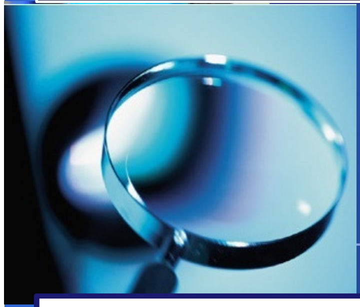
Intrusion Detection Systems: A Formal Algorithmic approach



Santosh Biswas
Associate Professor
Dept. of CSE, IIT Guwahati

What is Intrusion Detection System ?

- Intrusion
  - A set of actions aimed to compromise the security goals, namely
    - Integrity, confidentiality, or availability, of a computing and networking resource
- Intrusion detection
  - The process of identifying and responding to intrusion activities

IDS Taxonomy

❖Location of Deployment

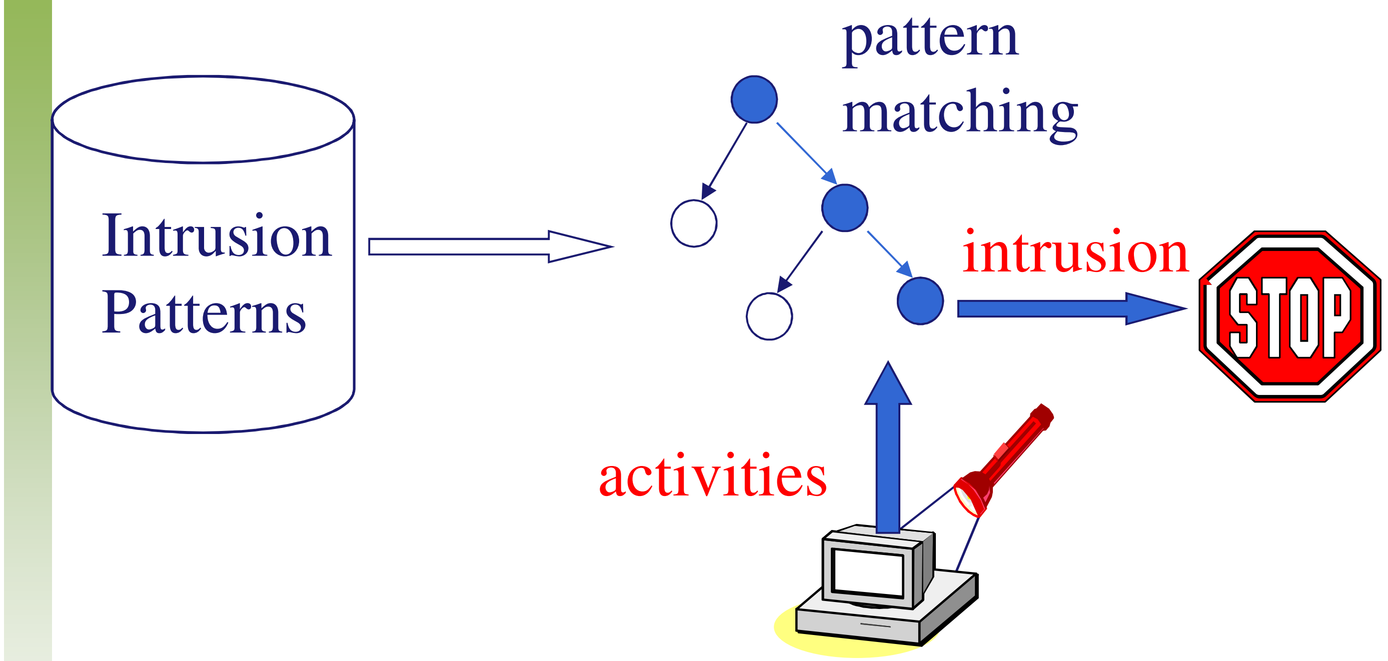
- Host based
  - Monitor Computer Processes
  - File Integrity Checkers (system files, checksum e.g. hash value)
  - Log File Analysis (attack s are encoded in terms of regular exp.)
  - Statistical Approach (session duration, CPU uses, no. of files open)
  - System Call Monitoring (any deviation is compared with normal seq.)
- Network based
  - Monitor Network Traffic
  - Packet Signatures
  - Anomalous Activity

IDS Taxonomy

❖Detection Methodology

- Signature based
  - Detects known attacks whose syntax and behavior is known
  - Can not detects new or novel attacks
  - Generate large number of False Positive Alarms

Signature based IDS



Example: ***if*** (src_ip == dst_ip) ***then*** "land attack"

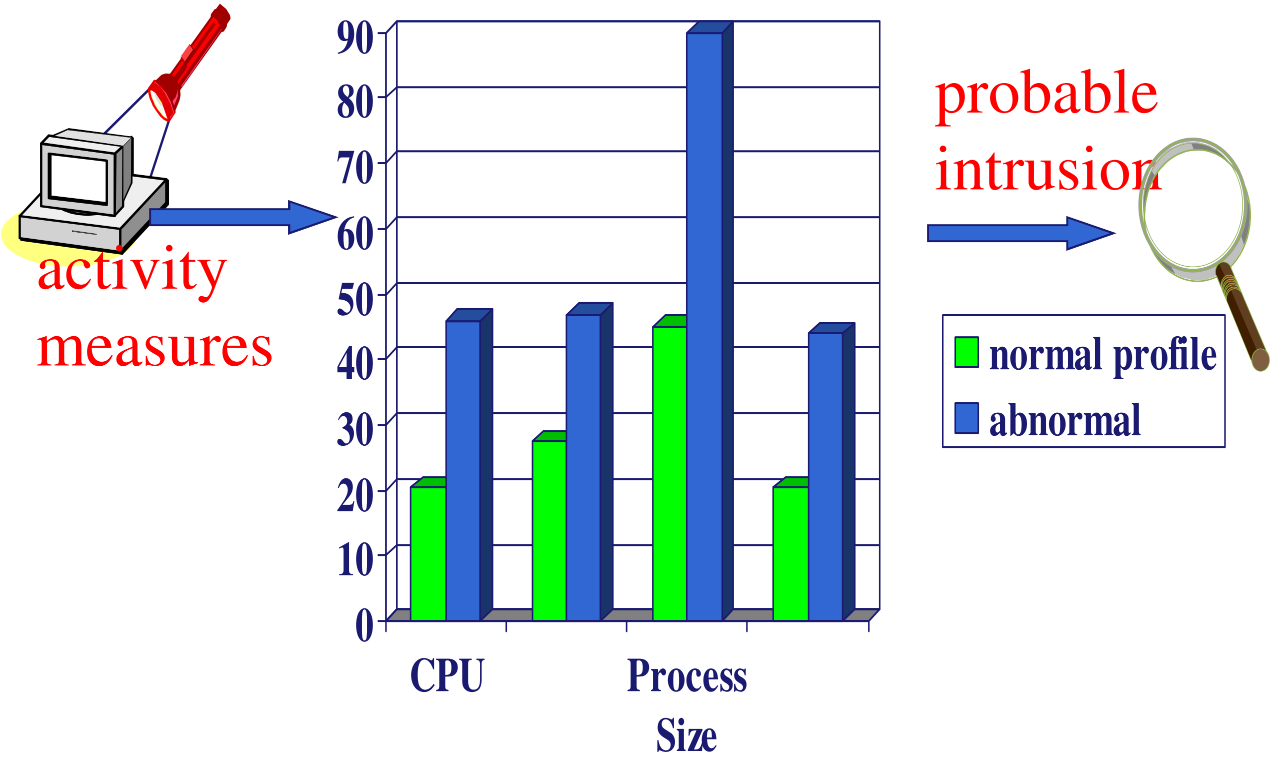
alert ip any any – > any any (msg : "BAD TRAFFIC sameSRC/DST"; sameip; reference : cve,CVE–1999–0016; url,www.cert.org/advisories/CA–1997–28.html; classtype : bad – unknown; sid : 527; rev : 3;)

Anomaly based IDS

❖Detection Methodology

- Anomaly based
  - Can detects both known and unknown attacks
  - Create normal (and/or attack) profile from training data set
  - Require pure training dataset for profile generation
  - Network packets are classified as Normal and Anomalous based on the profile
  - Detects patterns that do not confirm expected or normal behavior
  - Generate large number of False Positive Alarms

