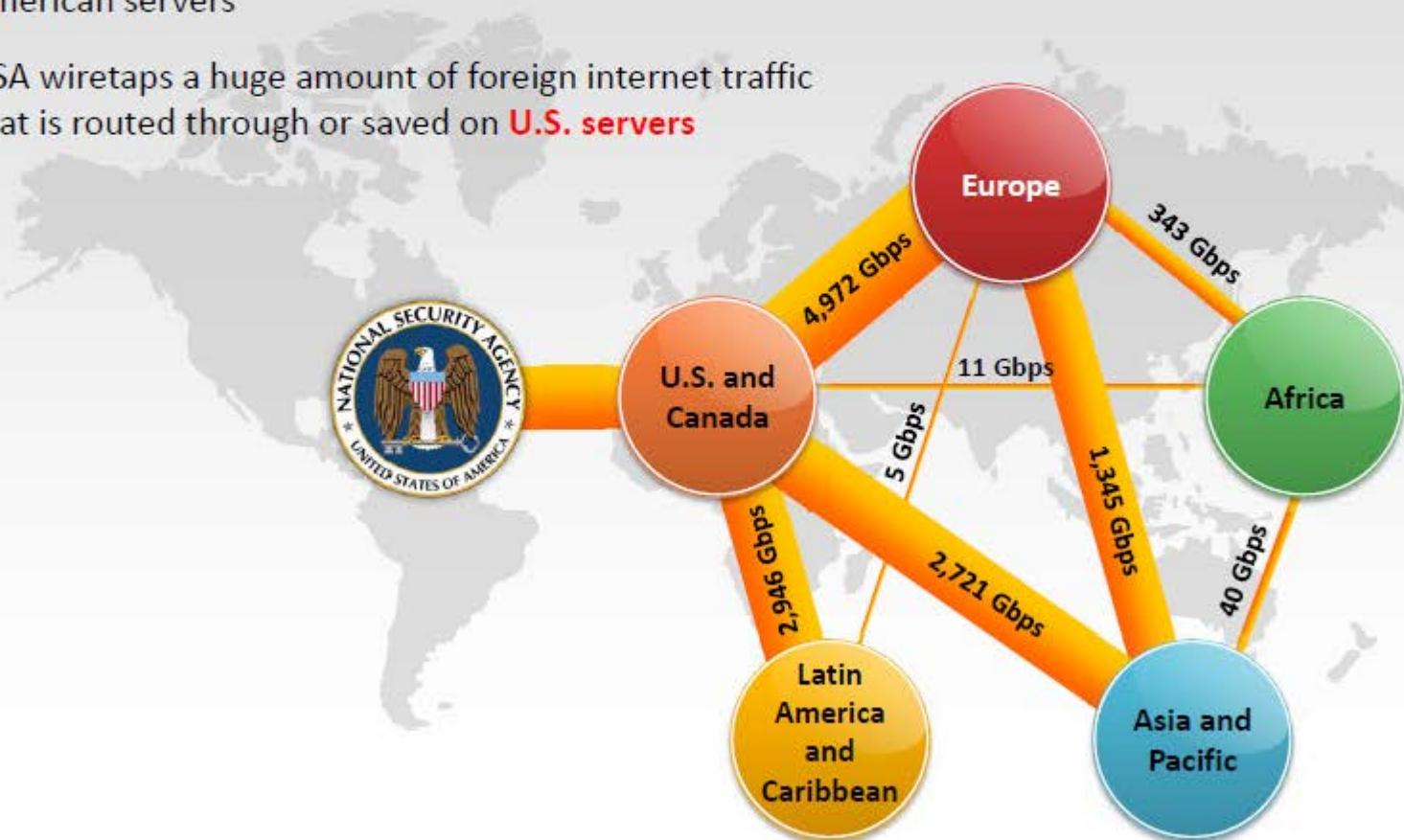
Lawful interception refers to legally **intercepting data communication** between two end points for surveillance on the traditional telecommunications, VoIP, data, and multiservice networks



Wiretapping Case Study: PRISM



- PRISM stands for "**Planning Tool for Resource Integration, Synchronization, and Management**," 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

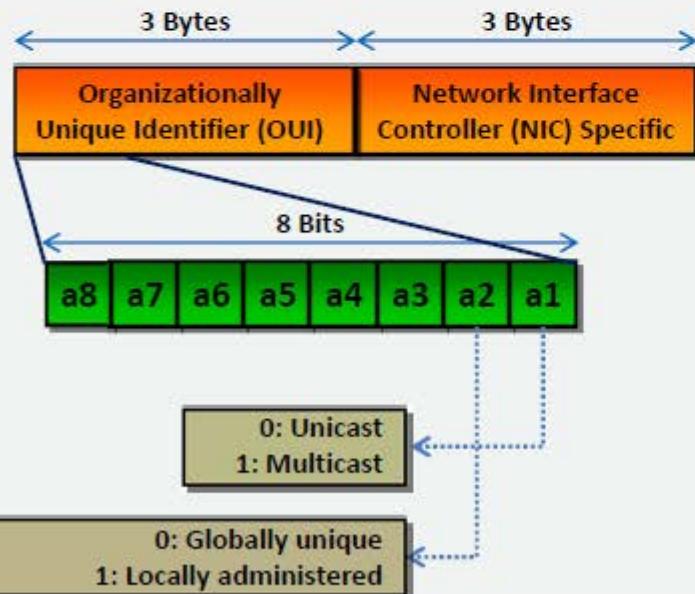


Each switch has a **fixed size dynamic Content Addressable Memory (CAM) table**



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

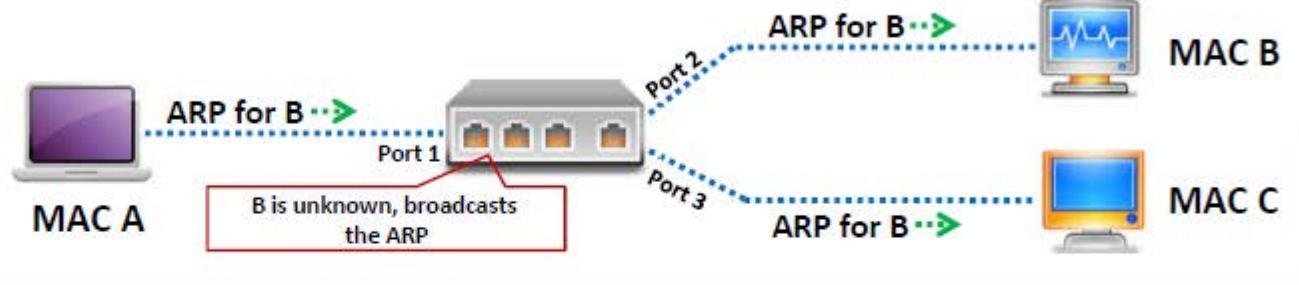


How CAM Works



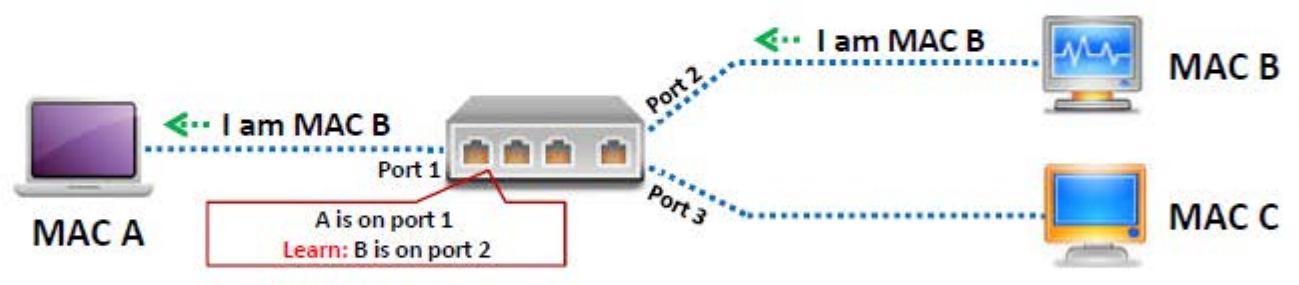
MAC	PORT
A	1
C	3

CAM Table



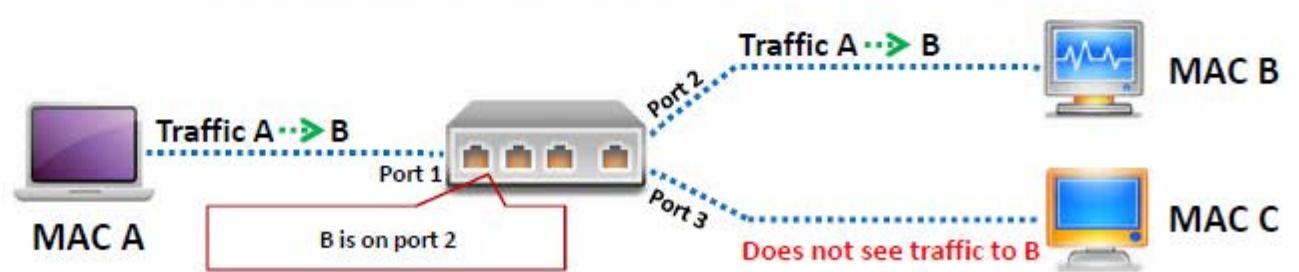
MAC	PORT
A	1
B	2
C	3

CAM Table



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**



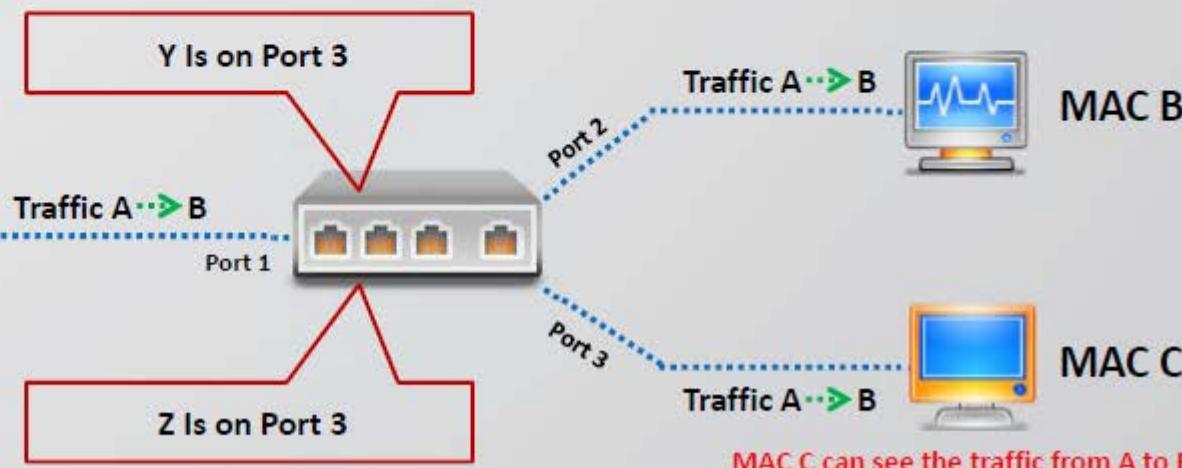
This will **change the behavior of the switch** to reset to it's learning mode, broadcasting on every port similar to a hub



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



MAC	PORT
Y	3
Z	3
C	3



MAC Flooding



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



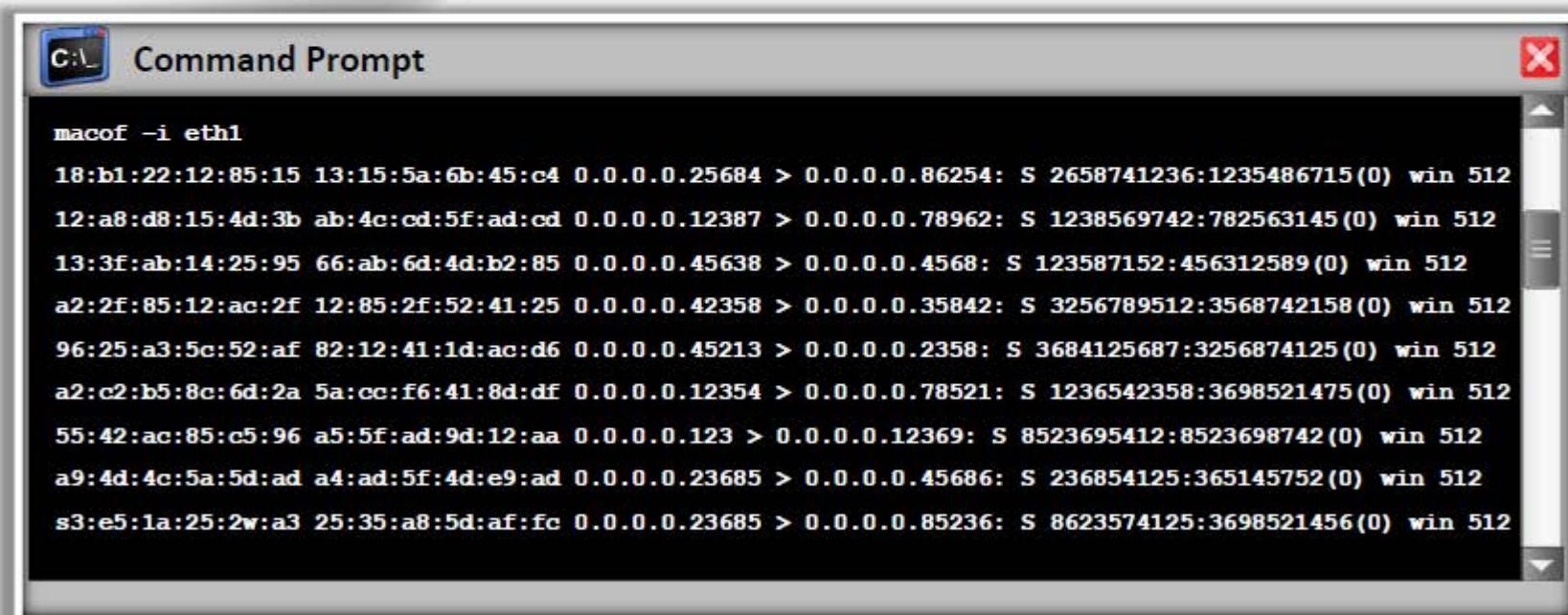
Switch then **acts as a hub** by broadcasting packets to all machines on the network and attackers can sniff the traffic easily