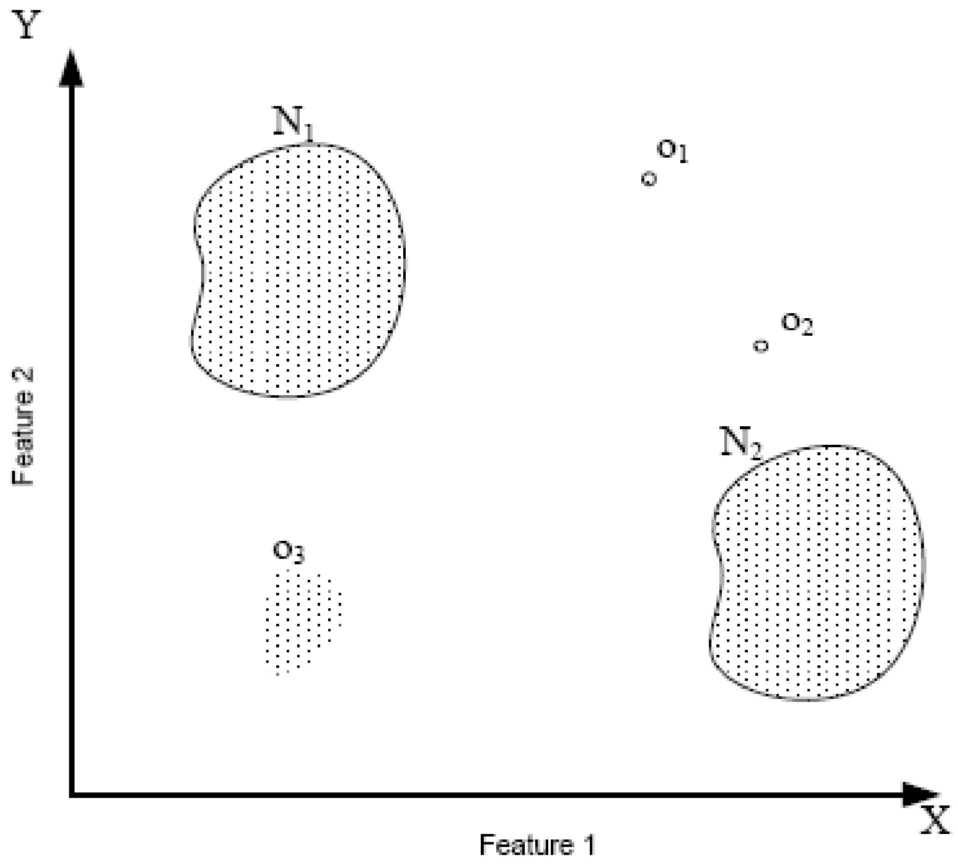
Anomaly based IDS



activity measures

probable intrusion

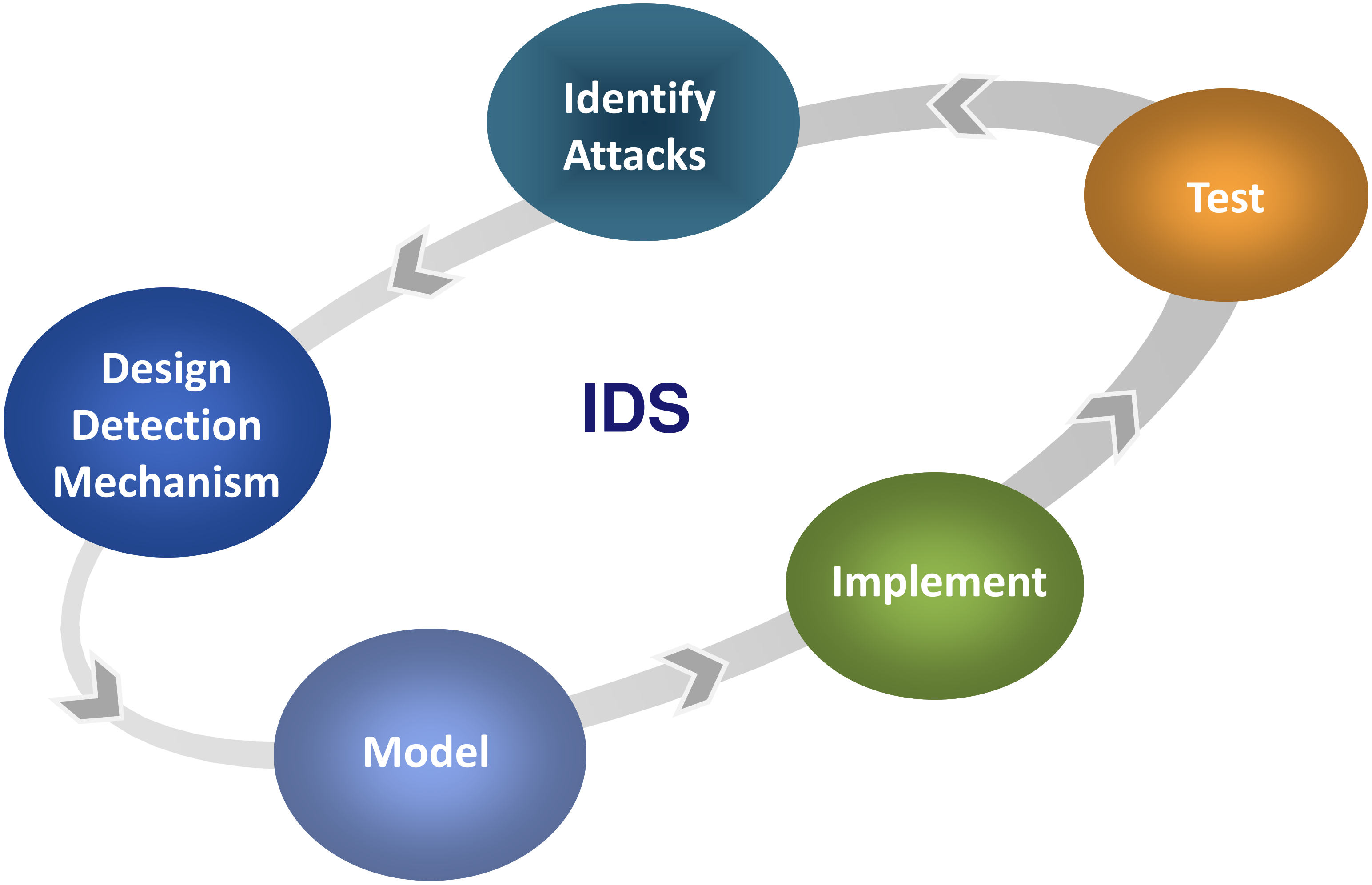
Anomaly based IDS



Event based IDS

- ❖Detection Methodology
  - Event based
    - Detects known attacks for which a signature can not be generated
    - These attacks do not change the syntax and sequence of network traffic under normal and compromised situation
    - Detection is through monitoring the difference in sequence of events (i.e. network packets) under normal and compromised situations

Objective



IDS

Identify Attacks

Design Detection Mechanism

Model

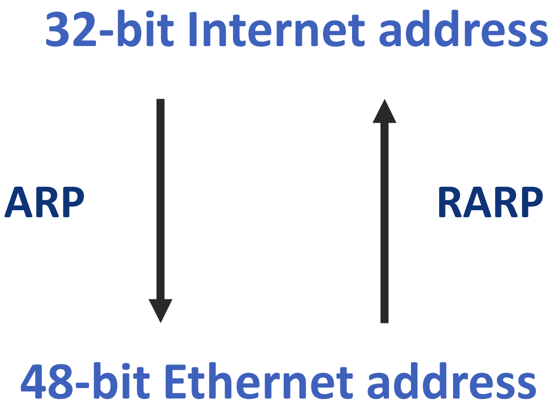
Implement

Test

What is ARP?

- Address Resolution Protocol maps IP address to MAC address

Purpose of ARP



32-bit Internet address

ARP

RARP

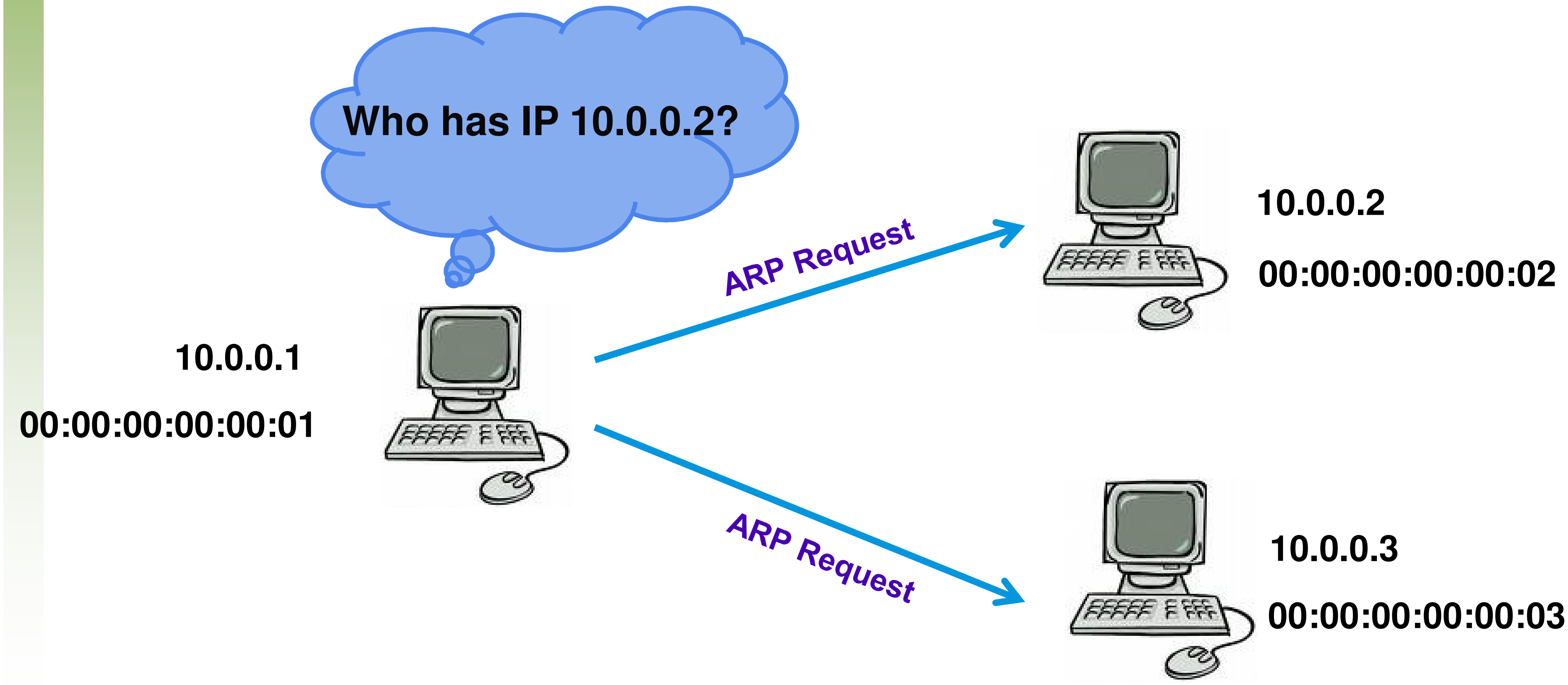
48-bit Ethernet address

- ARP CACHE : IP – MAC Bindings

IP	MAC	TYPE
10.0.0.2	00:00:00:00:00:02	dynamic

How ARP works?

- ❖ **ARP Request is Broadcasted to all the hosts in LAN**



Who has IP 10.0.0.2?

10.0.0.1

00:00:00:00:00:01

ARP Request

ARP Request

10.0.0.2

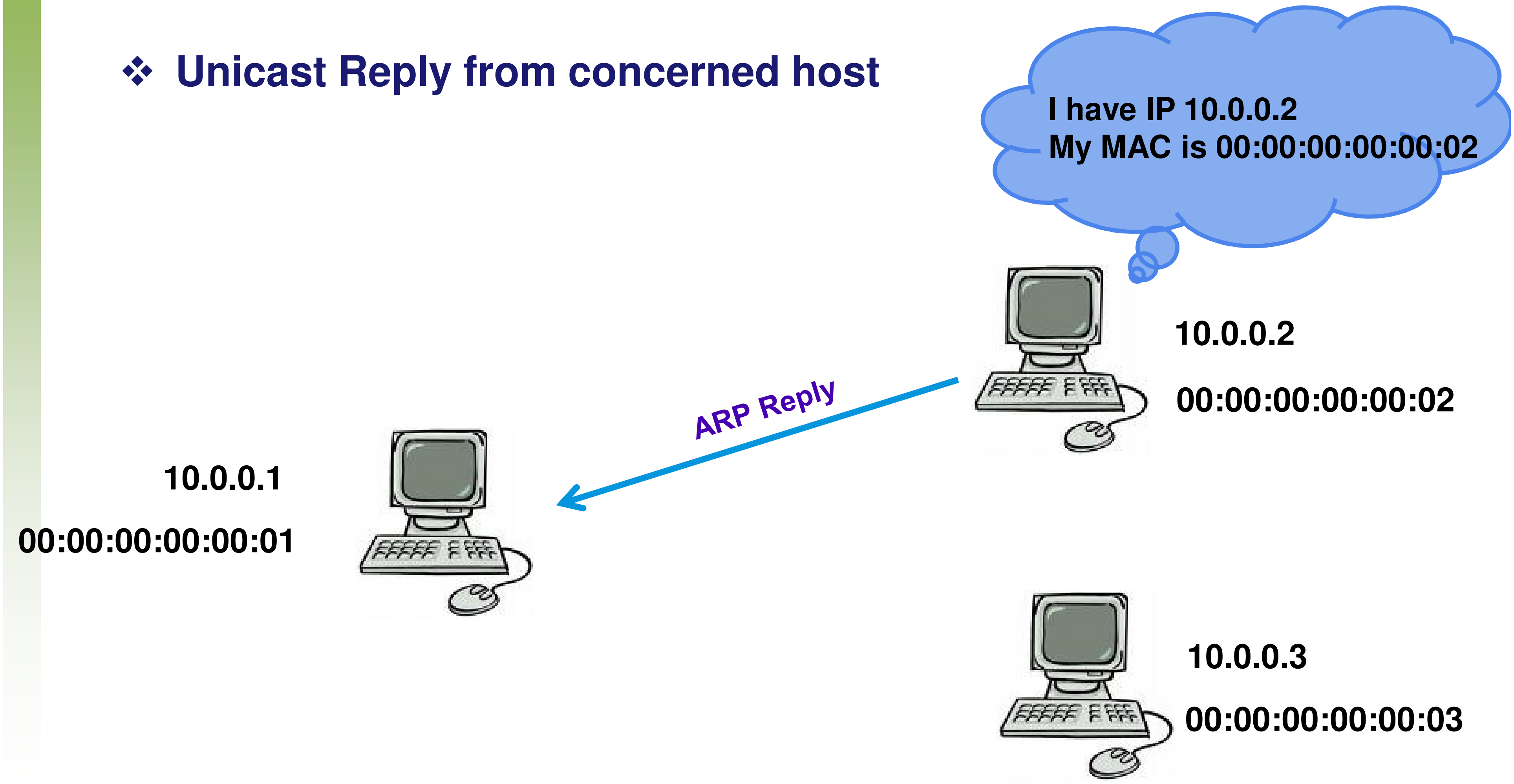
00:00:00:00:00:02

10.0.0.3

00:00:00:00:00:03

How ARP works?

- Unicast Reply from concerned host



I have IP 10.0.0.2
My MAC is 00:00:00:00:00:02

10.0.0.2

00:00:00:00:00:02

ARP Reply

10.0.0.1

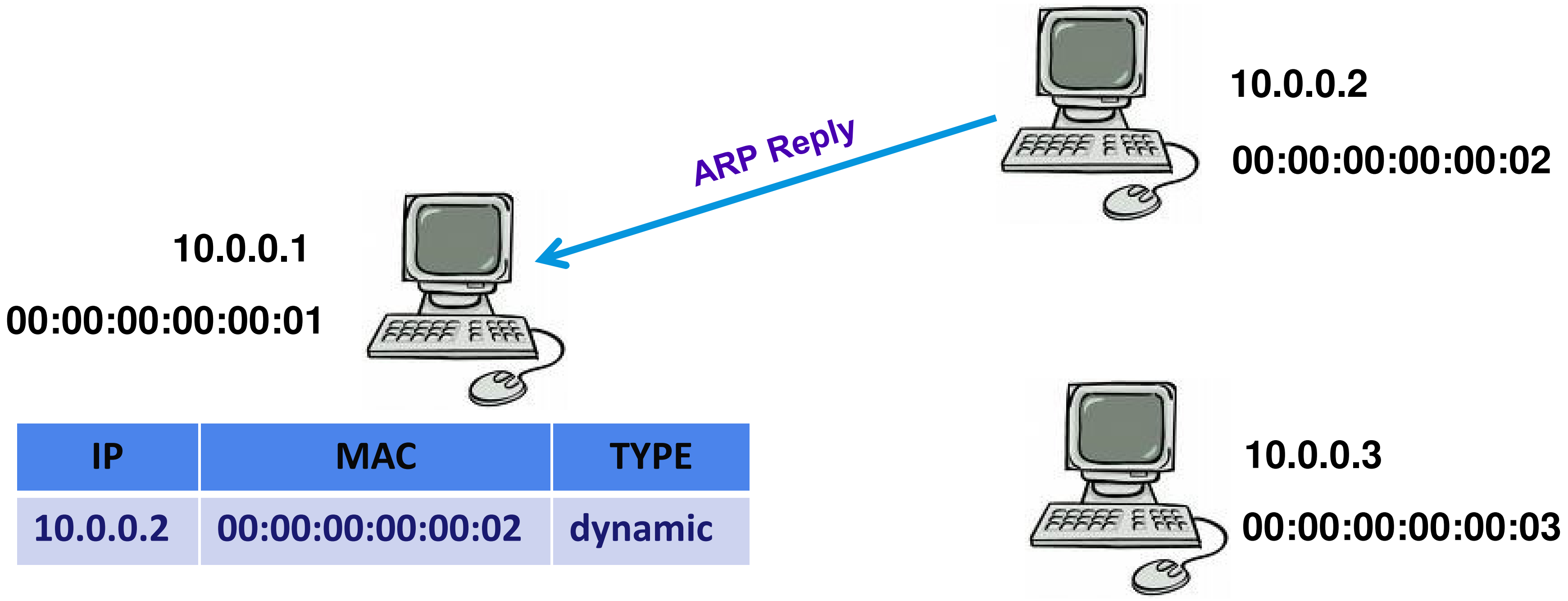
00:00:00:00:00:01

10.0.0.3

00:00:00:00:00:03

What is ARP cache?

- ARP cache : updated



10.0.0.2

00:00:00:00:00:02

ARP Reply

10.0.0.1

00:00:00:00:00:01

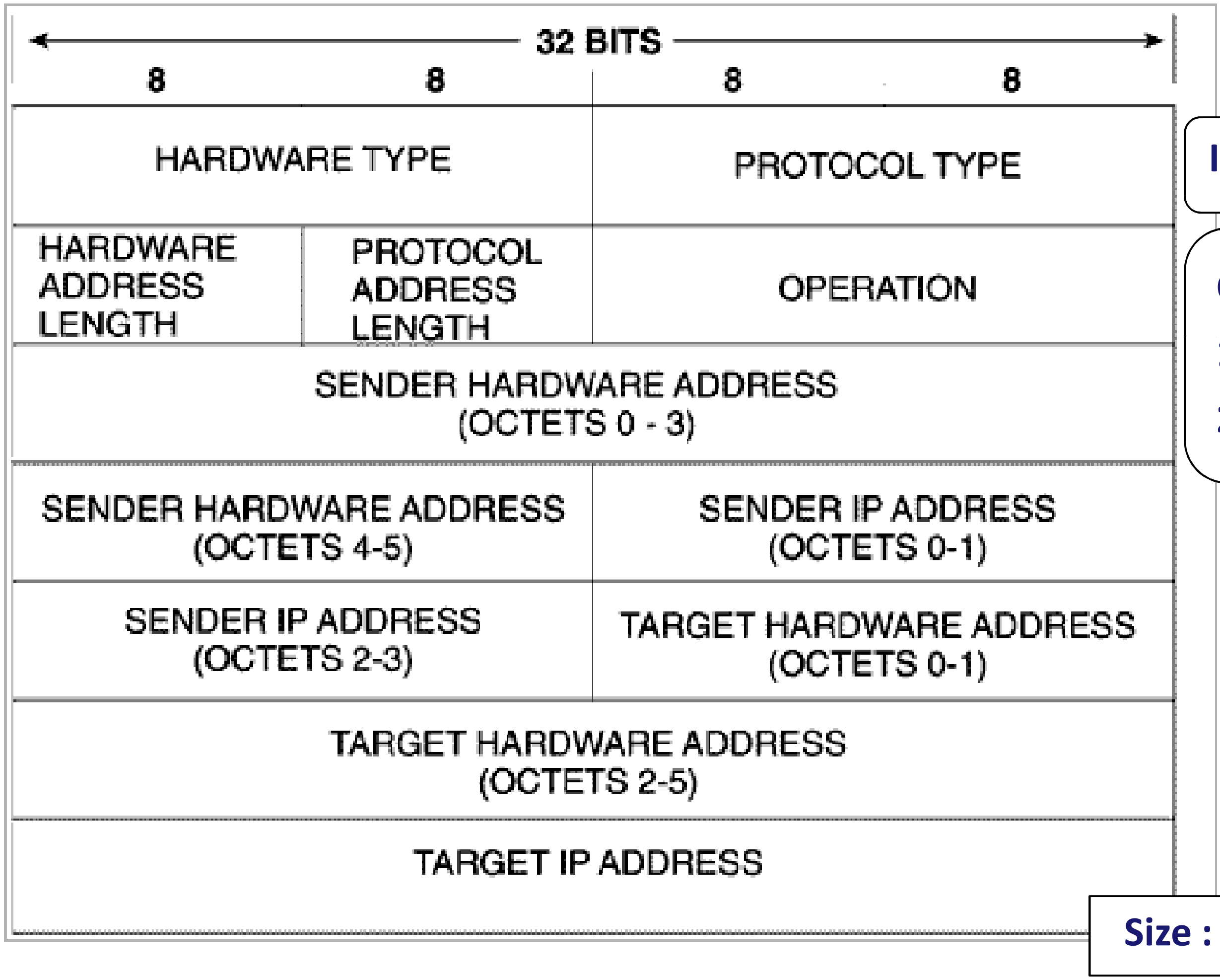
10.0.0.3

00:00:00:00:00:03

IP	MAC	TYPE
10.0.0.2	00:00:00:00:00:02	dynamic

ARP Packet

Ethernet : 1



32 BITS

8	8	8	8
HARDWARE TYPE		PROTOCOL TYPE	
HARDWARE ADDRESS LENGTH	PROTOCOL ADDRESS LENGTH	OPERATION	
SENDER HARDWARE ADDRESS (OCTETS 0 - 3)			
SENDER HARDWARE ADDRESS (OCTETS 4-5)		SENDER IP ADDRESS (OCTETS 0-1)	
SENDER IP ADDRESS (OCTETS 2-3)		TARGET HARDWARE ADDRESS (OCTETS 0-1)	
TARGET HARDWARE ADDRESS (OCTETS 2-5)			
TARGET IP ADDRESS			

IP : 0X800

OPCODE

1: ARP Request

2: ARP Reply

Size : 28 bytes

Why is ARP vulnerable?

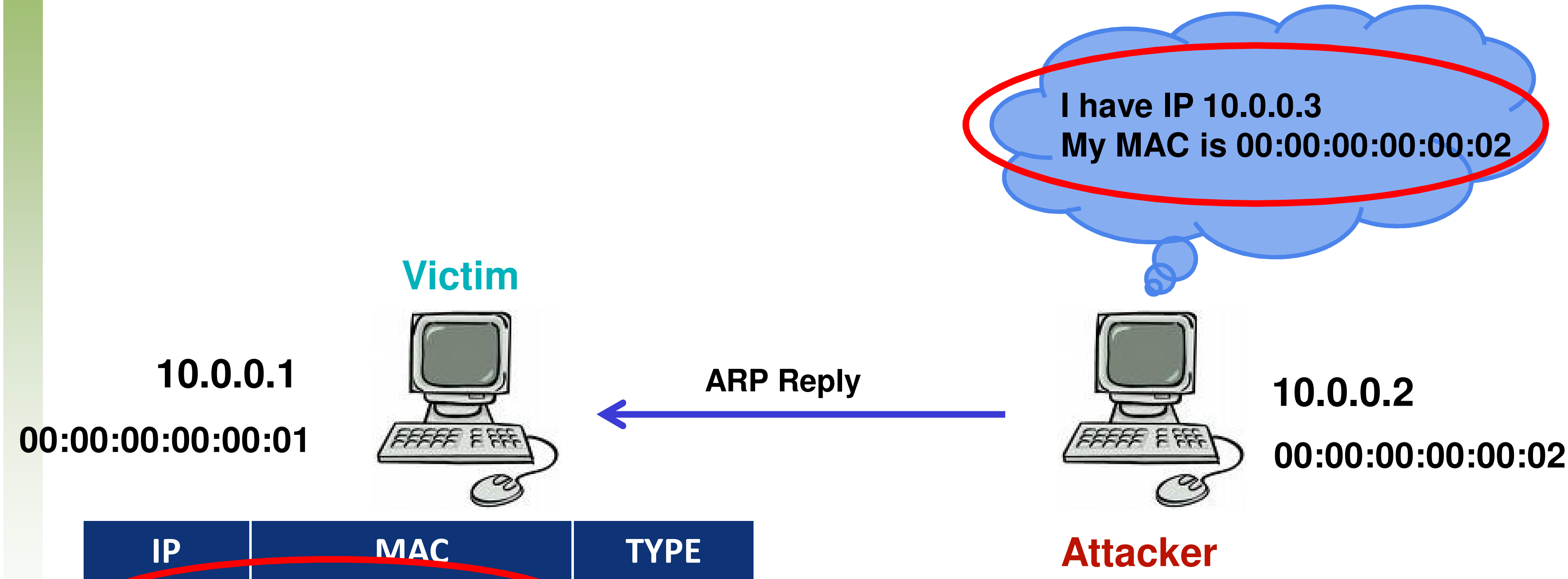
- ARP is a stateless protocol
  - Hosts cache all ARP replies sent to them even if they had not sent an explicit ARP request for it.
- No mechanism to authenticate their peer