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



```
C:\ 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>

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

Switch Port Stealing



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

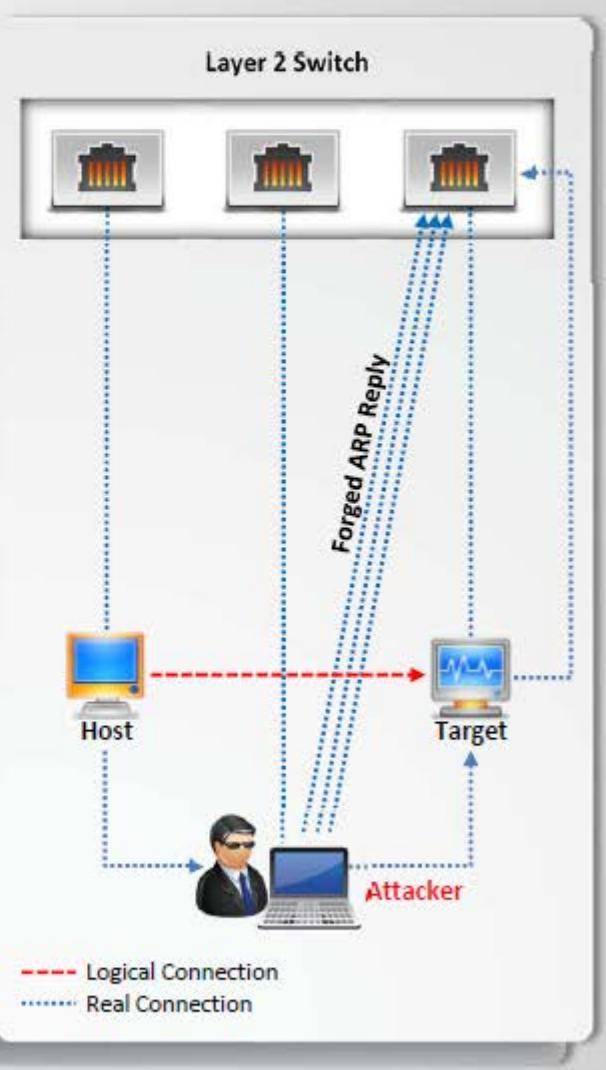
Attacker floods the switch with **forged gratuitous ARP packets** with target MAC address as source and his own MAC address as destination

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

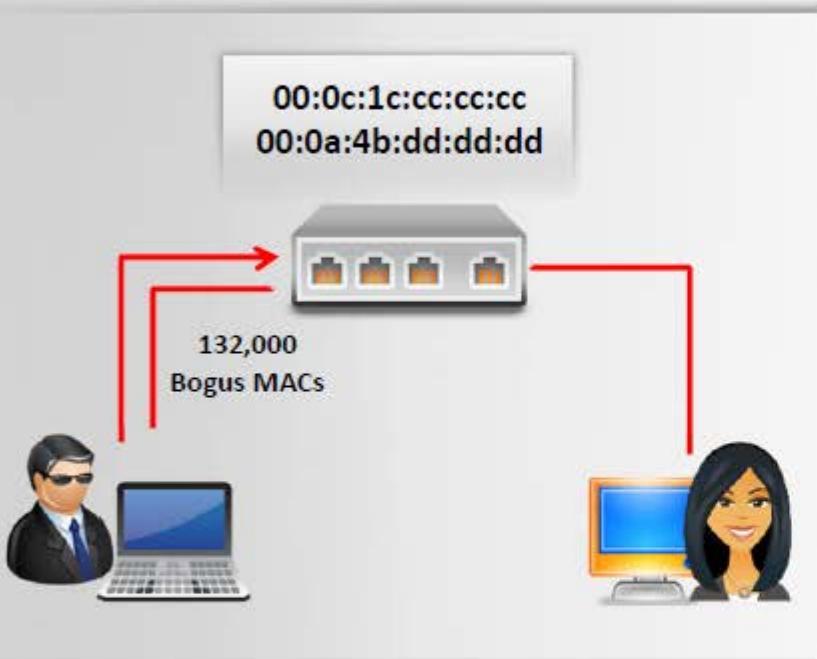
In such case if attacker is fast enough, he will able to **direct the packets** intended for the target host toward his switch port

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

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

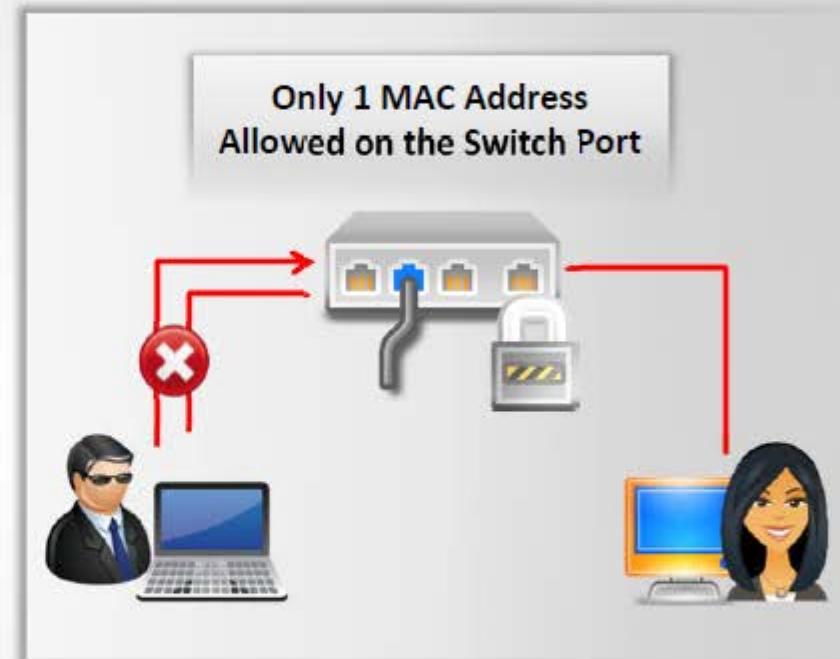


How to Defend against MAC Attacks



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

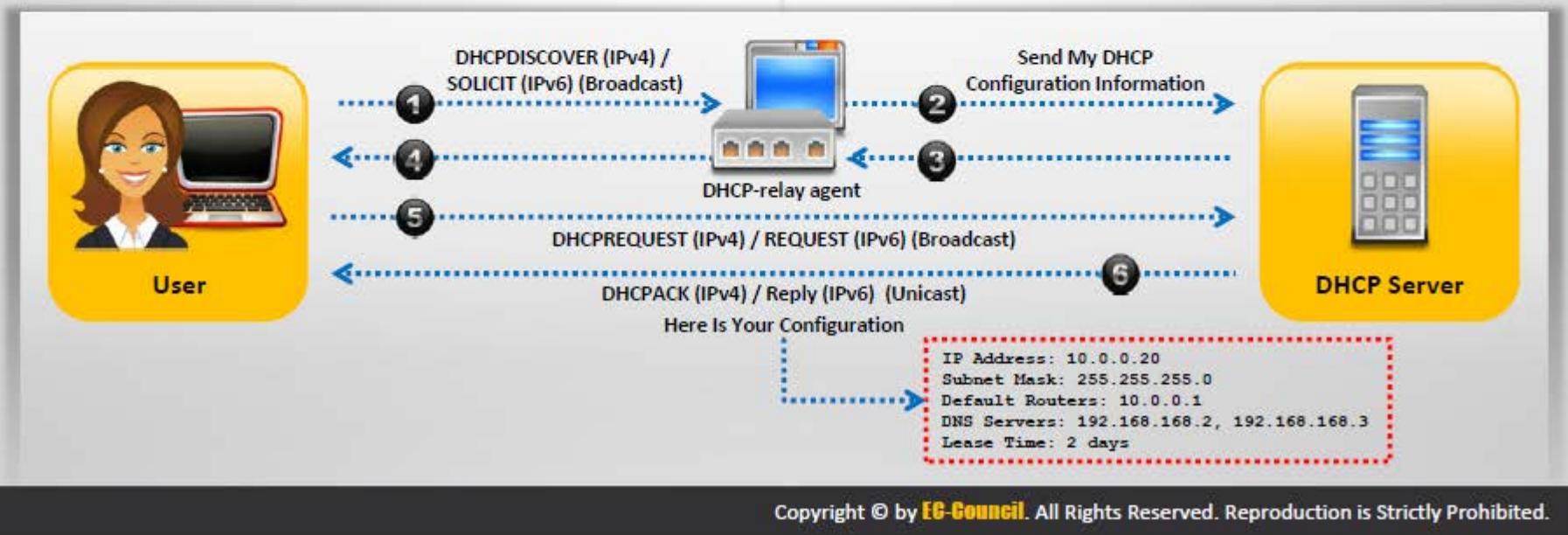


How DHCP Works



- 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
DHCPIinform	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 has expired

IPv4 DHCP Packet Format

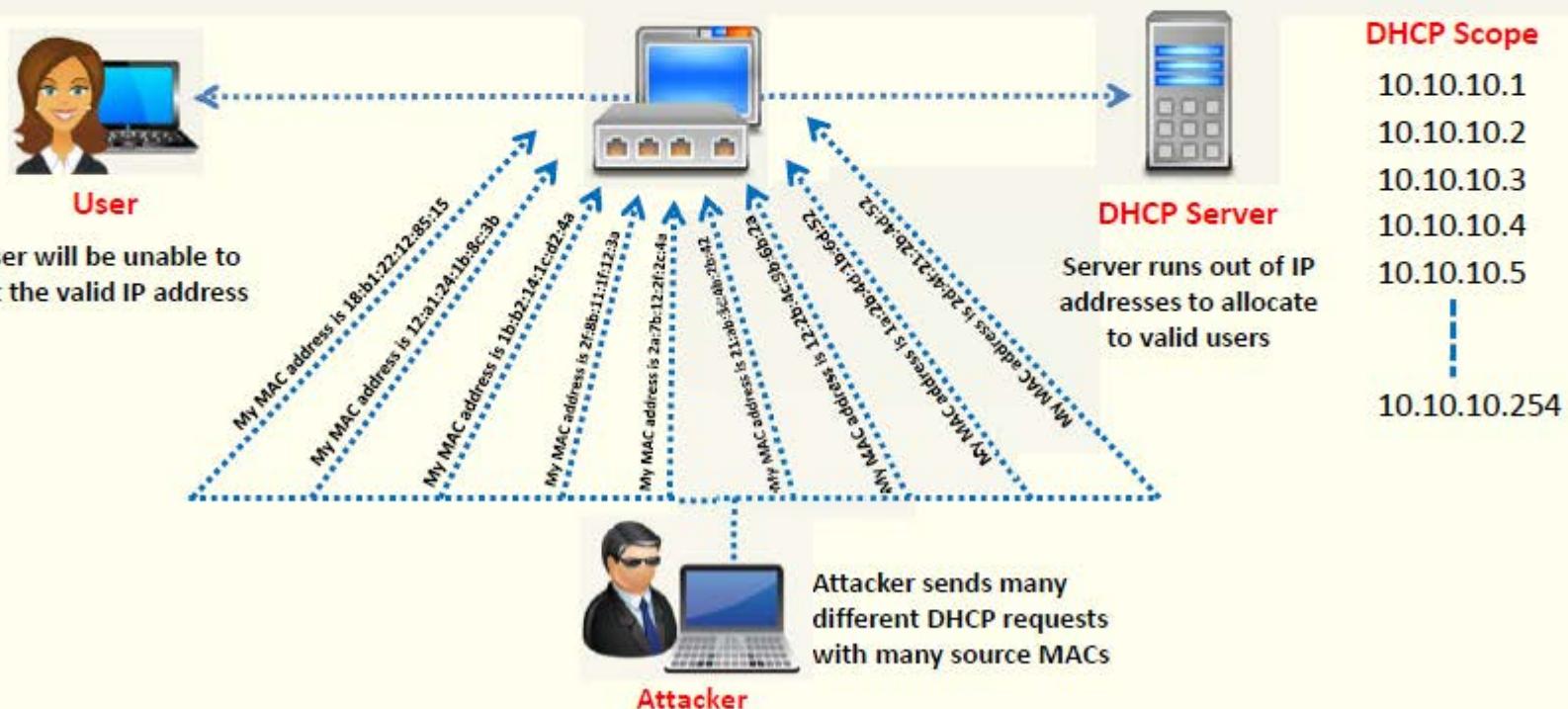


OP Code	Hardware Type	Hardware Length	HOPS
Transaction ID (XID)			
Seconds		Flags	
Client IP Address (CIADDR)			
Your IP Address (YIADDR)			
Server IP Address (SIADDR)			
Gateway IP Address (GIADDR)			
Client Hardware Address (CHADDR)—16 bytes			
Server Name (SNAME)—64 bytes			
Filename—128 bytes			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

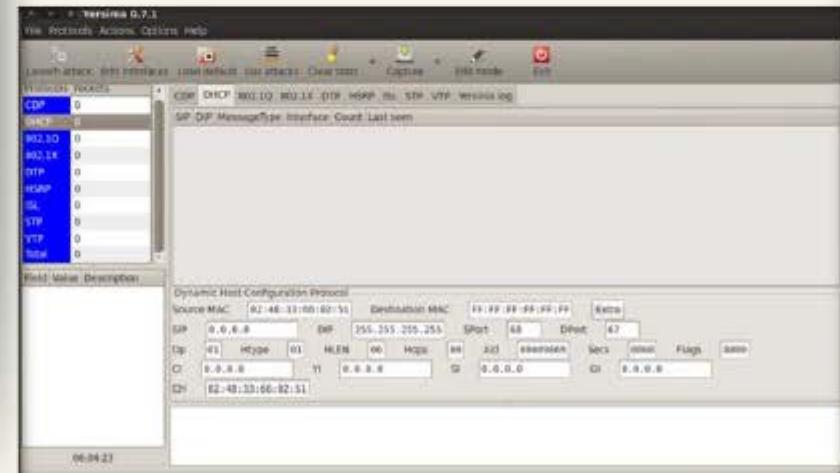
- dhcpstarv implements DHCP starvation attack. It requests **DHCP leases** on specified interface, saves them, and renews on regular basis



<http://dhcpstarv.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**



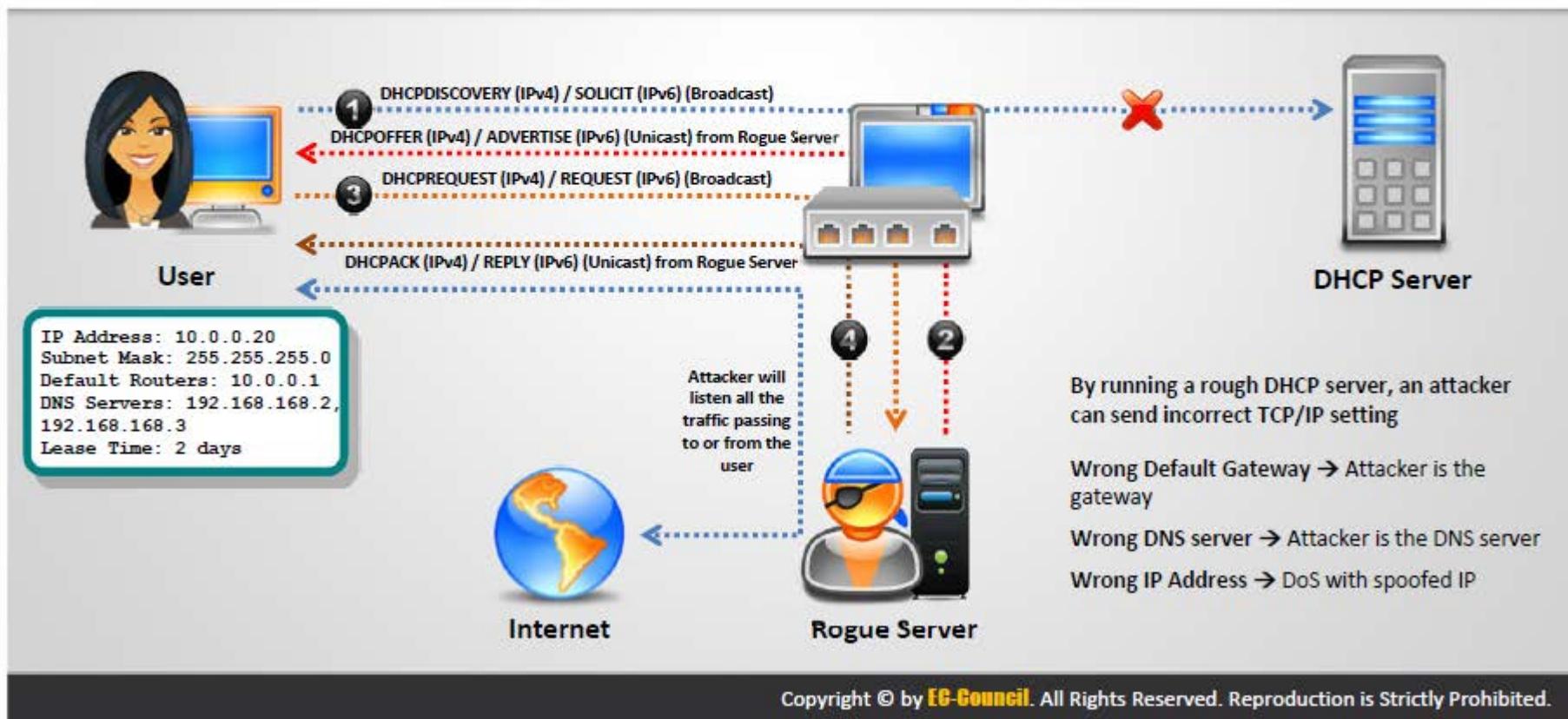
<http://www.yersinia.net>

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

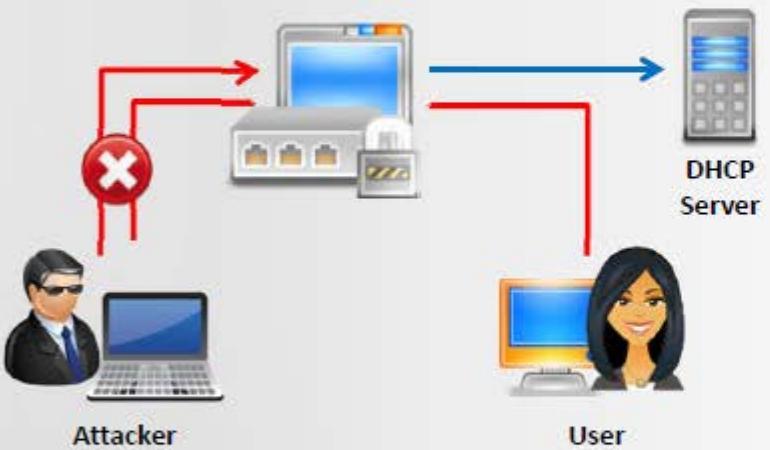


How to Defend Against DHCP Starvation and Rogue Server Attack



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

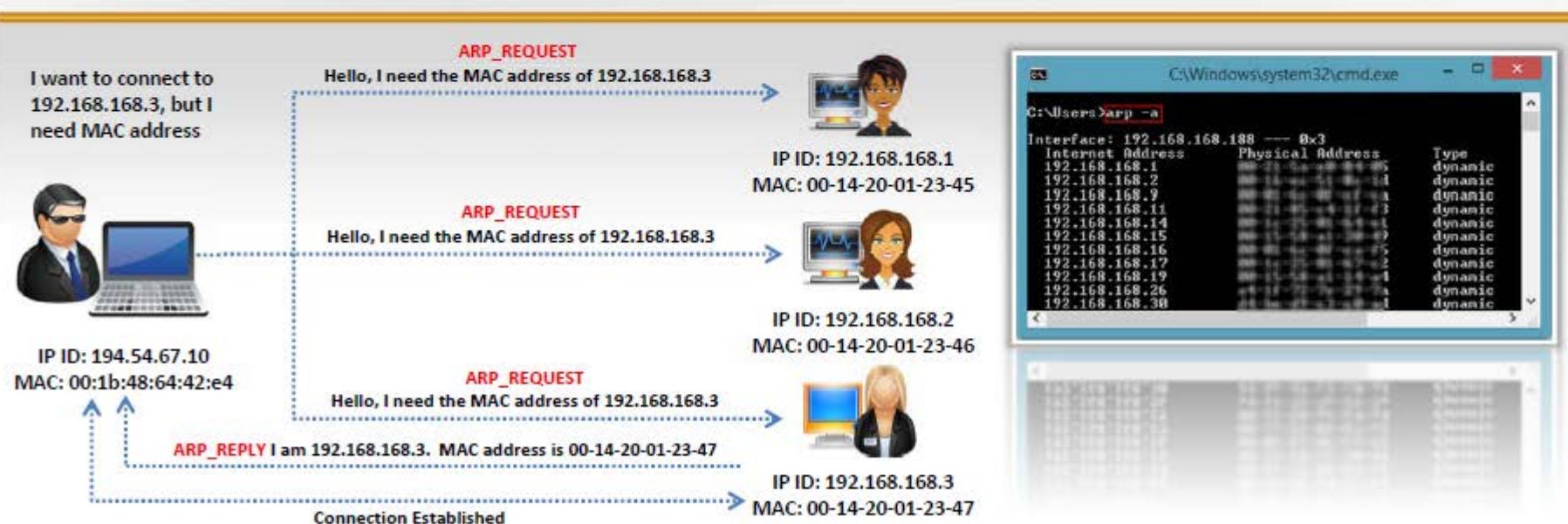
Module Flow



What Is Address Resolution Protocol (ARP)?



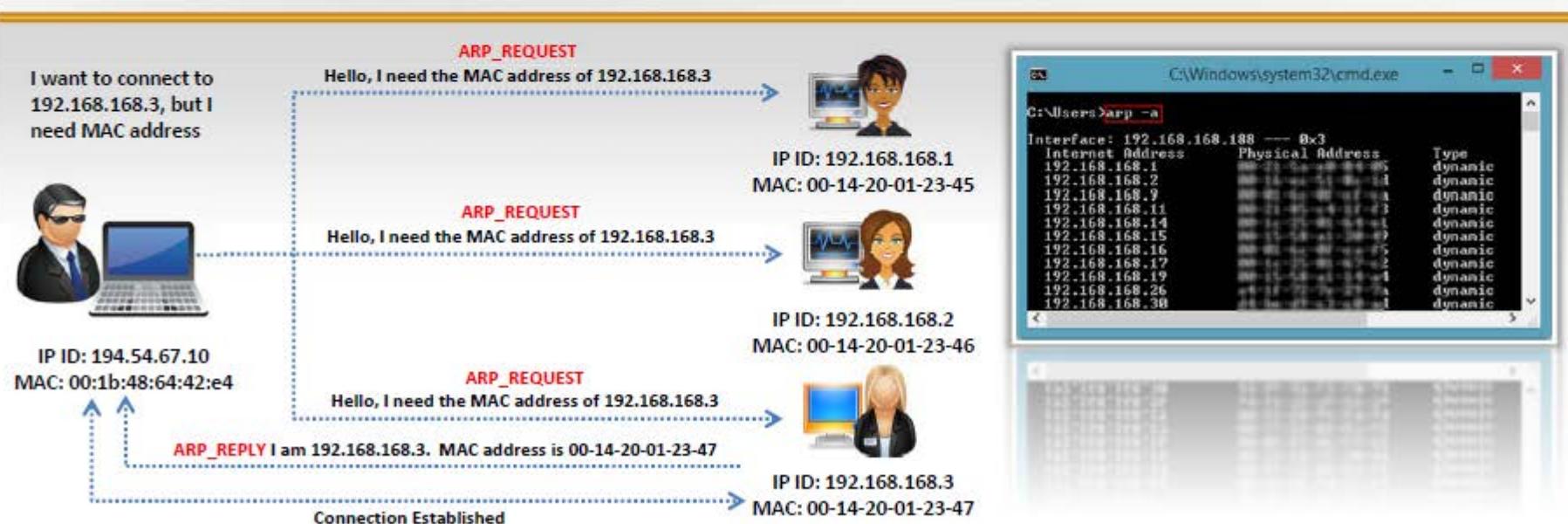
- Address Resolution Protocol (ARP) is a stateless protocol used for **resolving IP addresses to machine (MAC) addresses**
- All network devices (that needs to communicate on the network) broadcasts ARP queries in the network to find out other **machines' MAC addresses**
- When one machine needs to communicate 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



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



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

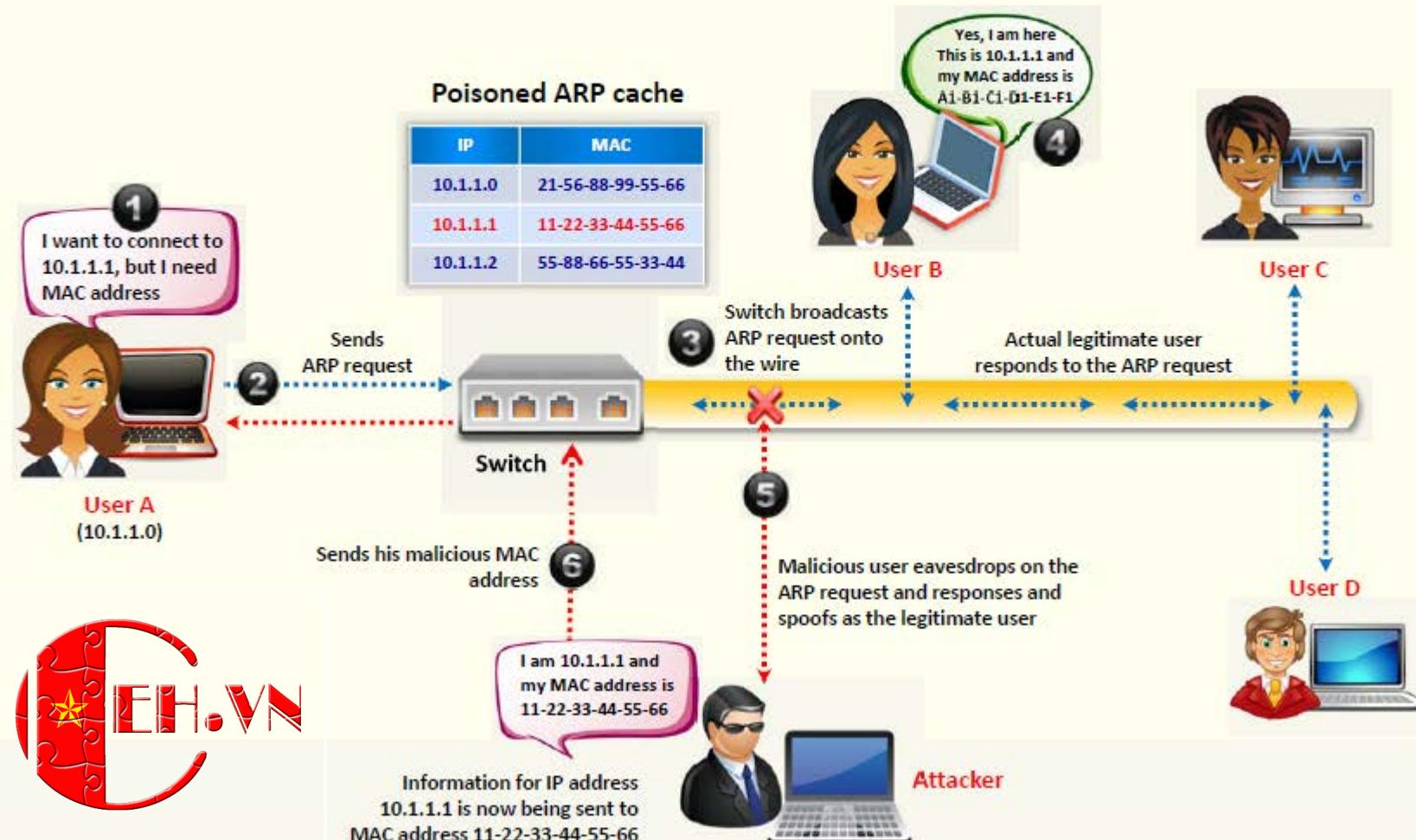


Switch is set in '**forwarding mode**' after ARP table is flooded with spoofed ARP replies and attackers can sniff all the network packets



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



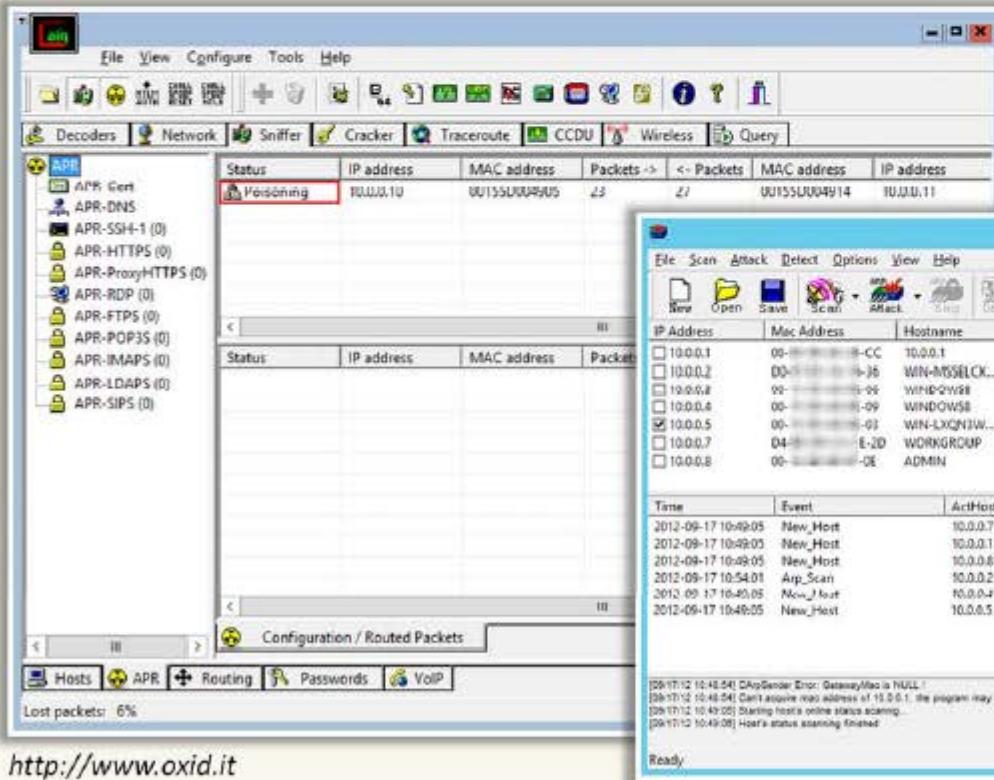
Denial-of-Service (DoS) Attack

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



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

Ufasoft Packet Snif

File Edit View Capture Tools Help

Plugins ARP-spoofing

Order	Timestamp	Length	Summary
7633	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294206764:4294206767
7634	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294206764:42942062
7635	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294208234:42942096
7636	10/17/2013 14...	54	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [L] ack 4294209664, win 1990
7637	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294210444:42942111
7638	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
7639	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294212004:42942140
7660	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294214054:42942155
7661	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294215524:42942169