ARP-based Attacks

- ARP Spoofing
- Man-in-the-Middle Attack
- Denial-of-Service Attack
- MAC Flooding (on Switch)
- ARP Flooding
- DoS by spurious ARP packets

ARP Spoofing

- Attacker sends forged ARP packets to the victim



I have IP 10.0.0.3
My MAC is 00:00:00:00:00:02

Victim

10.0.0.1
00:00:00:00:00:01

ARP Reply

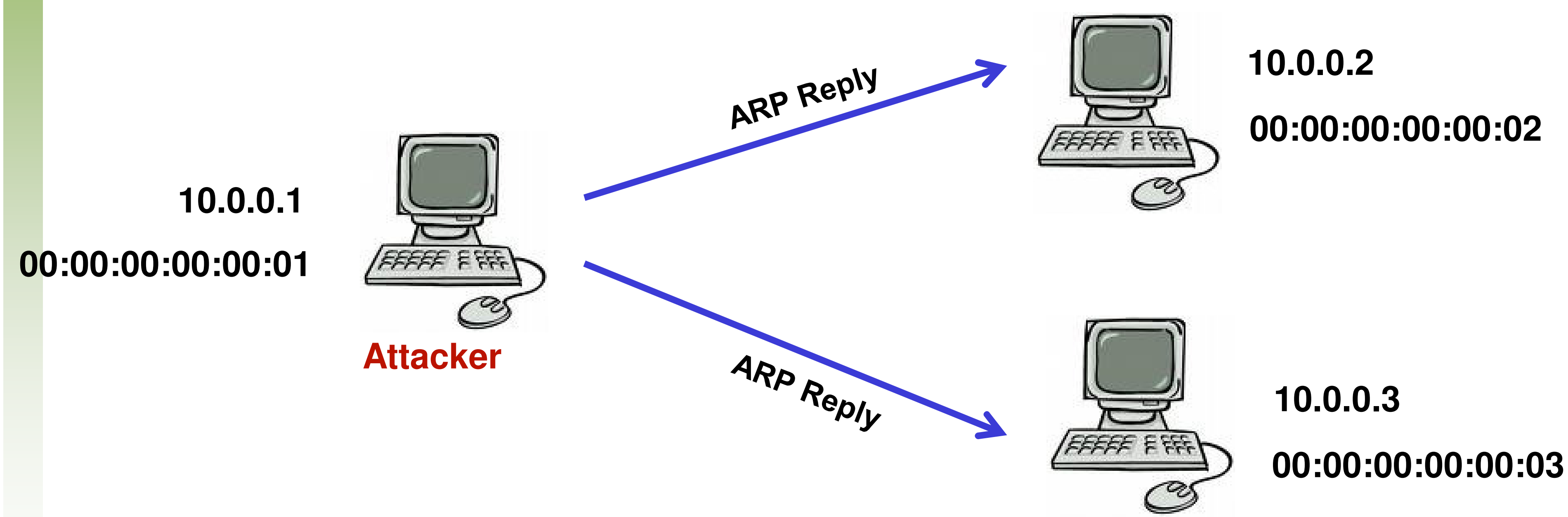
10.0.0.2
00:00:00:00:00:02

Attacker

IP	MAC	TYPE
10.0.0.3	00:00:00:00:00:02	dynamic

Man-in-the-Middle Attack

IP	MAC	TYPE
10.0.0.3	00:00:00:00:00:01	dynamic

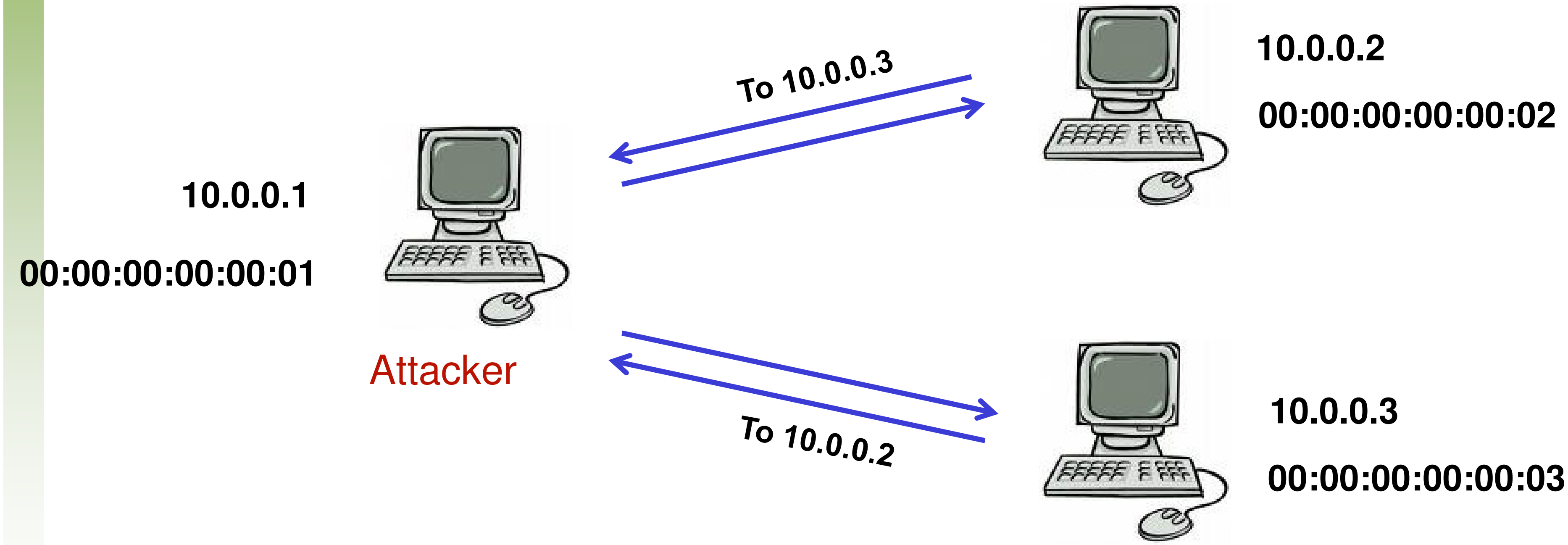


IP	MAC	TYPE
10.0.0.2	00:00:00:00:00:01	dynamic

Man-in-the-Middle Attack

- **Session Hijacking**

IP	MAC	TYPE
10.0.0.3	00:00:00:00:00:01	dynamic



10.0.0.2

00:00:00:00:00:02

To 10.0.0.3

10.0.0.1

00:00:00:00:00:01

Attacker

To 10.0.0.2

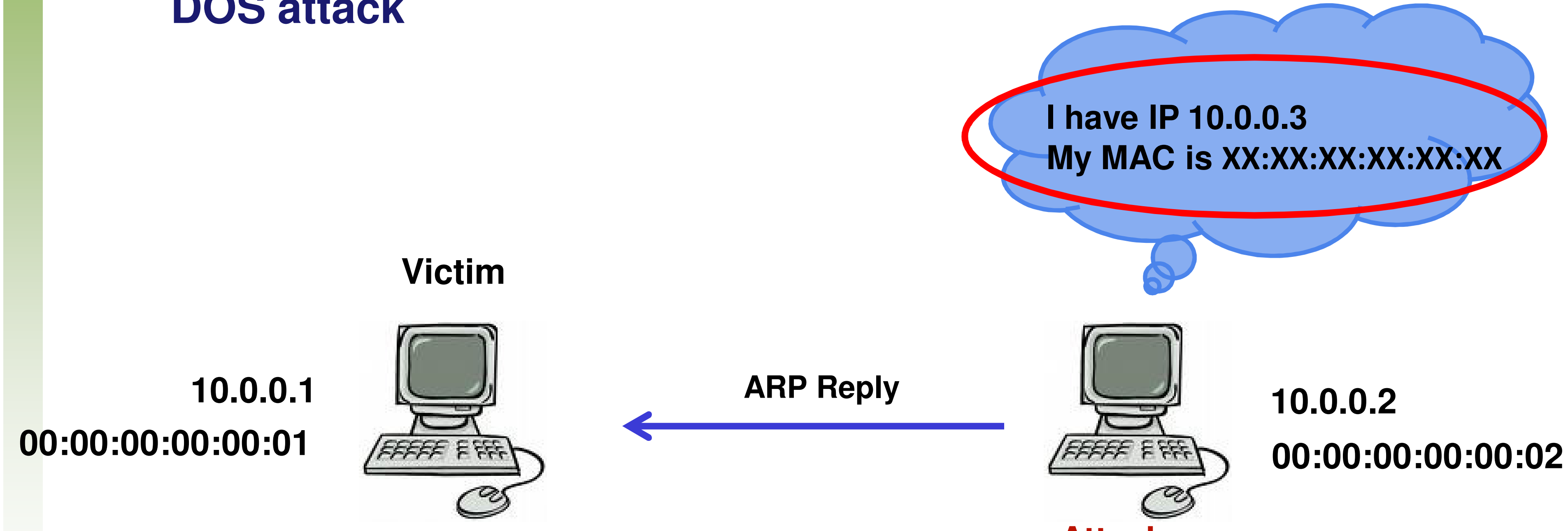
10.0.0.3

00:00:00:00:00:03

IP	MAC	TYPE
10.0.0.2	00:00:00:00:00:01	dynamic

Denial of Service

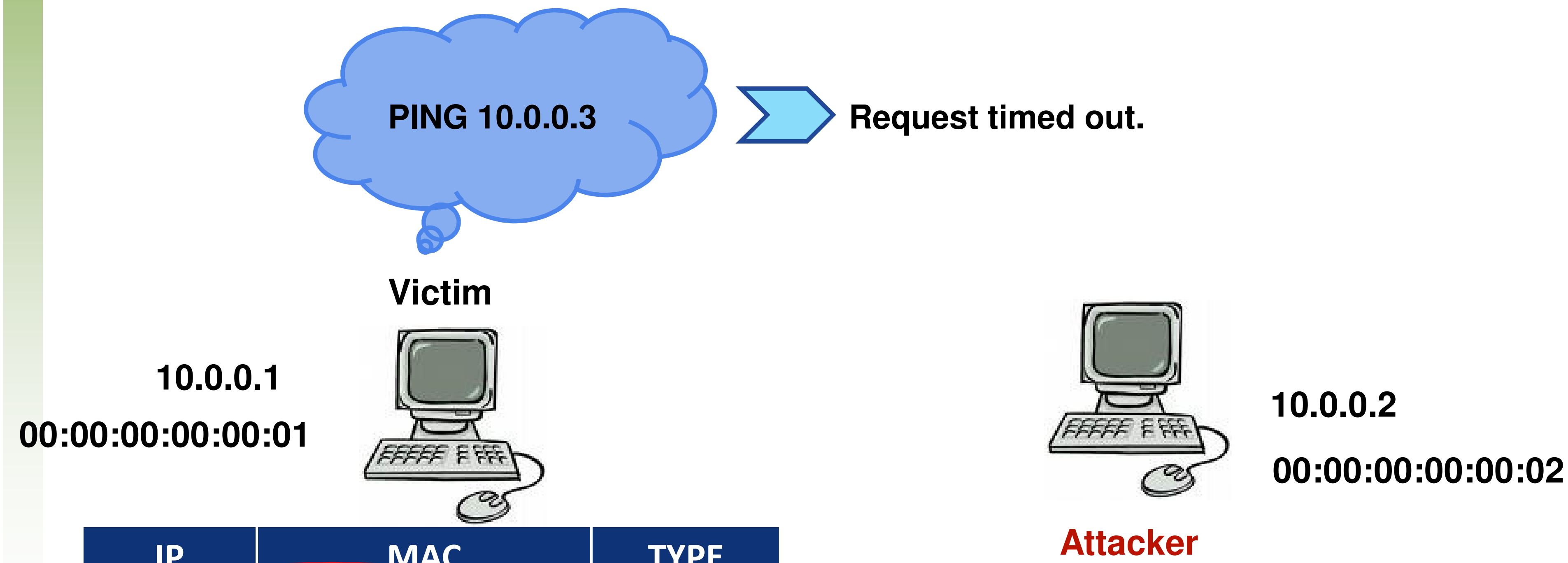
- A malicious entry with a non-existent MAC address can lead to a DOS attack



IP	MAC	TYPE
10.0.0.3	XX:XX:XX:XX:XX:XX	dynamic

Denial of Service

- Victim unable to reach the IP for which the forged packet was sent by the attacker



PING 10.0.0.3

Request timed out.

Victim

10.0.0.1

00:00:00:00:00:01

Attacker

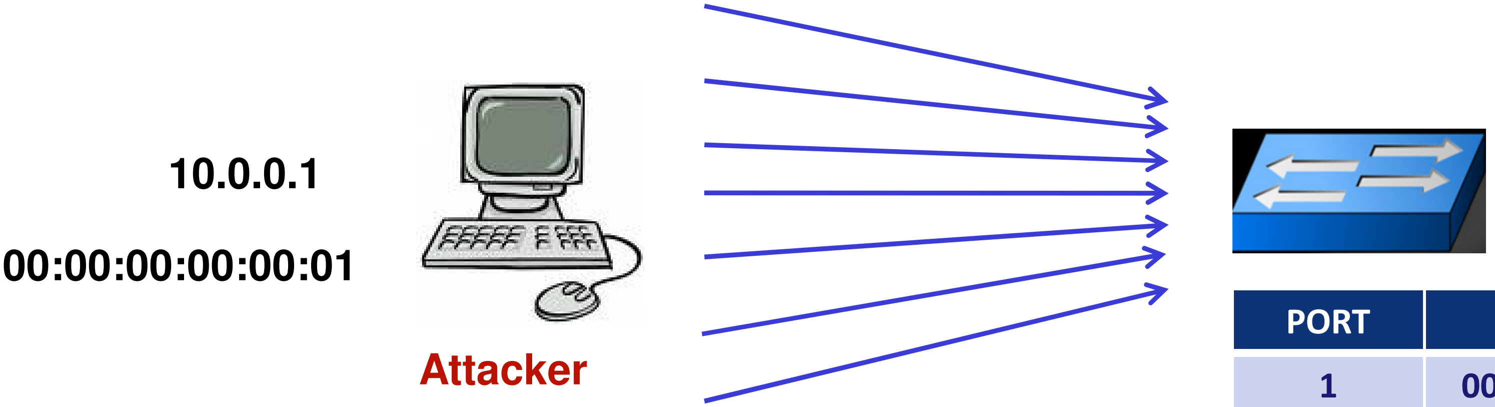
10.0.0.2

00:00:00:00:00:02

IP	MAC	TYPE
10.0.0.3	XX:XX:XX:XX:XX:XX	dynamic

MAC Flooding

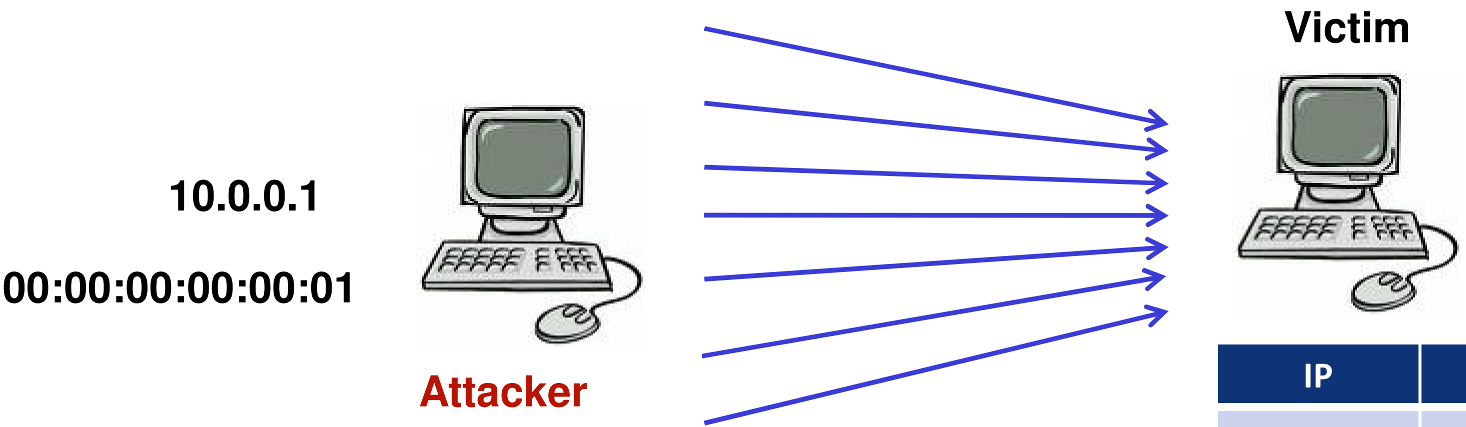
- Attacker bombards the switch with numerous forged ARP packets at an extremely rapid rate such that its CAM table overflows



PORT	MAC
1	00:00:01:01:01:01
2	00:00:02:02:02:02
….	……
…..	…….

ARP Flooding

- Attacker sends numerous forged ARP packets at the victim such that its ARP cache overflows leading to ARP Cache Poisoning
- Results in Victim unable to contact other hosts



Victim

10.0.0.1

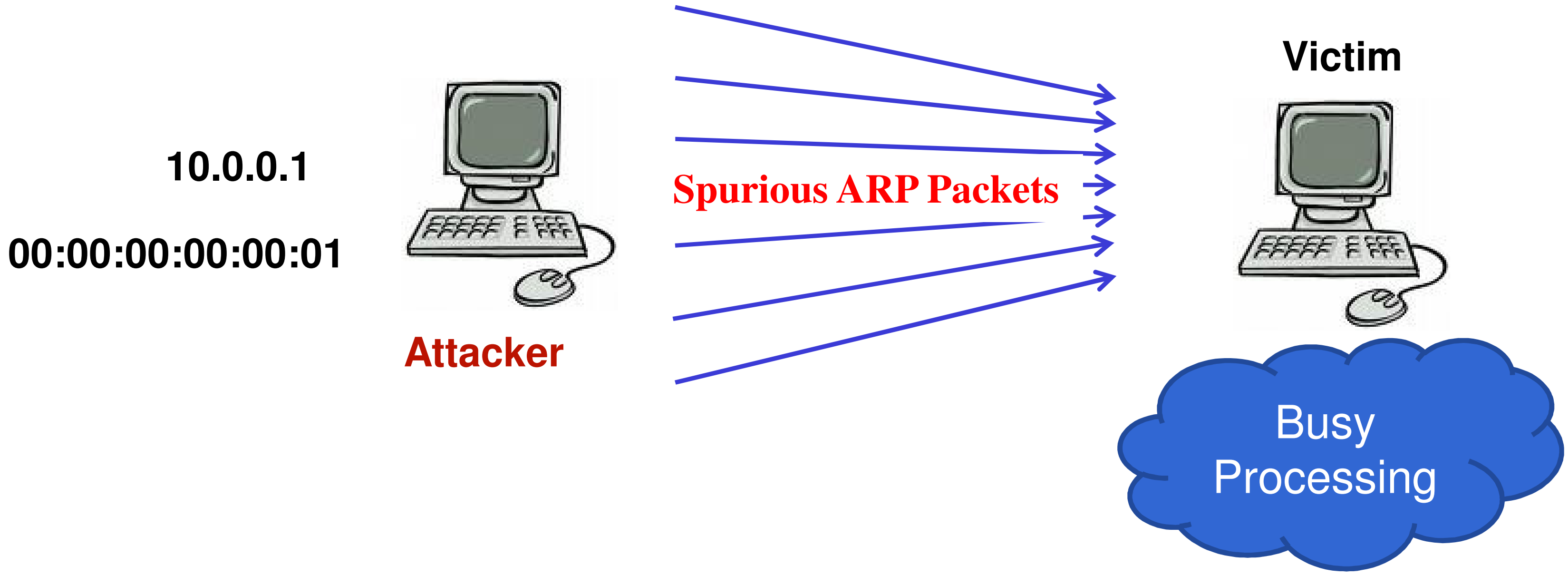
00:00:00:00:00:01

Attacker

IP	MAC
10.0.11.12	00:00:01:01:01:01
10.0.11.15	00:00:02:02:02:02
….	……
…..	…….

DoS by spurious ARP packets

- Attacker sends numerous spurious ARP packets at the victim such that it gets engaged in processing these packets
- Makes the Victim busy and might lead to Denial of Service



10.0.0.1

00:00:00:00:00:01

Attacker

Spurious ARP Packets

Victim

Busy Processing

EXISTING TOOLS AND TECHNIQUES

EXISTING TOOLS AND TECHNIQUES

- **Static ARP Cache entries—Fixed IP-MAC pairs**
  - **Huge administrative effort**
  - **Does not scale on a large dynamic network**
  - **One new/changed host affects all the hosts**
- **Port Security -- Bind switch port to specified MAC address and shut down pot in case of change in MAC address of a transmitter IP.**
  - **If the first packet sent has spoofed IP-MAC pair, then genuine packets may be dropped.**

EXISTING TOOLS AND TECHNIQUES

- **ARPWATCH**
  - maintains a database with IP-MAC mappings
  - any change detected is reported to administrator using syslog/email
- **ARP Defender**
  - Hardware device running ARPWATCH
- **ArpGuard**
  - keeps track of a MAC-IP mappings and alerts changes and invalid mappings

If the first packet sent has spoofed IP-MAC pair, then genuine packets may be dropped.