7662	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294216884:42942164
7663	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294218444:42942199
7664	10/17/2013 14...	54	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [L] ack 4294219904, win 1095
7665	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294219904:42942191
7666	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294220421184:42942222
7667	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294222524:42942242
7668	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294224234:42942257
7669	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294225744:42942272
7670	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294227034:42942286
7671	10/17/2013 14...	54	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [L] ack 4294222716, win 1095
7672	10/17/2013 14...	54	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [L] ack 4294228664, win 1095
7673	10/17/2013 14...	171	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [P,L] seq 4294996748:42949966
7674	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294228664:42942301
7675	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294230124:42942315
7676	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294231584:42942330
7677	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294233044:42942343
7678	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294234044:42942349

Source 021:Sand045
Destination 0:27:73:24:9F:09
Broadcast 0:0:0:0:0:0

Ufasoft Packet Snif

File Edit View Capture Tools Help

Plugins ARP-spoofing

Order	Timestamp	Length	Summary
6053	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294209406:4294209406
6054	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294209406:4294209406
6055	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294209406, win 1095
6056	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
6057	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
6058	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
6059	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
6060	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294211444:42942126
6061	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294211444, win 1095
6062	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294213444:42942146
6063	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294213444:42942146
6064	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294213444, win 1095
6065	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294215444:42942166
6066	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294215444:42942166
6067	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294215444, win 1095
6068	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294217444:42942186
6069	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294217444:42942186
6070	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294217444, win 1095
6071	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294219444:42942206
6072	10/17/2013 14...	54	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] ack 4294219444, win 1095
6073	10/17/2013 14...	171	IP 192.168.168.188.1405 > 192.168.168.1.445: Flags [P,L] seq 4294996748:42949966
6074	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294228664:42942301
6075	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294230124:42942315
6076	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294231584:42942330
6077	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294233044:42942343
6078	10/17/2013 14...	1514	IP 192.168.168.1.445 > 192.168.168.188.1405: Flags [L] seq 4294234044:42942349

Source 021:Sand045
Destination 0:0:0:0:0:0

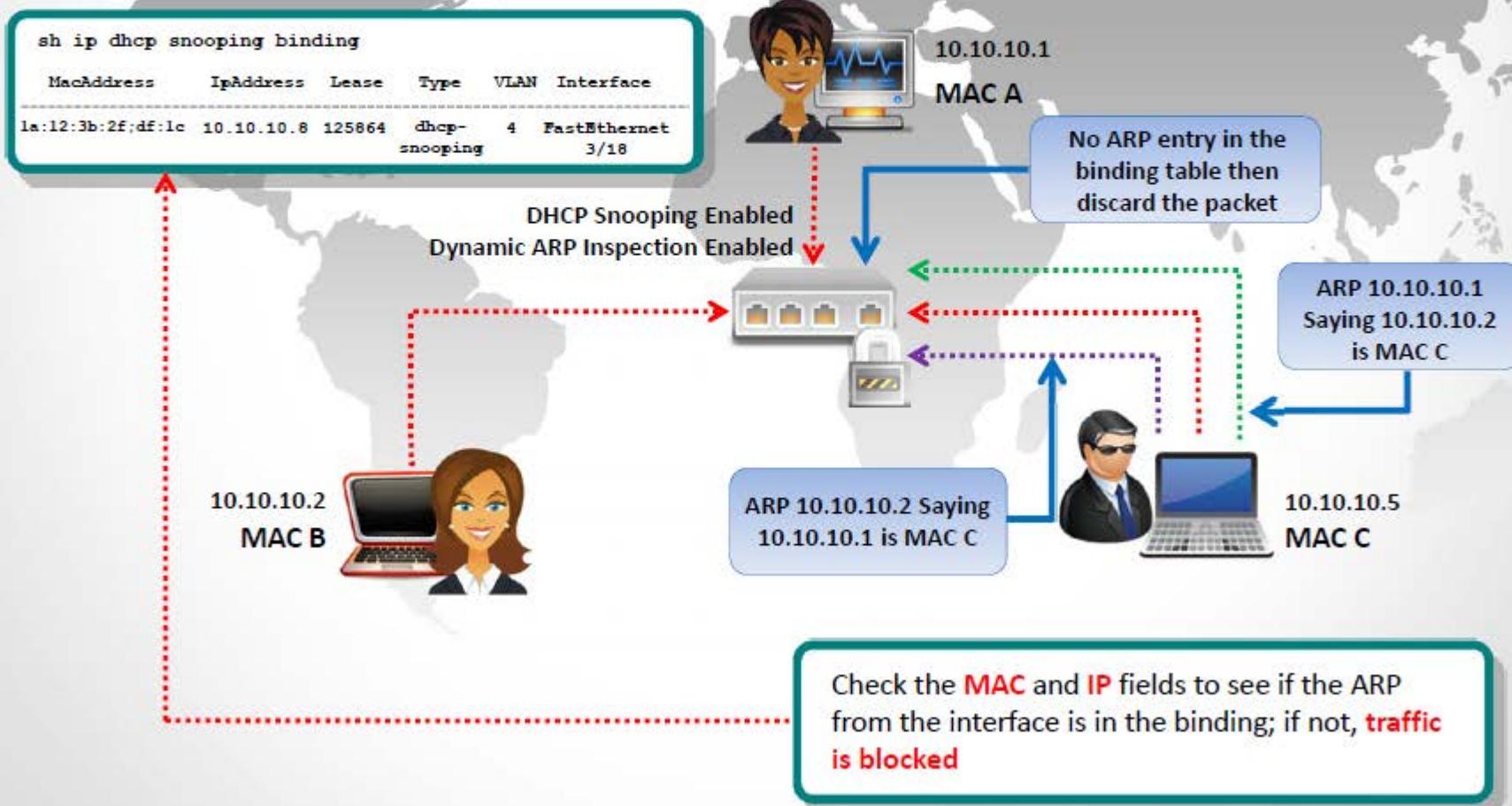
<http://ufasoft.com>

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



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)
-----	-----	-----

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 0
```

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-    4  FastEthernet
                           snooping          0/3
Total number of bindings: 1
```

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



- 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

XArp - unregistered version

Status: ARP attacks detected!

Security level set to: aggressive

IP	MAC	Host	Vendor	Interface	Online	Cache	First seen
192.168.168.83	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.87	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.99	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.100	00-...-00-00-00-00	Windows	0 Foxconn	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.101	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.110	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.111	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.113	d4-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.133	00-...-00-00-00-00	Windows	Micro-star Int'l...	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.144	00-...-00-00-00-00	Windows	4 Netgear, Inc.	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.153	a4-...-00-0c-00-00	Windows	3 unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.168	00-...-00-00-00-00	Windows	3 Sonicwall	0x3 - Intel(R) P...	yes	yes	10/17/2013
192.168.168.188	08-...-00-0c-00-00	Windows	Cadmus Com...	0x3 - Intel(R) P...	yes	no	10/17/2013
192.168.168.189	f0-...-00-0c-00-00	Windows	unknown	0x3 - Intel(R) P...	yes	yes	10/17/2013

XArp 2.2.2 - 35 mappings - 1 interface - 2 alerts

OK

< Alert 1 of 2 >

10/17/2013 15:32:55

DirectedRequestFilter: targeted request, destination mac of arp request not set to broadcast/invalid address