EXISTING TOOLS AND TECHNIQUES

- Signature and Anomaly based IDS
  - High number of false alarms

- Modifying ARP using Cryptographic Techniques
  - *Secure-ARP* - Digital Signature for authentication
  - *Ticket-based ARP* – Tickets from Ticket-issuing Agents

Calls for Replacement of entire Network Stack

Additional overhead of cryptographic calculations

Change Standard ARP

EXISTING TOOLS AND TECHNIQUES

- Active Spoof Detection Engine
  - Send TCP SYN packets to probe IP-MAC pairs
  - Receive SYN/ACK if port is open or RST if closed
  - No response => malicious host

  Violation of network layering architecture

- Active Man in the Middle Attack Detector
  - IDS finds Systems with IP forwarding enabled
  - Spoof the ARP cache of all such systems: Now all traffic forwarded by such systems reach IDS

  Additional network Traffic

  Difficulty in poisoning ARP cache of the attacker

Motivation: What is Required in an IDS for ARP attacks

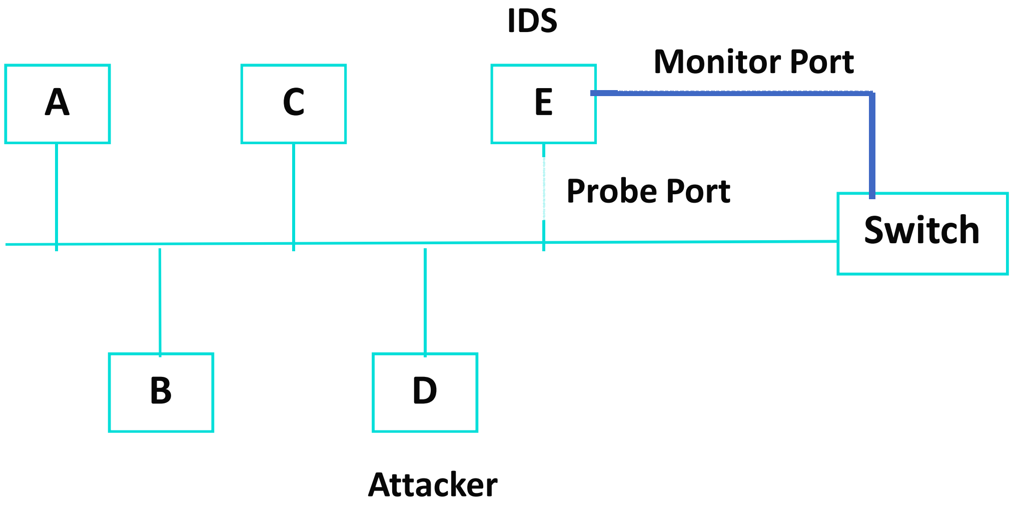
- Should not modify the standard ARP
- Should generate minimal extra traffic in the network
- Should not require patching, installation of extra software in all the systems
- Should detect a large set of LAN based attacks

ARP ATTACK DETECTION

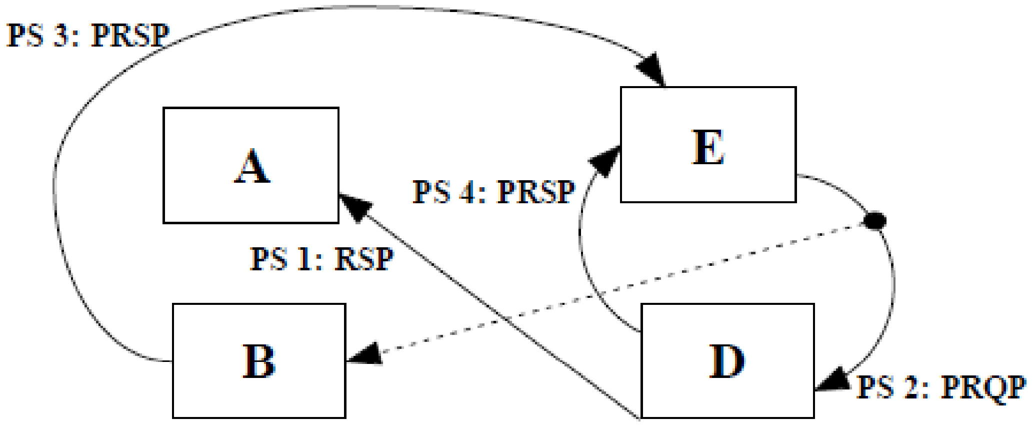
USING DISCRETE EVENT SYSTEM

Network Architecture

- Port Mirroring is enabled at the switch
- E is working as IDS



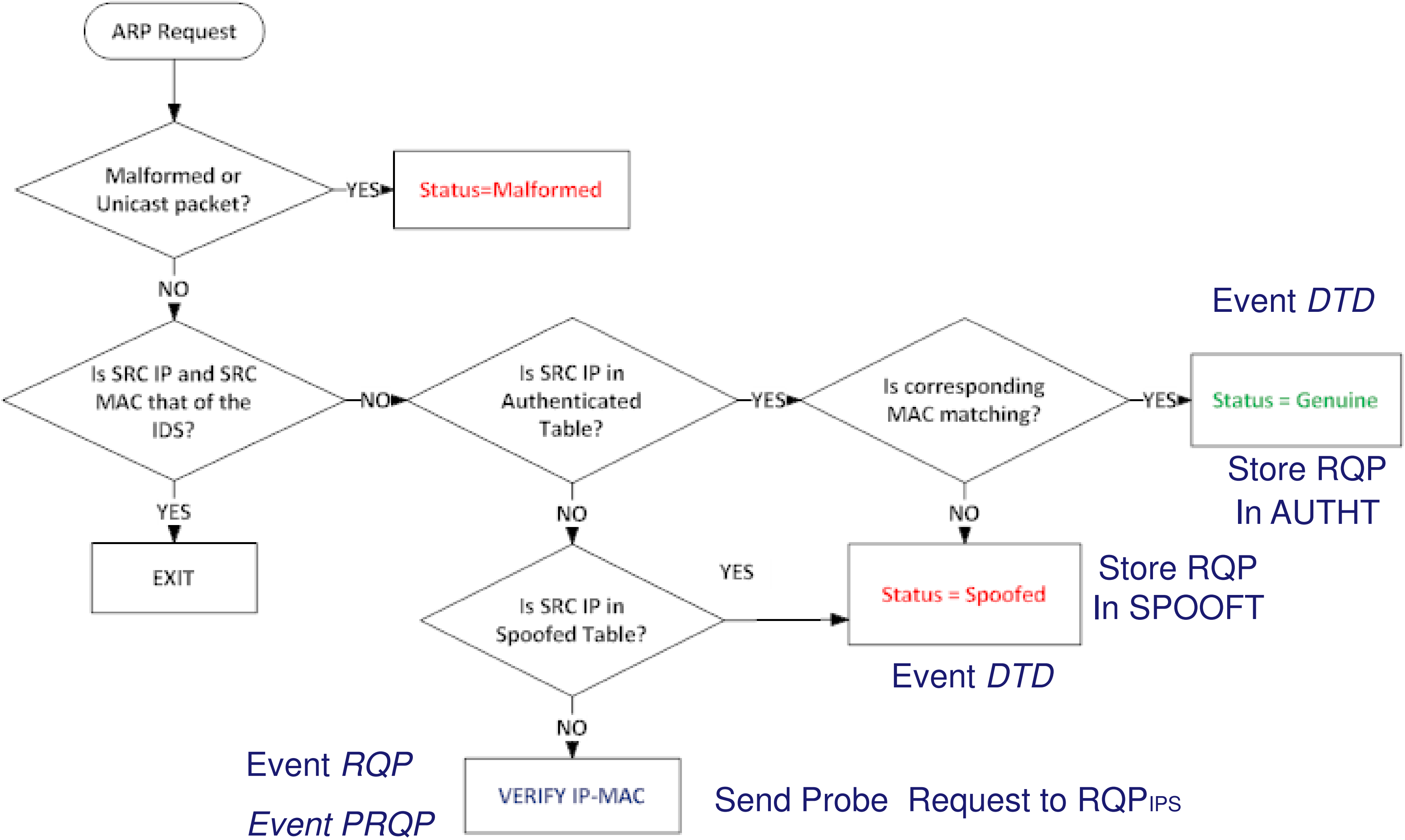
Test Scenario



TABULATION OF THE PACKET SEQUENCES AND EVENTS IN THE EXAMPLE

PS: Events	SRC IP	SRC MAC	Dest IP	Dest MAC
PS 1:RSP	IP B	MAC D	IP A	MAC A
PS 2:$PRQP$	IP E	MAC E	IP B	–
PS 3:$PRSP$	IP B	MAC B	IP E	MAC E
PS 4:$PRSP$	IP B	MAC D	IP E	MAC E

ARP Request Handler


ARP Request
Malformed or Unicast packet?
YES
Status=Malformed
NO
Is SRC IP and SRC MAC that of the IDS?
YES
EXIT
NO
Is SRC IP in Authenticated Table?
YES
Is corresponding MAC matching?
YES
Status = Genuine
NO
Status = Spoofed
NO
Is SRC IP in Spoofed Table?
YES
NO
VERIFY IP-MAC