```

Interface : 0x3
[ethernet]
source mac: 08-...-00-00-00-00
dest mac : d4-...-00-0c-00-00
type : 0x806
[arp]
direction : out
type : request
source ip : 192.168.168.188
dest ip : 192.168.168.87
source mac: 08-...-00-00-00-00
dest mac : d4-...-00-0c-00-00

```

<http://www.chrismc.de>

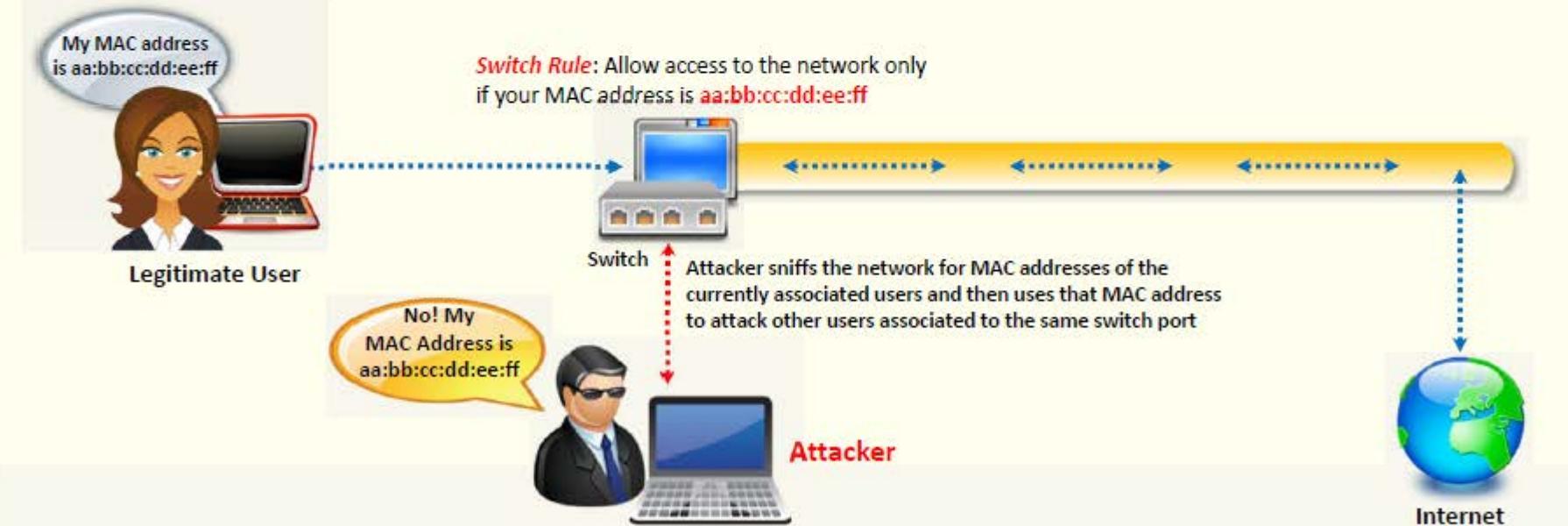
Module Flow



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

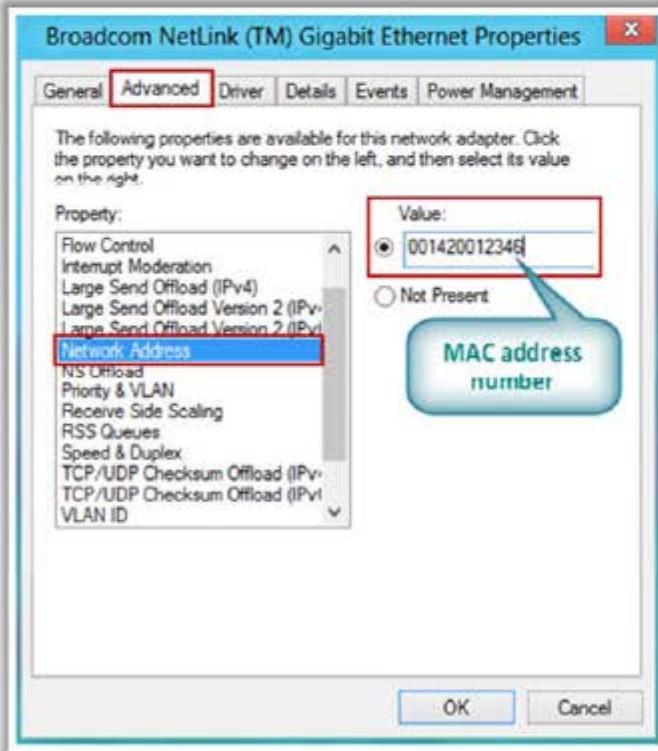
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

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)

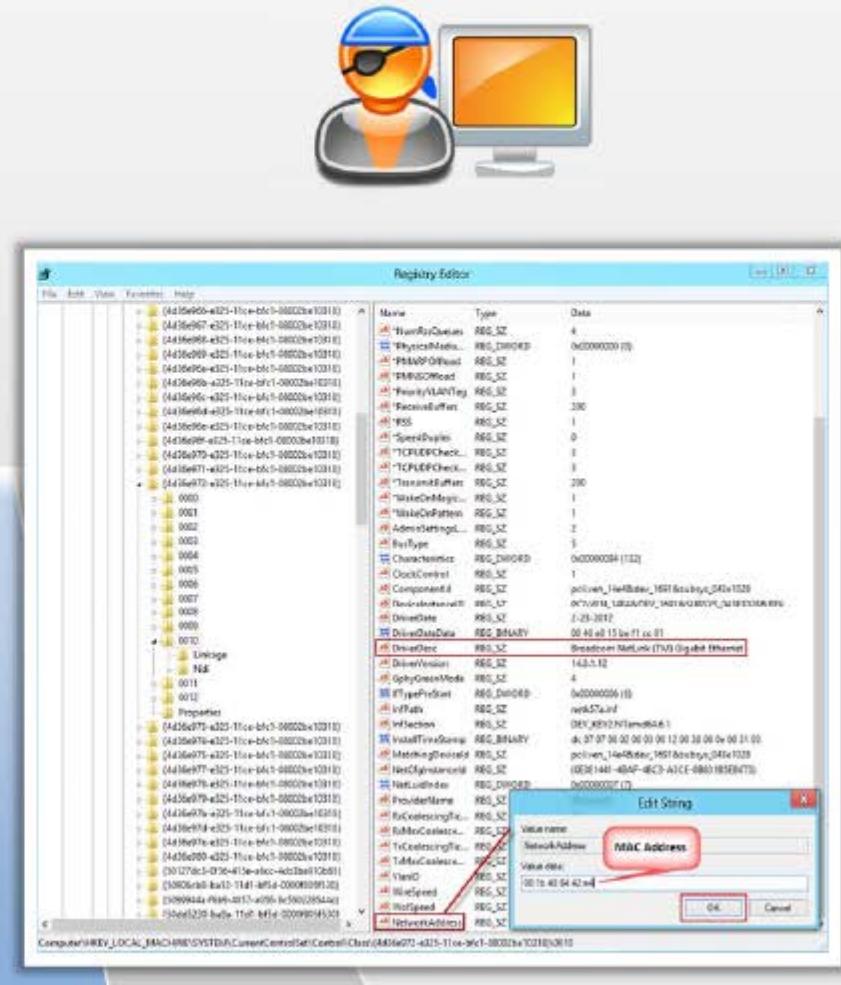


Method 2: Steps to change MAC address in Registry

- Go to **Start → Run**, type **regedit32** to start registry editor

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

- Go to
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControls et\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

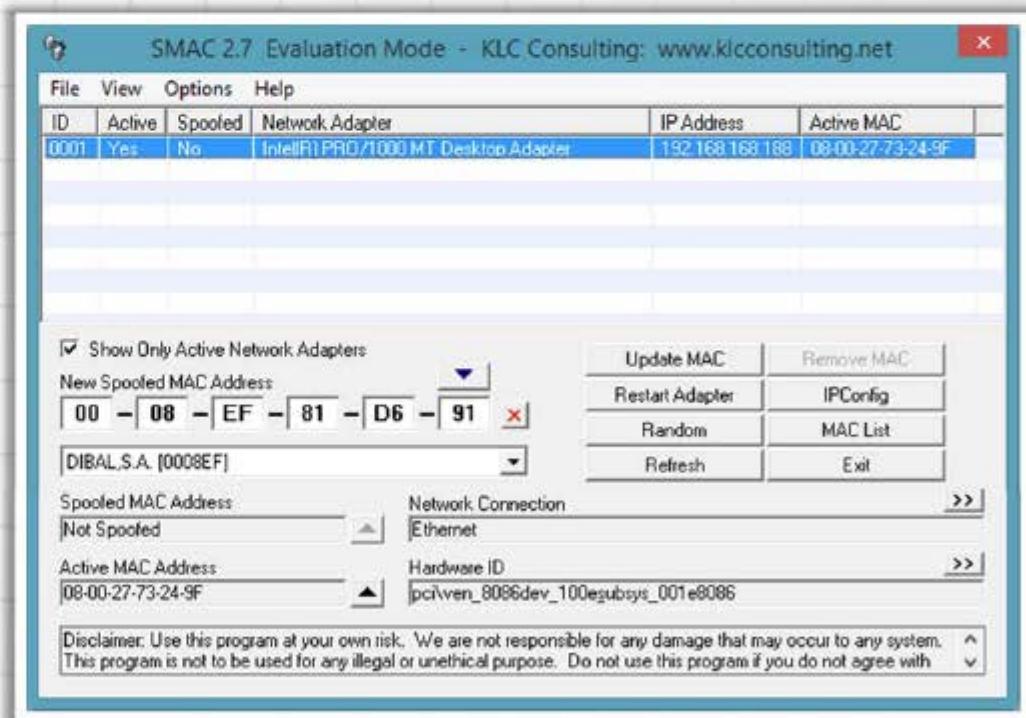


SMAC is a MAC Address Changer (Spoofing) tool that allows users to **change MAC address** for any network interface cards (NIC) on the Windows systems.



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.kleeeconsulting.net>

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

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	FastEthernet 3/18



10.10.10.1
MAC A

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

DHCP Snooping Enabled
Dynamic ARP Inspection Enabled
IP Source Guard Enabled

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

Traffic Sent with IP
10.10.10.5 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	FastEthernet 3/18



10.10.10.1
MAC A

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

DHCP Snooping Enabled
Dynamic ARP Inspection Enabled
IP Source Guard Enabled

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

Traffic Sent with IP
10.10.10.5 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

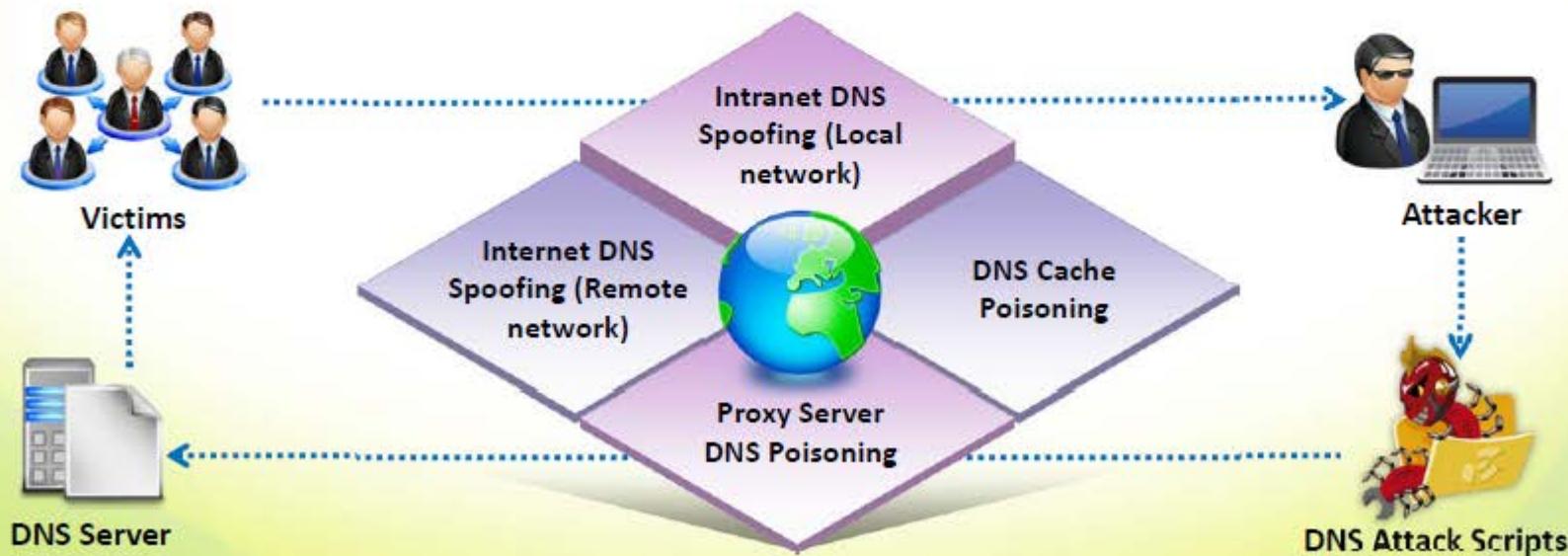


DNS Poisoning Techniques



- 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

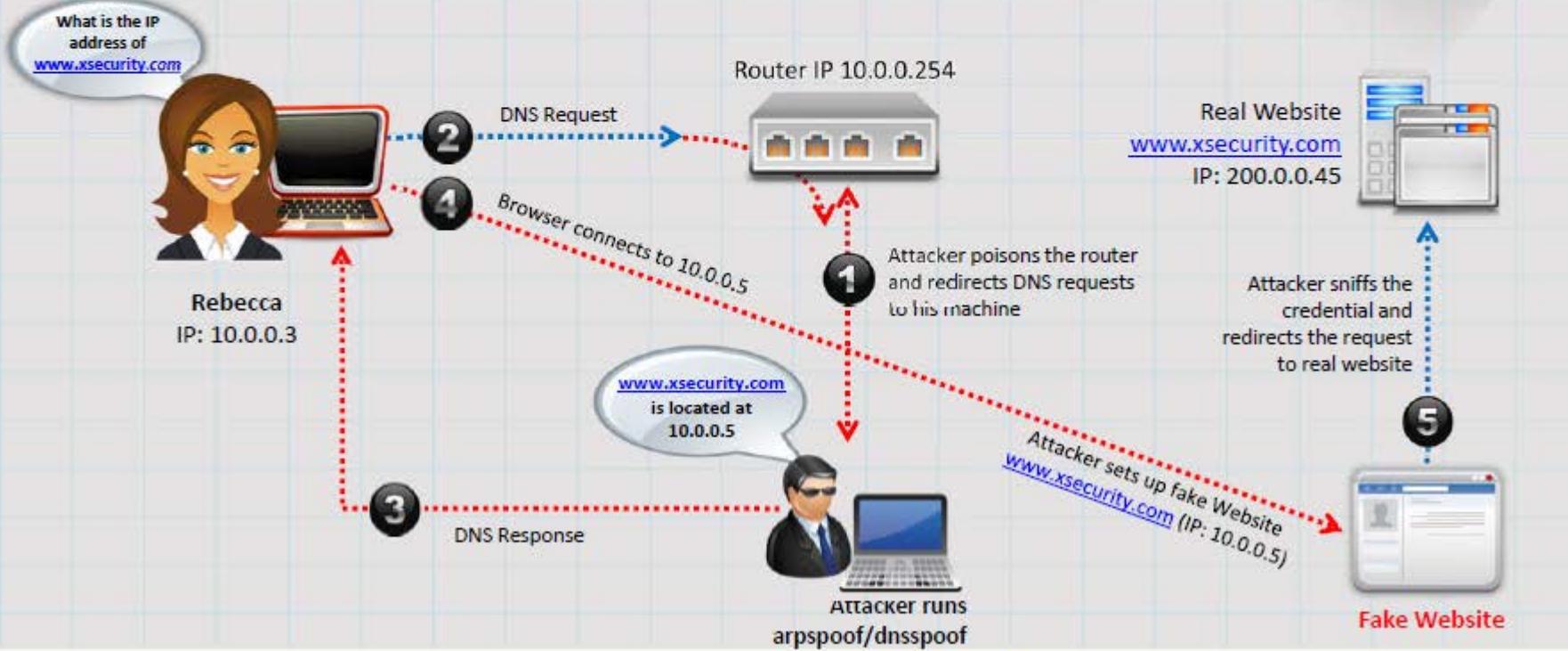
- It allows attacker to replace **IP address entries** for a target site on a given DNS server with IP address of the server he/she controls
- Attacker can create **fake DNS entries** for the server (containing malicious content) with same names as that of the target server



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

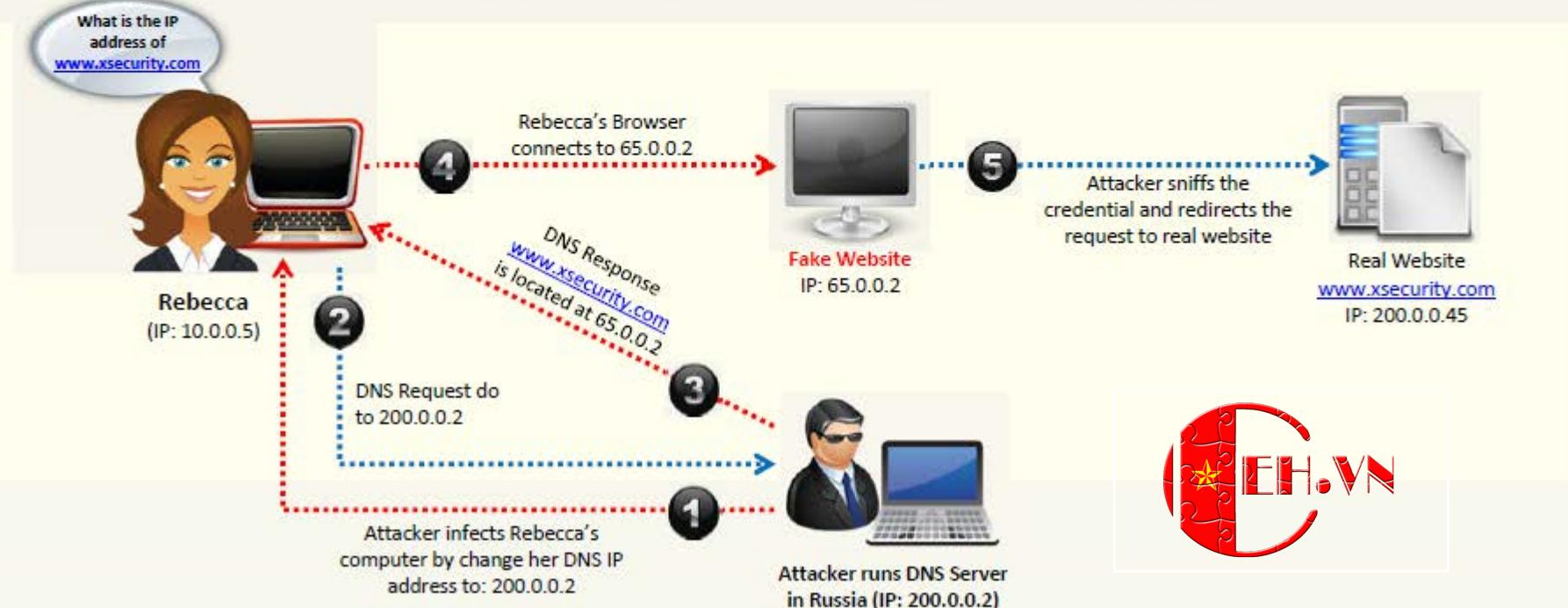


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Internet DNS Spoofing



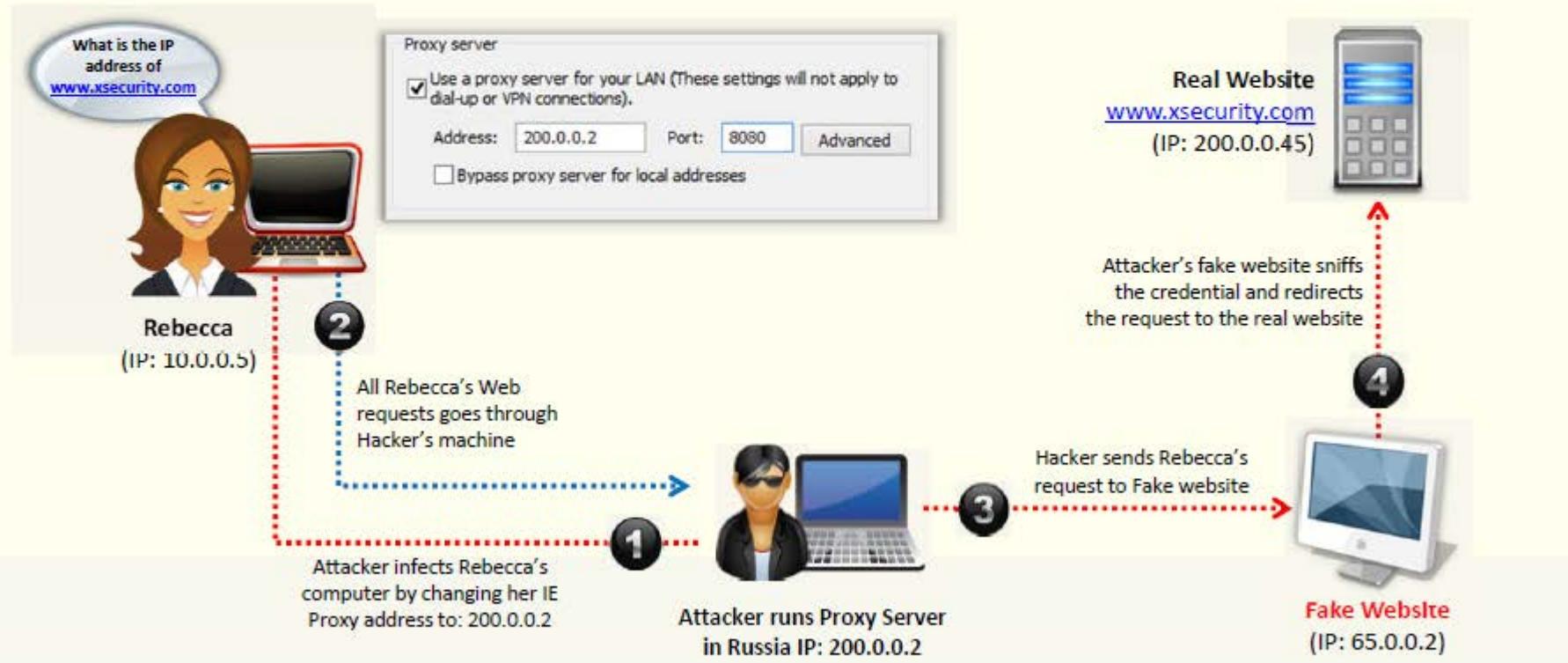
Internet DNS Spoofing, attacker **infects Rebecca's machine** with a Trojan and **changes her DNS IP address** to that of the attacker's



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

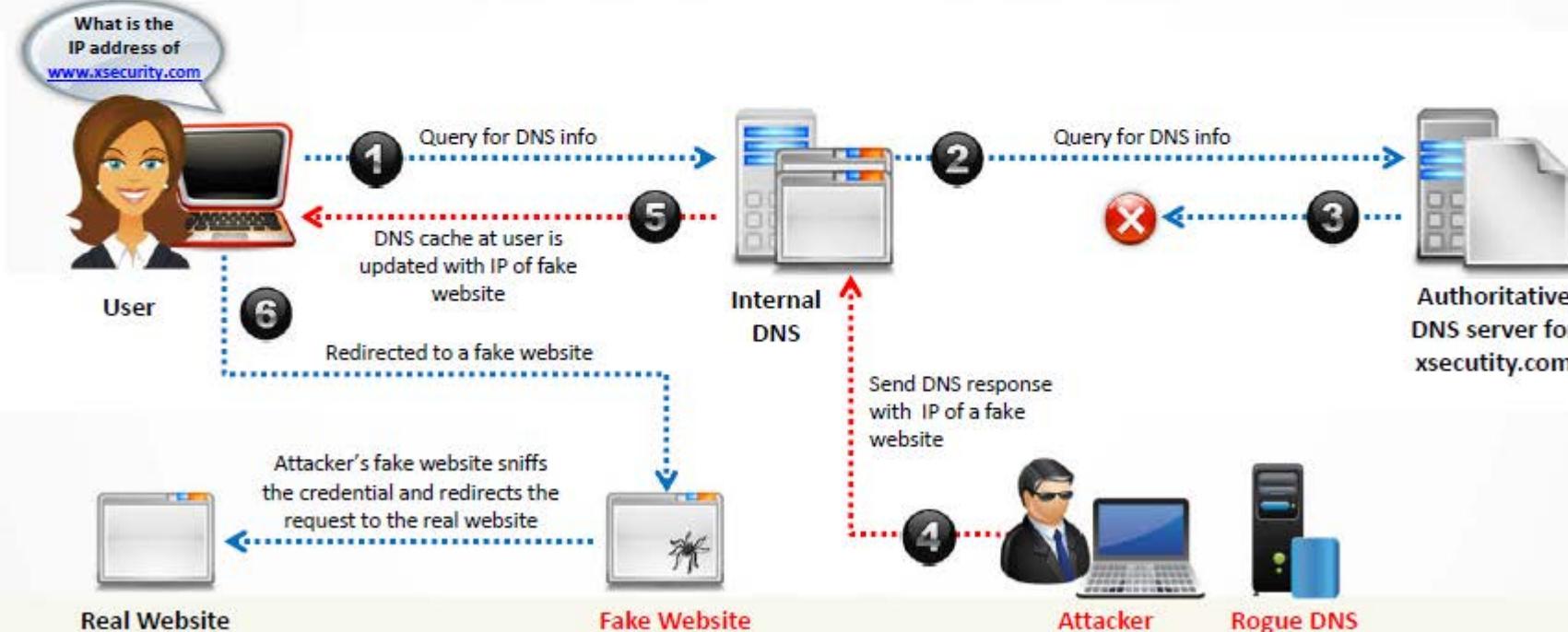


DNS Cache Poisoning



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

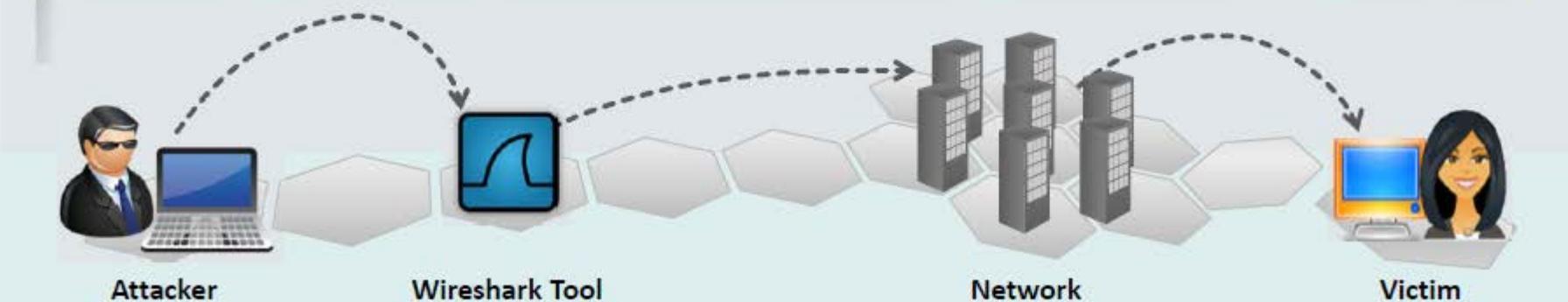
03

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)



Capturing from Ethernet [Wireshark 1.10.2 (SVN Rev. 51934 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.168.133	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
2	1.602768000	fe80::4855:5c3d:b13ff02::1:3		LLMNR	85	Standard query 0x4d7a A seot4
3	1.603425000	192.168.168.61	224.0.0.252	LLMNR	65	Standard query 0x4d7a A seot4
4	1.702724000	fe80::4855:5c3d:b13ff02::1:3		LLMNR	85	Standard query 0x4d7a A seot4
5	1.702728000	192.168.168.61	224.0.0.252	LLMNR	65	Standard query 0x4d7a A seot4
6	1.729089000	Dell_c3:b1:8b	Broadcast	ARP		
7	1.729119000	CadmusCo_73:24:9f	Dell_c3:b1:8b	ARP		
8	1.729869000	192.168.168.75	192.168.168.133	TCP		
9	1.730016000	CadmusCo_73:24:9f	Broadcast	ARP		
10	1.730676000	Dell_c3:b1:8b	CadmusCo_73:24:9f	ARP		
11	1.730693000	192.168.168.133	192.168.168.75	TCP		
12	1.731392000	192.168.168.75	192.168.168.133	TCP		
13	1.732178000	107.168.148.75	107.168.148.133	HTTP		

Frame 3: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface Ethernet II, Src: Elitegro_22:30:de (00:25:11:22:30:de), Dst: Broadcast (255.255.255.255)
 Internet Protocol Version 4, Src: 192.168.168.61 (192.168.168.61), Dst: 224.0.0.252 (224.0.0.252)
 User Datagram Protocol, Src Port: 49279 (49279), Dst Port: 111 (111)
 Link-local Multicast Name Resolution (query)

0000 01 00 5e 00 00 fc 00 25 11 22 30 de 08 00 45 00 ..^....
 0010 00 33 07 f3 00 00 01 11 67 e5 c0 a8 a8 3d e0 00 .3.....
 0020 00 fc c0 7f 14 eb 00 1f b1 d0 4d 7a 00 00 00 01 00
 0030 00 00 00 00 00 05 73 65 6f 74 34 00 00 01 00
 0040 01 ..

Ethernet: <live capture in progress> File: C:\... Packets: 2194 - Displayed: 2194 (100.0%)

http://www.wireshark.org

Wireshark: Filter Expression - Profile: Default

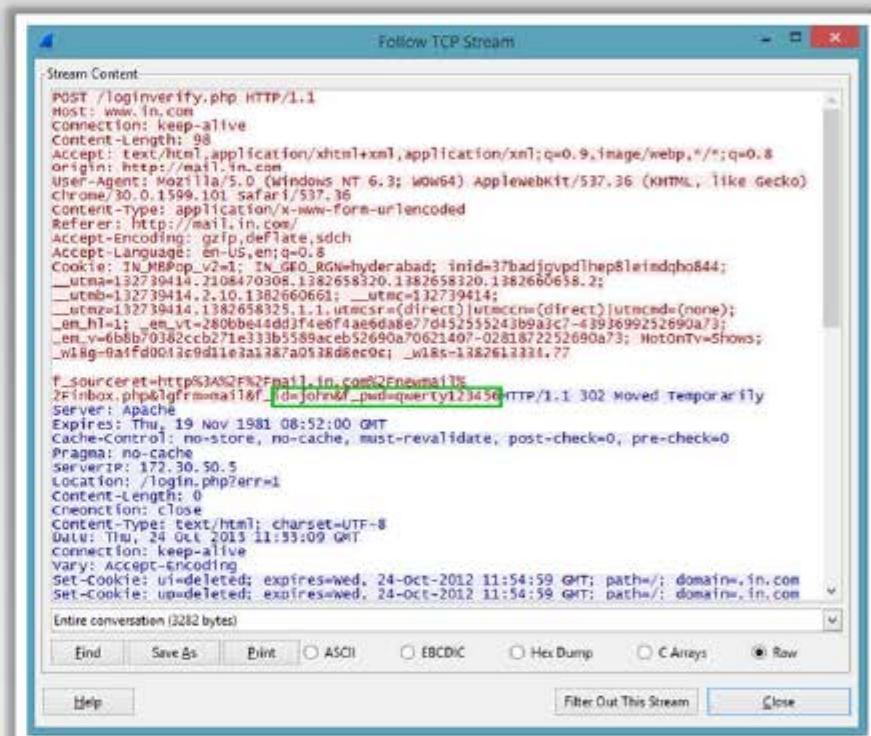
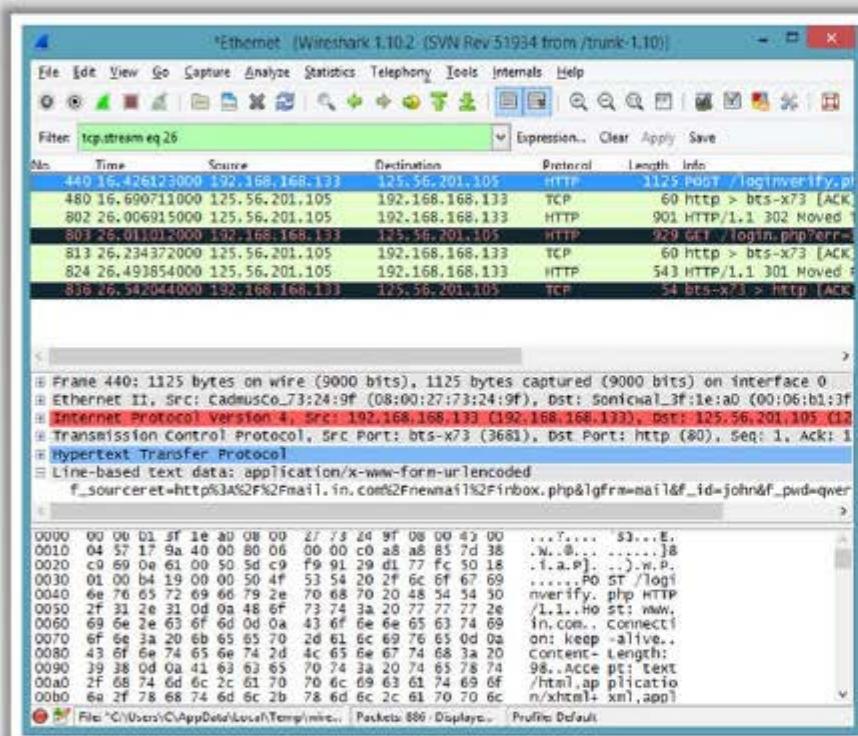
Field name: 104asdu - IEC 60870-5-104-Asdu
 Relation: is present
 Value (Protocol):

== Predefined values:
 !=
 >
 <
 >=

Range (offset:length):

OK Cancel

Follow TCP Stream in Wireshark



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 your interest



Sniffing Tool: SteelCentral Packet Analyzer



The image shows four windows of the SteelCentral Packet Analyzer software:

- Top Left Window:** MAC Overview. It displays a line graph of MAC traffic over time, a bar chart of Top Ten MAC Sources (with 'Recently Used' at 14.46%), and a network map showing various MAC addresses.
- Top Right Window:** IP Overview. It shows a network map with nodes like 224.0.2.22, 224.0.2.22, 192.0.2.56, 10.0.0.3, and 10.0.0.2. A summary box indicates End Point Bytes: 144.00, Conversation Bytes: 213.33, 104.00, 819.79, 184.00.
- Bottom Left Window:** Protocols Overview. It displays two bar charts: Network Protocols (HTTP at 12.11%, TCP at 11.88%) and Transport Protocols (TCP at 10.40%, UDP at 3.64%).
- Bottom Right Window:** Protocols Distribution. It shows a bar chart of protocols: HTTP (1.60%), SSL (1.56%), and others.

Text Overlay:

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

<http://www.riverbed.com>

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

```
tcpdump -i eth0
13:13:48.437836 10.20.21.03.router > RIP2-
ROUTERS.MCAST.NET.router: RIPv2
13:13:48.437836 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.filenet-tms >
btwdns01.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_{6A410B-BB83-4...
15:18:35.004005 IP [REDACTED].137 > 192.168.168.255.137: UDP, length
15:18:35.372362 IP6 [REDACTED]F2JB069755.546 > FF02::1:2.547: dhcp6 soli
15:18:35.469372 IP6 admin.50347 > FF02::1:3.5355: UDP, length 46
15:18:35.669718 IP admin.50347 > 224.0.0.252.5355: UDP, length 46
15:18:35.854857 IP6 [REDACTED].61220 > FF02::1:3.5355: UDP, length 23
15:18:35.855677 IP [REDACTED].63168 > 224.0.0.252.5355: UDP, length 23
15:18:35.954878 IP6 [REDACTED].61220 > FF02::1:3.5355: UDP, length 23
15:18:35.955385 IP [REDACTED].63168 > 224.0.0.252.5355: UDP, length 23
15:18:36.082704 IP6 admin.50347 > FF02::1:3.5355: UDP, length 46
15:18:36.083064 IP admin.50347 > 224.0.0.252.5355: UDP, length 46
15:18:36.154829 IP [REDACTED].137 > 192.168.168.255.137: UDP, length
15:18:36.459859 IP [REDACTED]-PC.137 > 192.168.168.255.137: UDP, length
15:18:36.494136 IP admin.137 > [REDACTED].137: UDP, length 50
15:18:36.494641 IP6 admin.64799 > FF02::1:3.5355: UDP, length 45
15:18:36.494898 IP admin.64799 > 224.0.0.252.5355: UDP, length 45
15:18:36.495848 IP [REDACTED].137 > admin.137: UDP, length 175
15:18:36.496685 IP [REDACTED].5355 > admin.64799: UDP, length 94
15:18:36.496743 IP admin > [REDACTED]: ICMP admin udp port 64799 unrec
th 130
15:18:36.497512 IP6 admin.49395 > FF02::1:3.5355: UDP, length 90
15:18:36.497750 IP admin.49395 > 224.0.0.252.5355: UDP, length 90
15:18:36.908465 IP [REDACTED].137 > 192.168.168.255.137: UDP, length
15:18:36.908626 IP6 admin.49395 > FF02::1:3.5355: UDP, length 90
15:18:36.908853 IP admin.49395 > 224.0.0.252.5355: UDP, length 90
15:18:37.218184 IP [REDACTED]-PC.137 > 192.168.168.255.137: UDP, length
15:18:37.252186 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**



OmniPeek

File Edit View Capture Send Monitor Tools Window Help

Packets received: 2,000 Buffer usage: 2% Filter state: Accept all packets Start Capture

(Enter a filter expression here (use ? for help))

Dashboard	Packet	Source	Destination	Flags	Size	Relative Time	Protocol	Summary
Network	1	97.65.218.170	10.0.0.2		1510	0.000000000	TCP	Src=47611,Dst=4192,AP...,2=3
Voice & Video	2	10.0.0.2	97.65.218.170		64	0.009803000	TCP	Src=4192,Dst=47611,A...,2=3
Apdex	3	10.0.0.2	49.90.118.83		842	0.104861000	HTTP	C PORT=4192 GET /Sichtbarkeit
CoSpans	4	69.90.118.83	10.0.0.2		64	0.138966000	HTTP	Src=10,Dst=4192,A...,2=3
Capture	5	97.65.218.170	10.0.0.2		1510	0.142944000	TCP	Src=47611,Dst=4192,A...,2=3
# Packets	6	97.65.218.170	10.0.0.2		1510	0.144892000	TCP	Src=47611,Dst=4192,A...,2=3
Log	7	10.0.0.2	97.65.218.170		64	0.148944000	TCP	Src=4192,Dst=47611,V...,2=3
Filters	8	97.65.218.170	10.0.0.2		1510	0.154866000	TCP	Src=47611,Dst=4192,A...,2=3
Expert	9	97.65.218.170	10.0.0.2		1510	0.160971000	TCP	Src=4192,Dst=47611,A...,2=3
Hierarchy	10	10.0.0.2	97.65.218.170		64	0.160971000	TCP	Src=4192,Dst=47611,A...,2=3
Map	11	97.65.218.170	10.0.0.2		1510	0.170426000	TCP	Src=47611,Dst=4192,A...,2=3
Application	12	97.65.218.170	10.0.0.2		1510	0.176989000	TCP	Src=47611,Dst=4192,A...,2=3
Web	13	97.65.218.170	10.0.0.2		64	0.177032000	TCP	Src=4192,Dst=47611,A...,2=3
Servers	14	97.65.218.170	10.0.0.2		1510	0.190413000	TCP	Src=47611,Dst=4192,A...,2=3
Clients	15	97.65.218.170	10.0.0.2		1510	0.205014000	TCP	Src=4192,Dst=47611,A...,2=3
Pages	16	10.0.0.2	97.65.218.170		64	0.205035000	TCP	Src=4192,Dst=47611,A...,2=3
Requests	17	97.65.218.170	10.0.0.2		1510	0.211077000	TCP	Src=47611,Dst=4192,A...,2=3
Voice & Video	18	97.65.218.170	10.0.0.2		1510	0.217798000	TCP	Src=47611,Dst=4192,A...,2=3
Call	19	10.0.0.2	97.65.218.170		64	0.217769000	TCP	Src=4192,Dst=47611,A...,2=3
Mails	20	97.65.218.170	10.0.0.2		1510	0.225853000	TCP	Src=47611,Dst=4192,A...,2=3
Visuals	21	97.65.218.170	10.0.0.2		1510	0.231254000	TCP	Src=47611,Dst=4192,A...,2=3
Peer Map	22	10.0.0.2	97.65.218.170		64	0.231274000	TCP	Src=4192,Dst=47611,A...,2=3
Graphs	23	97.65.218.170	10.0.0.2		1510	0.237170000	TCP	Src=47611,Dst=4192,A...,2=3
Statistics	24	97.65.218.170	10.0.0.2		1510	0.243422000	TCP	Src=47611,Dst=4192,A...,2=3
Nodes	25	10.0.0.2	97.65.218.170		64	0.243440000	TCP	Src=4192,DST=47611,A...,2=3
Protocols	26	97.65.218.170	10.0.0.2		1510	0.249407000	TCP	Src=47611,Dst=4192,A...,2=3
Summary	27	97.65.218.170	10.0.0.2		1510	0.255429000	TCP	Src=47611,Dst=4192,A...,2=3
	28	10.0.0.2	97.65.218.170		64	0.258455000	TCP	Src=4192,DST=47611,A...,2=3
	29	97.65.218.170	10.0.0.2		1510	0.281348000	TCP	Src=47611,Dst=4192,A...,2=3

Idle vEthernet (Health) PCIe GBE Family Controller - Virtual Switch Duration: 0:00:12
<http://www.wildpackets.com>

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**

The screenshot displays the NetworkMiner interface, which is a network traffic analysis tool. The main window shows a list of captured network packets, each with detailed information such as source and destination addresses, type, and summary. A specific packet is selected, and its detailed structure is shown in the bottom pane, including fields like Network, Frame number, Type, and Destination Address. Below the main window, there is a log viewer showing system events and a status bar at the bottom.