Event *DTD*

Store RQP
In AUTHT

Store RQP
In SPOOFT

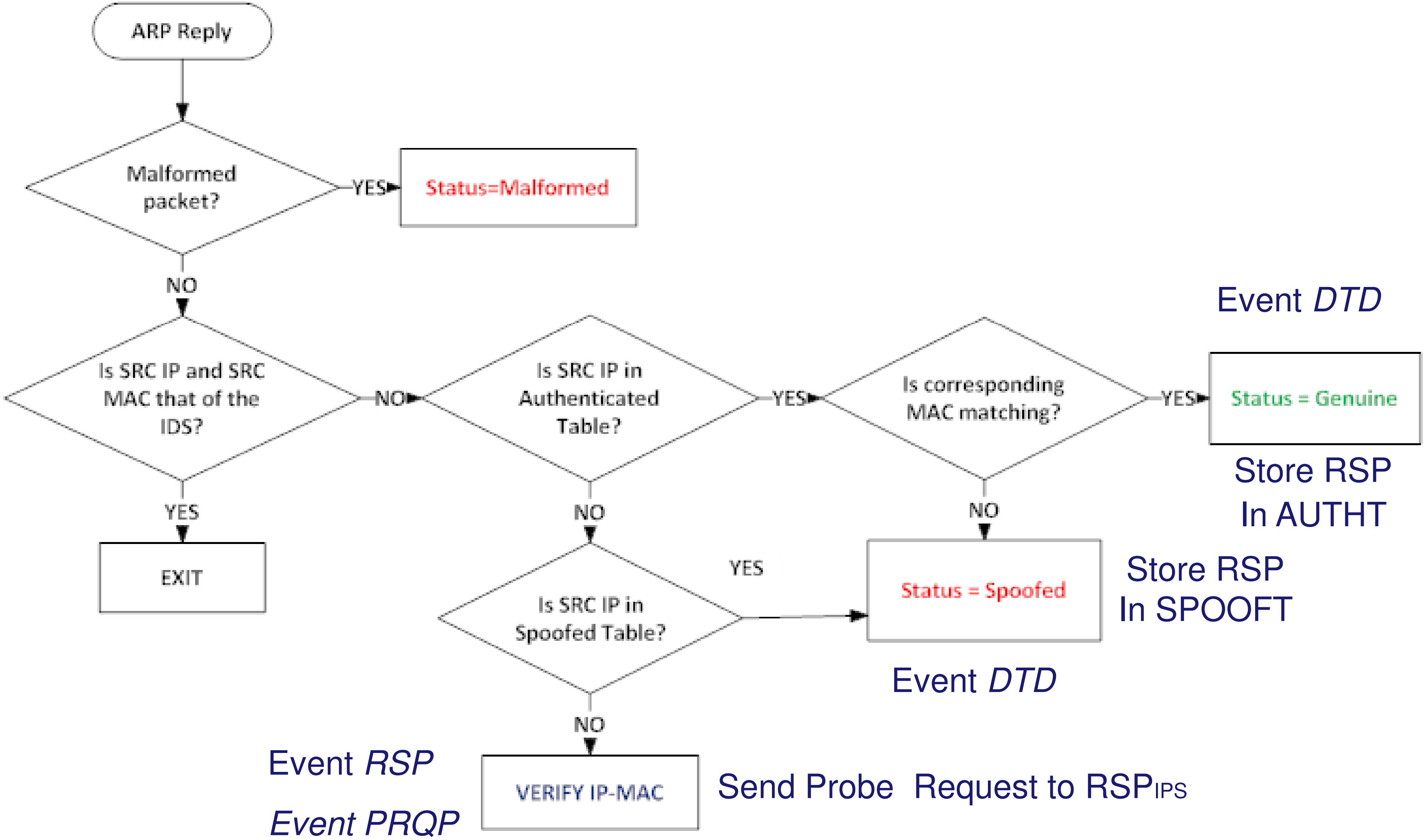
Event *DTD*

Event *RQP*

Event PRQP

Send Probe Request to RQP_{IPS}

ARP Response Handler


ARP Reply
Malformed packet?
YES
Status=Malformed
NO
Is SRC IP and SRC MAC that of the IDS?
NO
Is SRC IP in Authenticated Table?
YES
Is corresponding MAC matching?
YES
Status = Genuine
YES
EXIT
NO
NO
Is SRC IP in Spoofed Table?
YES
Status = Spoofed
NO
VERIFY IP-MAC

Event *DTD*

Store RSP
In AUTHT

Store RSP
In SPOOFT

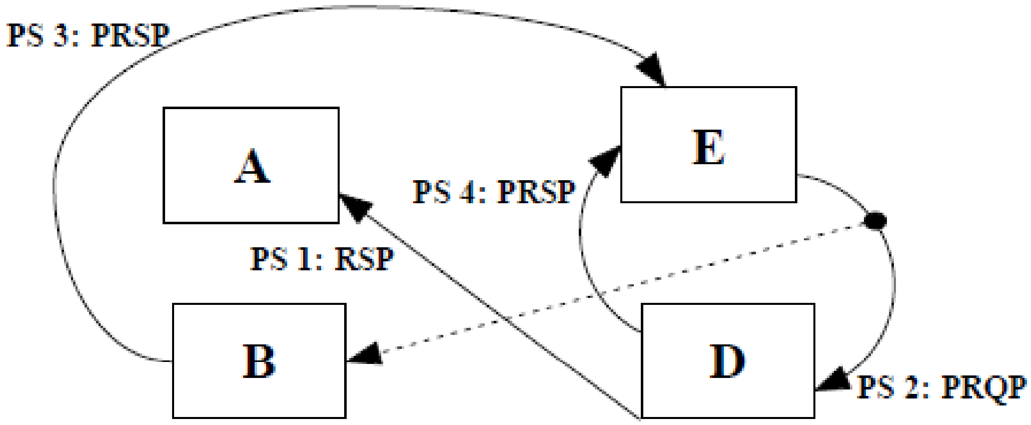
Event *DTD*

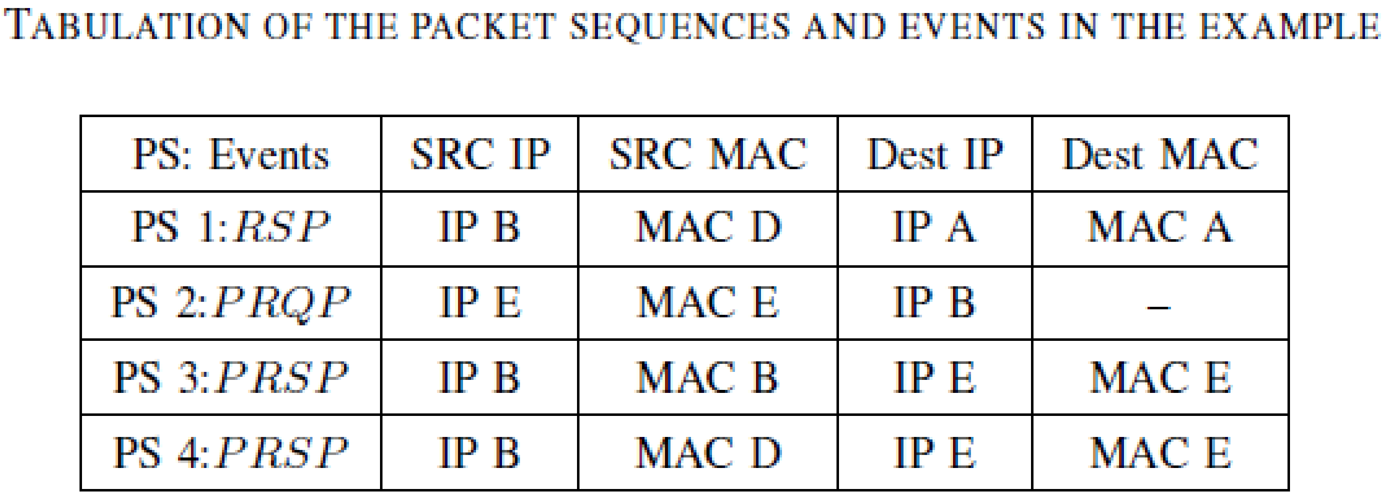
Event *RSP*

Event PRQP

Send Probe Request to RSPIPS

Test Scenario (Revised)