Packet	Source	Destination	Type	Summary	Off Time	Day Time	Relative Time
1	Export Information	Export Information	ExpertInfo	ExpertInfo: Packet Capture Started	0.000 000	12:03m 20.426 99%	00:00:000
2	00:00:00:14:00	00:00:00:42:C1	GlobalLS	NetTOD(0) Conn->17.0.1 client — IP(125.102.0.2) 000000000000	0.000 000	12:03m 21.427 00%	01:00:000
3	00:00:00:14:00	00:00:00:42:C1	GlobalLS	NetTOD(0) Conn->17.0.1 client — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 21.427 00%	01:00:000
4	00:00:00:14:00	00:00:00:42:C1	GlobalLS	NetTOD(0) Conn->17.0.1 client — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 21.427 00%	01:00:000
5	Export Information	Export Information	ExpertInfo	ExpertInfo: PktInfo=2, BytesInfo=28, UNR=0.0%	0.000 000	12:03m 21.859 99%	01:56:000
6	00:00:00:14:00	00:00:00:42:C1	GlobalLS	DIAGNOSTIC RESULTS — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 22.427 00%	02:00:000
7	00:00:00:14:00	00:00:00:42:C1	GlobalLS	DIAGNOSTIC RESPONSE — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 22.501 51%	02:08:004
8	00:00:00:14:00	00:00:00:42:C1	GlobalLS	DIAGNOSTIC RESPONSE — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 22.501 51%	02:08:004
9	00:00:00:14:00	00:00:00:42:C1	GlobalLS	DIAGNOSTIC RESPONSE — IP(125.102.0.2) 0000000042C0	0.000 000	12:03m 22.501 51%	02:08:004



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

Sniff - O - Matic 1.07 Trial Version

Packet	Source	Destination	Size	Proto.	Time	Port src	Port dest
1	192.168.168.61	224.0.0.252	65	UDP	10/24/13 11:06:21	64138	5365
2	192.168.168.37	255.255.255.255	153	UDP	10/24/13 11:06:21	17500	17500
3	192.168.168.37	192.168.168.255	153	UDP	10/24/13 11:06:21	17500	17500
4	192.168.168.61	224.0.0.252	65	UDP	10/24/13 11:06:21	64138	5365
5	192.168.168.61	192.168.168.255	92	UDP	10/24/13 11:06:22	137	137
6	192.168.168.133	239.255.255.250	175	UDP	10/24/13 11:06:22	63263	1900
7	192.168.168.133	239.255.255.250	175	UDP	10/24/13 11:06:22	63263	1900
8	192.168.168.61	192.168.168.255	92	UDP	10/24/13 11:06:22	137	137
9	192.168.168.38	255.255.255.255	138	UDP	10/24/13 11:06:23	7768	7768
10	192.168.168.61	192.168.168.255	92	UDP	10/24/13 11:06:23	137	137
11	192.168.168.11	224.0.0.252	65	UDP	10/24/13 11:06:23	55552	5365
12	192.168.168.11	224.0.0.252	65	UDP	10/24/13 11:06:23	55552	5365
13	192.168.168.11	192.168.168.255	92	UDP	10/24/13 11:06:24	137	137

IP Header
 Version = 4
 Header Length = 5 (20 bytes)
 Type Of Service = 0x00
 Total Length = 161
 Identification = 0x11DF
 Flags = 0x00
 Fragment offset = 0x0000
 Time To Live = 1
 Protocol = 17 (UDP)
 Header Checksum = 0x4E56
 Source IP = 192.168.168.133
 Dest. IP = 239.255.255.250
 UDP Header
 Source Port = 63263
 Destination Port = 1900
 Length = 141
 Checksum = 0x59C7
 Data
 Data length = 133

0X0000 45 00 00 A1 11 DF 00 00 01 11 4E 45 C0 A8 A8 85 E.....
 0X0010 EF FF FF FA F7 1F 07 6C 00 BD 59 CT 4D 2D 53 451.
 0X0020 41 32 43 48 20 2A 20 48 54 54 50 2F 31 2E 31 0D ARCH * HIT
 0X0030 0A 48 6F 73 74 3A 32 33 39 2E 32 35 36 2E 32 38 .Host:238.
 0X0040 35 2E 32 35 30 3A 31 39 30 30 0D 0A 53 54 3A 75 5.250:1900
 0X0050 72 0E 3A 73 63 65 69 60 61 73 2D 7D 70 6B 70 2D
 0X0060 6F 72 67 3A 64 65 76 69 63 65 3A 49 6E 74 65 72 org:device
 0X0070 6E 65 74 47 61 74 65 77 61 79 44 65 76 69 63 65 netGateway
 0X0080 3A 31 0D 0A 4D 61 6E 3A 22 73 73 64 70 3A 64 69 :1..Man!"
 0X0090 73 63 6F 76 65 72 22 0D DA 4D 58 3A 53 0D 0A 0D recover".H
 0X00A0 0A

<http://www.kwakkelflap.com>

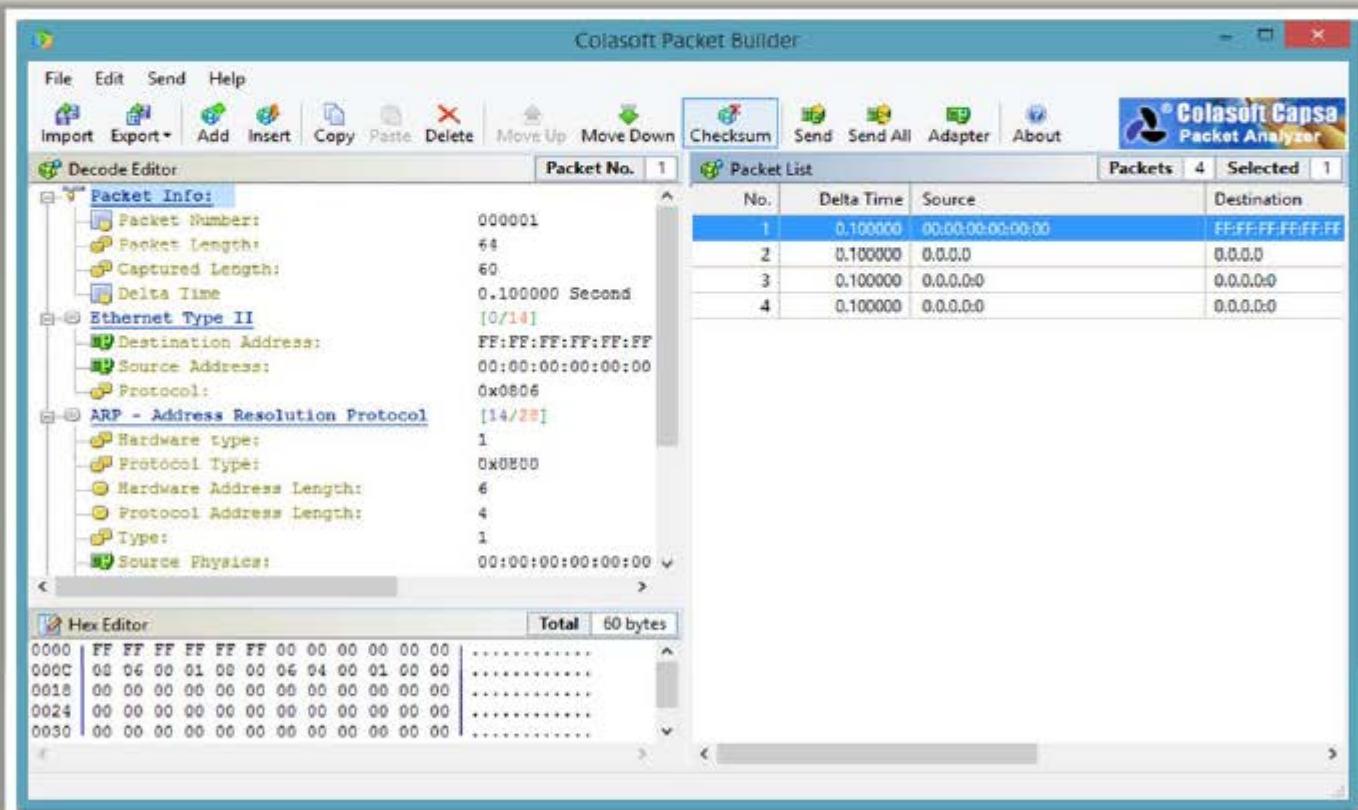
<http://www.kwakkelflap.com>

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

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



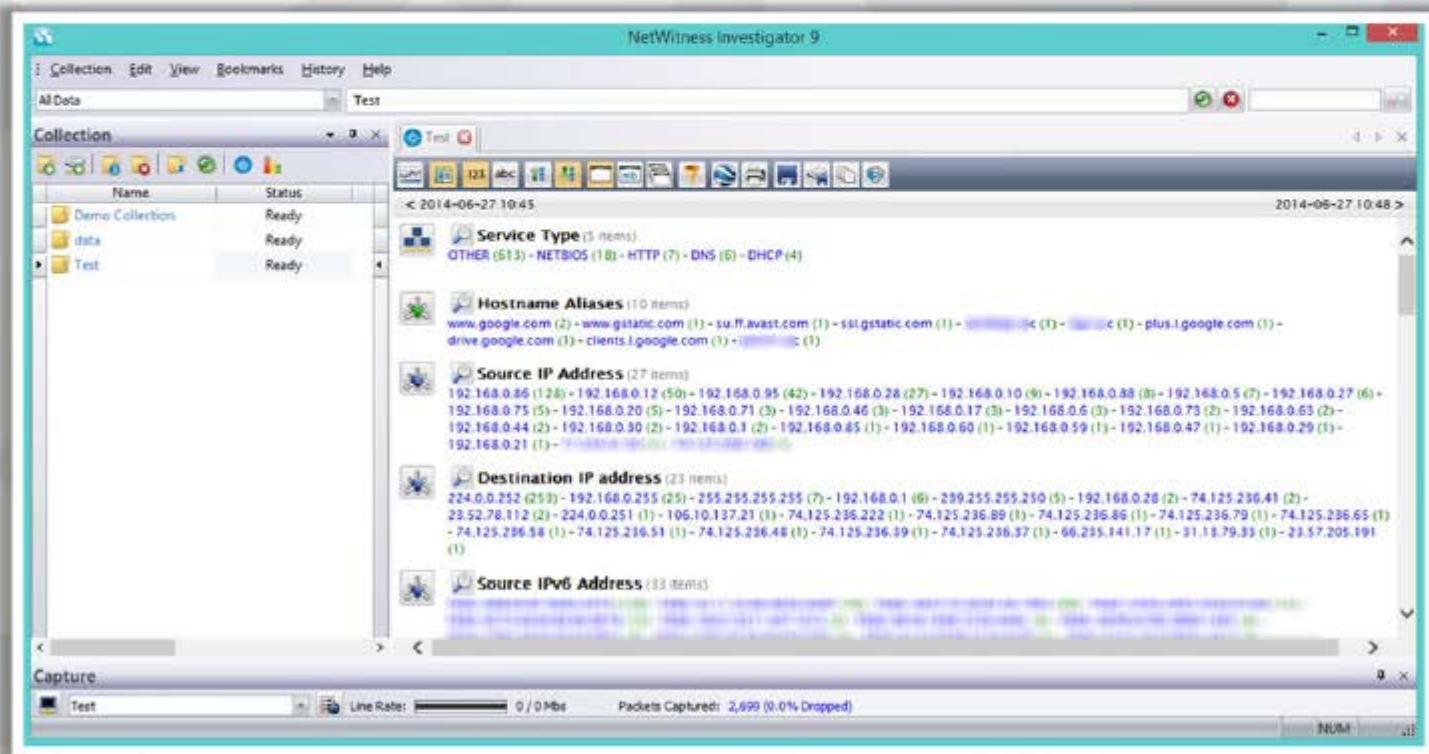
http://www.colasoft.com

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

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>

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

Additional Sniffing Tools

**Ace Password Sniffer**<http://www.effetech.com>**IPgrab**<http://ipgrab.sourceforge.net>**Big-Mother**<http://www.tupsoft.com>**EtherDetect Packet Sniffer**<http://www.etherdetect.com>**dsniff**<http://monkey.org>**EffeTech HTTP Sniffer**<http://www.effetech.com>**ntopng**<http://www.ntop.org>**Ettercap**<http://ettercap.sourceforge.net>**SmartSniff**<http://www.nirsoft.net>**EtherApe**<http://etherape.sourceforge.net>

Additional Sniffing Tools

(Cont'd)



Network Probe

<http://www.objectplanet.com>



WebSiteSniffer

<http://www.nirsoft.net>



ICQ Sniffer

<http://www.etherboss.com>



MaaTec Network Analyzer

<http://www.maatec.com>



Alchemy Network Monitor

<http://www.mishelpers.com>



CommView

<http://www.tamos.com>



NetResident

<http://www.tamos.com>



Kismet

<http://www.kismetwireless.net>



AIM Sniffer

<http://www.effetech.com>



Netstumbler

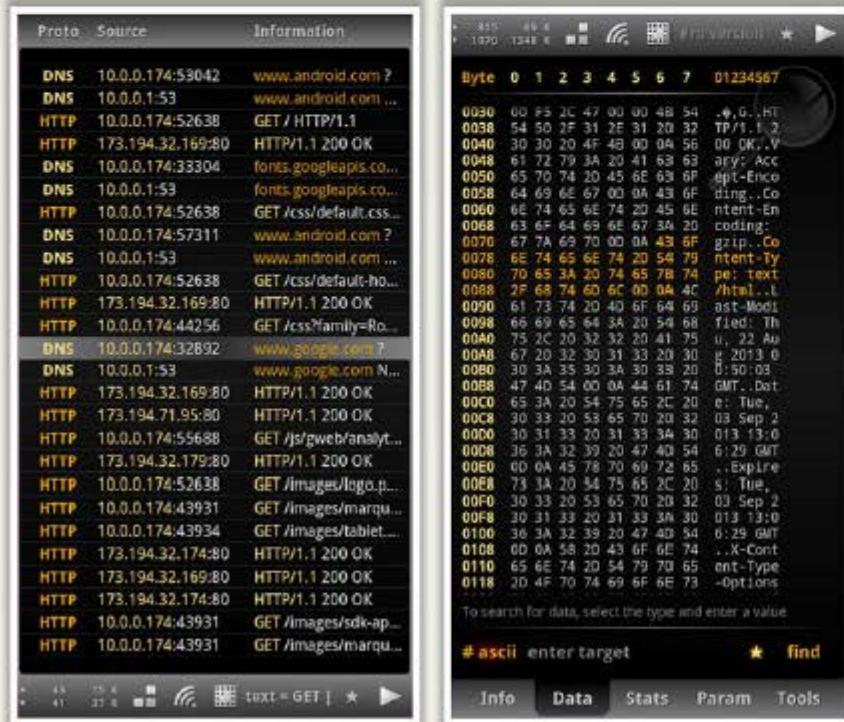
<http://www.netstumbler.com>

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



Wi.cap. Network Sniffer Pro

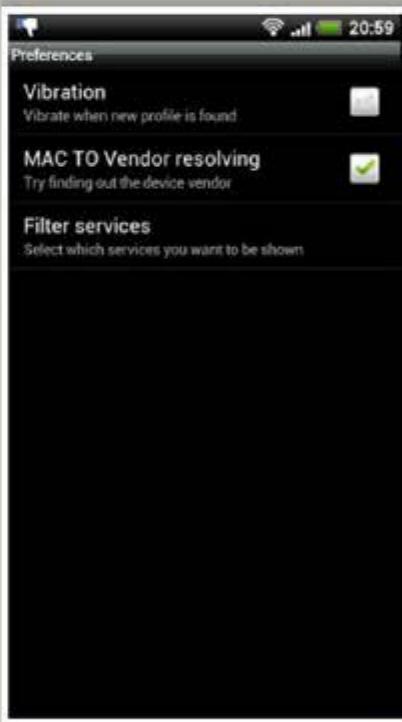
Mobile network packet sniffer for **ROOT ARM droids**



<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)



- Use **HTTPS** instead of HTTP to protect user names and passwords
- Use **switch instead of hub** as switch delivers data only to the intended recipient
- Use **SFTP**, instead of FTP for secure transfer of files
- Use **PGP and S/MIME, VPN, IPSec, SSL/TLS, Secure Shell (SSH)** and One-time passwords (OTP)
- Always encrypt the wireless traffic with a **strong encryption protocol** such as WPA and WPA2
- Retrieve MAC** directly from NIC instead of OS; this prevents MAC address spoofing
- Use **tools** to determine if any NICs are running in the promiscuous mode

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

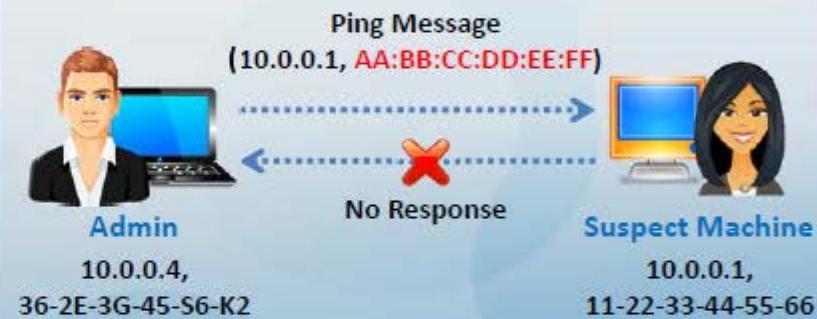
Sniffer Detection Technique: Ping Method



Promiscuous Mode

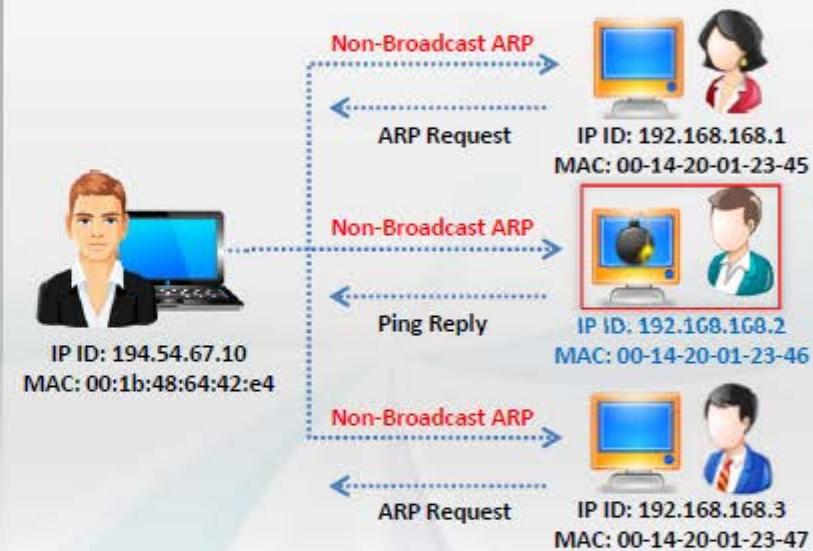
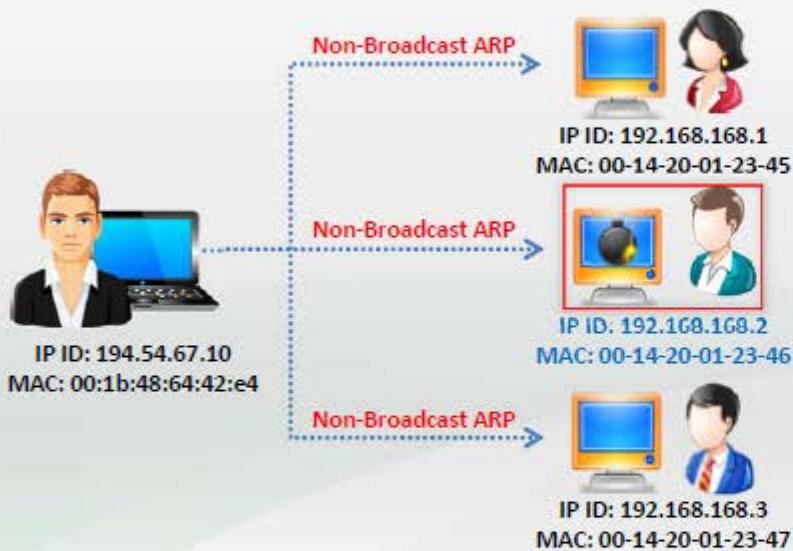


Non-Promiscuous Mode



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



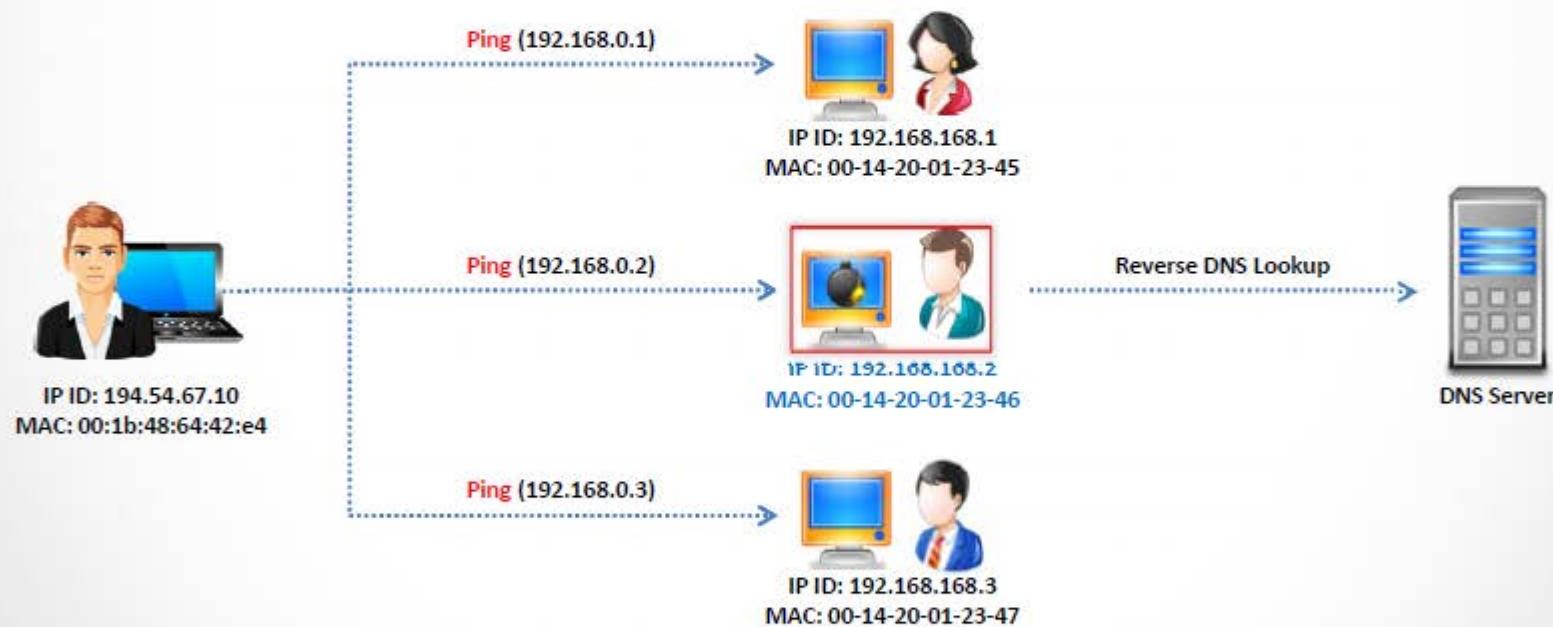
Only a machine in promiscuous mode
(machine C) **caches the ARP information**
(IP and MAC address mapping)

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



The screenshot shows the Promqry UI application window. On the left, under 'Systems To Query', there is a table with one row selected, showing 'Start IP address' as 192.168.168.133. A callout box contains the text: 'PromqryUI is a security tool from Microsoft that can be used to detect network interfaces that are running in promiscuous mode'. On the right, under 'Query Results', the application displays the status of four network interfaces:

- Intel(R) PRO/1000 MT Desktop Adapter: Active: True, InstanceName: Intel(R) PRO/1000 MT Desktop Adapter, NEGATIVE: Promiscuous mode currently NOT enabled.
- WAN Miniport (IP): Active: True, InstanceName: WAN Miniport (IP), NEGATIVE: Promiscuous mode currently NOT enabled.
- WAN Miniport (IPv6): Active: True, InstanceName: WAN Miniport (IPv6), NEGATIVE: Promiscuous mode currently NOT enabled.
- WAN Miniport (Network Monitor): Active: True, InstanceName: WAN Miniport (Network Monitor), NEGATIVE: Promiscuous mode currently NOT enabled.

At the bottom of the window are 'Add', 'Delete', and 'Start Query' buttons.

<http://www.microsoft.com>

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

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:~# 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  gmrupdate-serv
1433/tcp  open  ms-sql-s
1801/tcp  open  msmq
2103/tcp  open  zephyr-clt
2105/tcp  open  oklogin
2107/tcp  open  msmq-mgmt
2179/tcp  open  vncdcp
2383/tcp  open  ms-clap4
3309/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

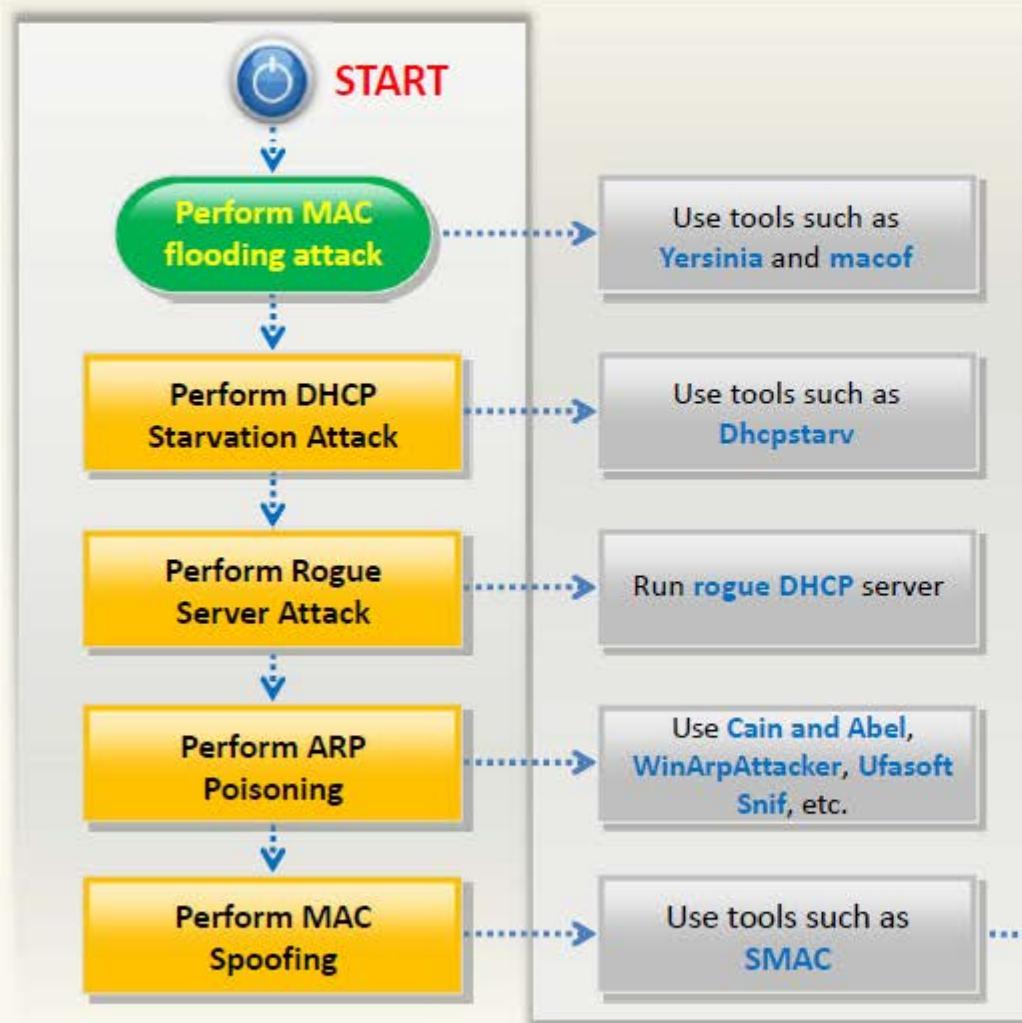


Discover the presence of unauthorized networking devices



Sniffing Pen Testing

(Cont'd)

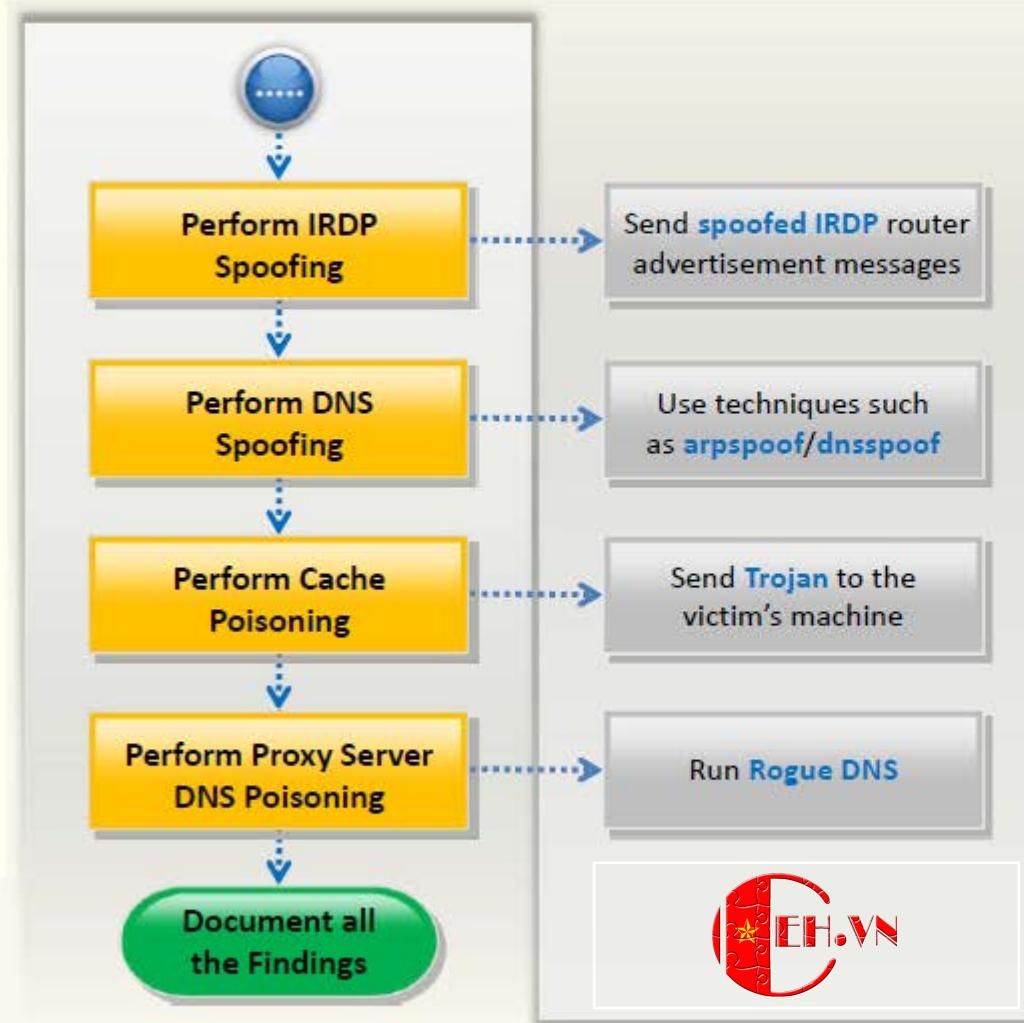


- Perform MAC flooding attack using tools such as **Yersinia** and **macof**
- Perform DHCP starvation attack using tools such as **Dhcpstarp** and **Yersinia**
- Perform rogue server attack by running **rogue DHCP server** in the network and responding to DHCP requests with **bogus IP addresses**
- 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