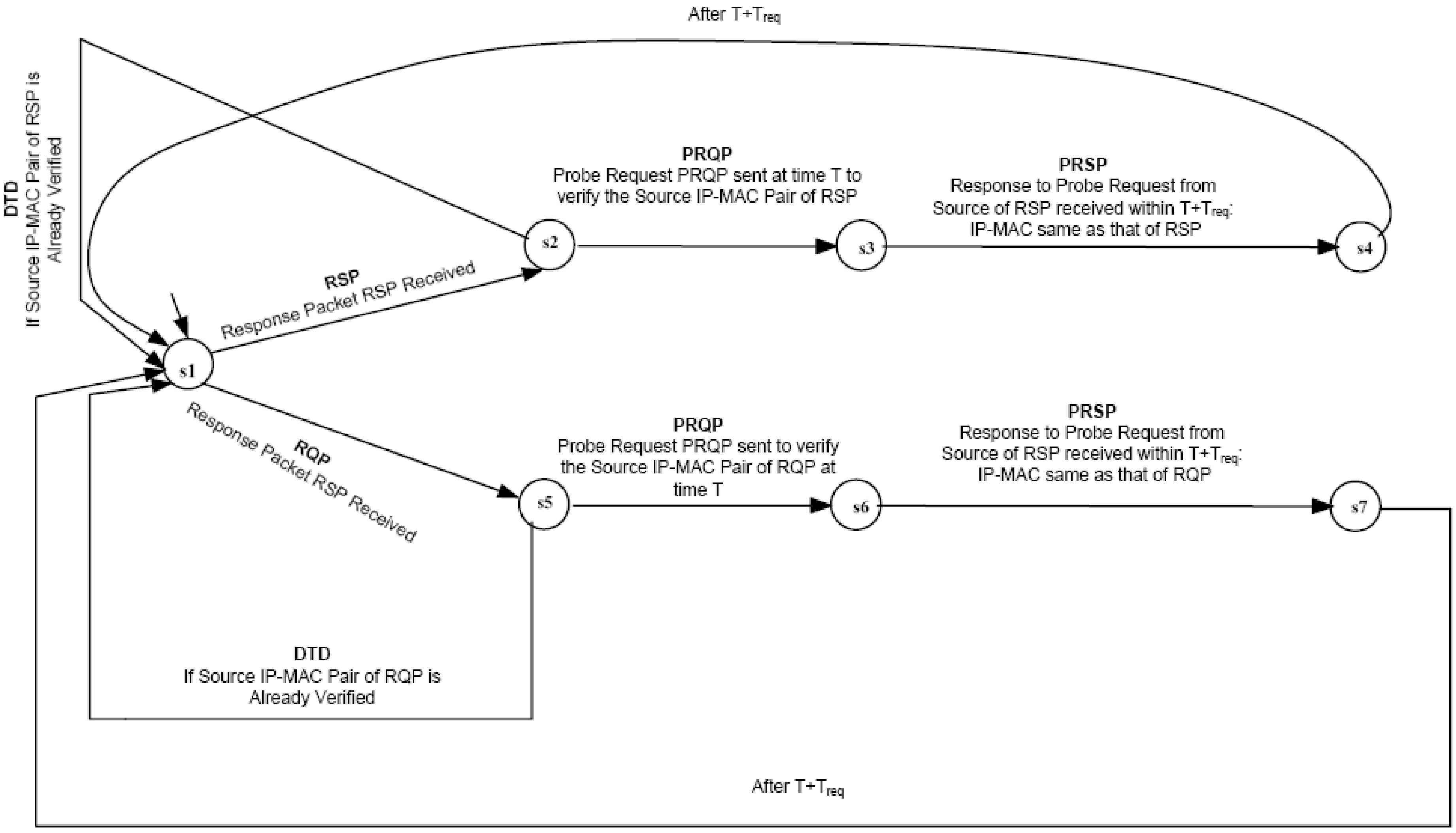




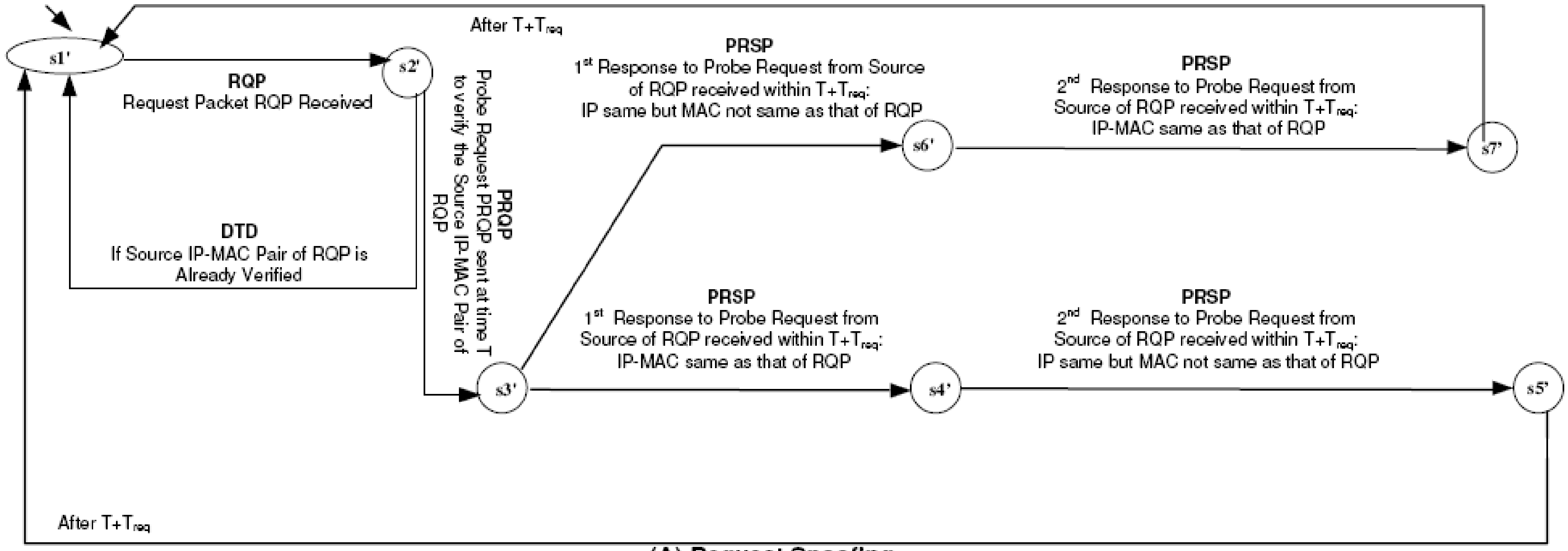
Tabulation of the packet sequences and events in the example

PS: Events	SRC IP	SRC MAC	Dest IP	Dest MAC
PS 1:RSP	IP B	MAC D	IP A	MAC A
PS 2:$PRQP$	IP E	MAC E	IP B	–
PS 3:$PRSP$	IP B	MAC B	IP E	MAC E
PS 4:$PRSP$	IP B	MAC D	IP E	MAC E

DES model: Normal Condition


After $T+T_{req}$
DTD
If Source IP-MAC Pair of RSP is Already Verified
PRQP
Probe Request PRQP sent at time T to verify the Source IP-MAC Pair of RSP
PRSP
Response to Probe Request from Source of RSP received within $T+T_{req}$: IP-MAC same as that of RSP
s2
s3
s4
RSP
Response Packet RSP Received
s1
Response Packet RSP Received
RQP
PRQP
Probe Request PRQP sent to verify the Source IP-MAC Pair of RQP at time T
PRSP
Response to Probe Request from Source of RSP received within $T+T_{req}$: IP-MAC same as that of RQP
s5
s6
s7
DTD
If Source IP-MAC Pair of RQP is Already Verified
After $T+T_{req}$

DES model: Request Spoofing



After $T+T_{req}$

RQP
Request Packet RQP Received

DTD
If Source IP-MAC Pair of RQP is Already Verified

PRQP
Probe Request PRQP sent at time T to verify the Source IP-MAC Pair of RQP

PRSP
1st Response to Probe Request from Source of RQP received within $T+T_{req}$:
IP same but MAC not same as that of RQP

PRSP
2nd Response to Probe Request from Source of RQP received within $T+T_{req}$:
IP-MAC same as that of RQP

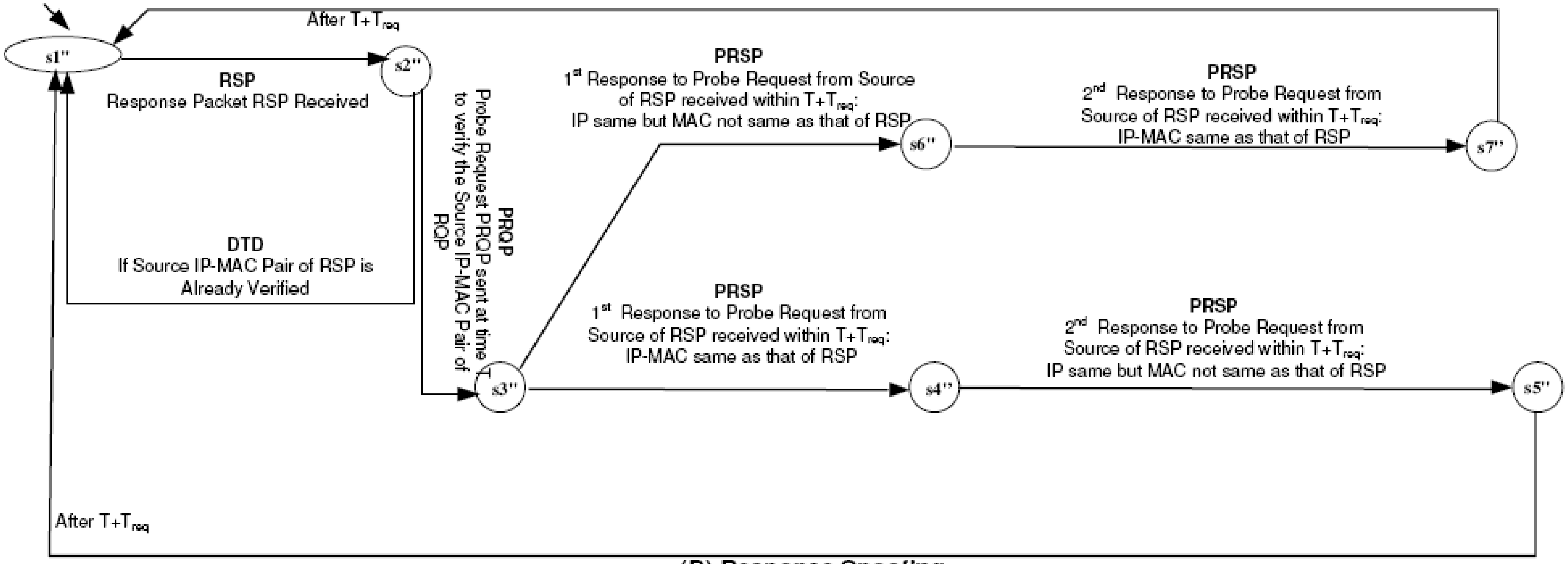
PRSP
1st Response to Probe Request from Source of RQP received within $T+T_{req}$:
IP-MAC same as that of RQP

PRSP
2nd Response to Probe Request from Source of RQP received within $T+T_{req}$:
IP same but MAC not same as that of RQP

After $T+T_{req}$

(A) Request Spoofing

DES model: Response Spoofing



After $T+T_{req}$

RSP
Response Packet RSP Received

DTD
If Source IP-MAC Pair of RSP is Already Verified

PRQP
Probe Request PRQP sent at time T to verify the Source IP-MAC Pair of RQP

PRSP
1st Response to Probe Request from Source of RSP received within $T+T_{req}$:
IP same but MAC not same as that of RSP

PRSP
2nd Response to Probe Request from Source of RSP received within $T+T_{req}$:
IP-MAC same as that of RSP

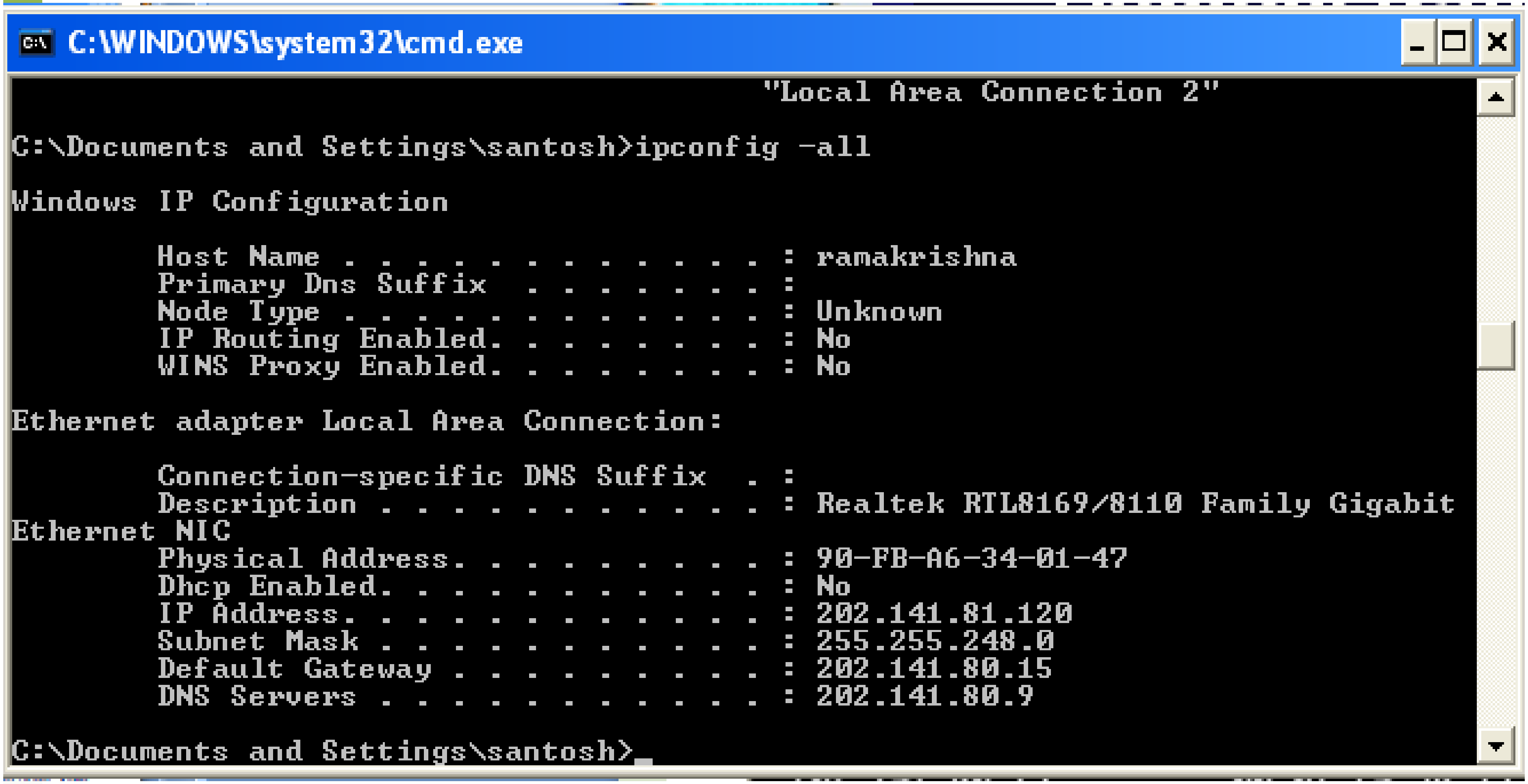
PRSP
1st Response to Probe Request from Source of RSP received within $T+T_{req}$:
IP-MAC same as that of RSP

PRSP
2nd Response to Probe Request from Source of RSP received within $T+T_{req}$:
IP same but MAC not same as that of RSP

After $T+T_{req}$

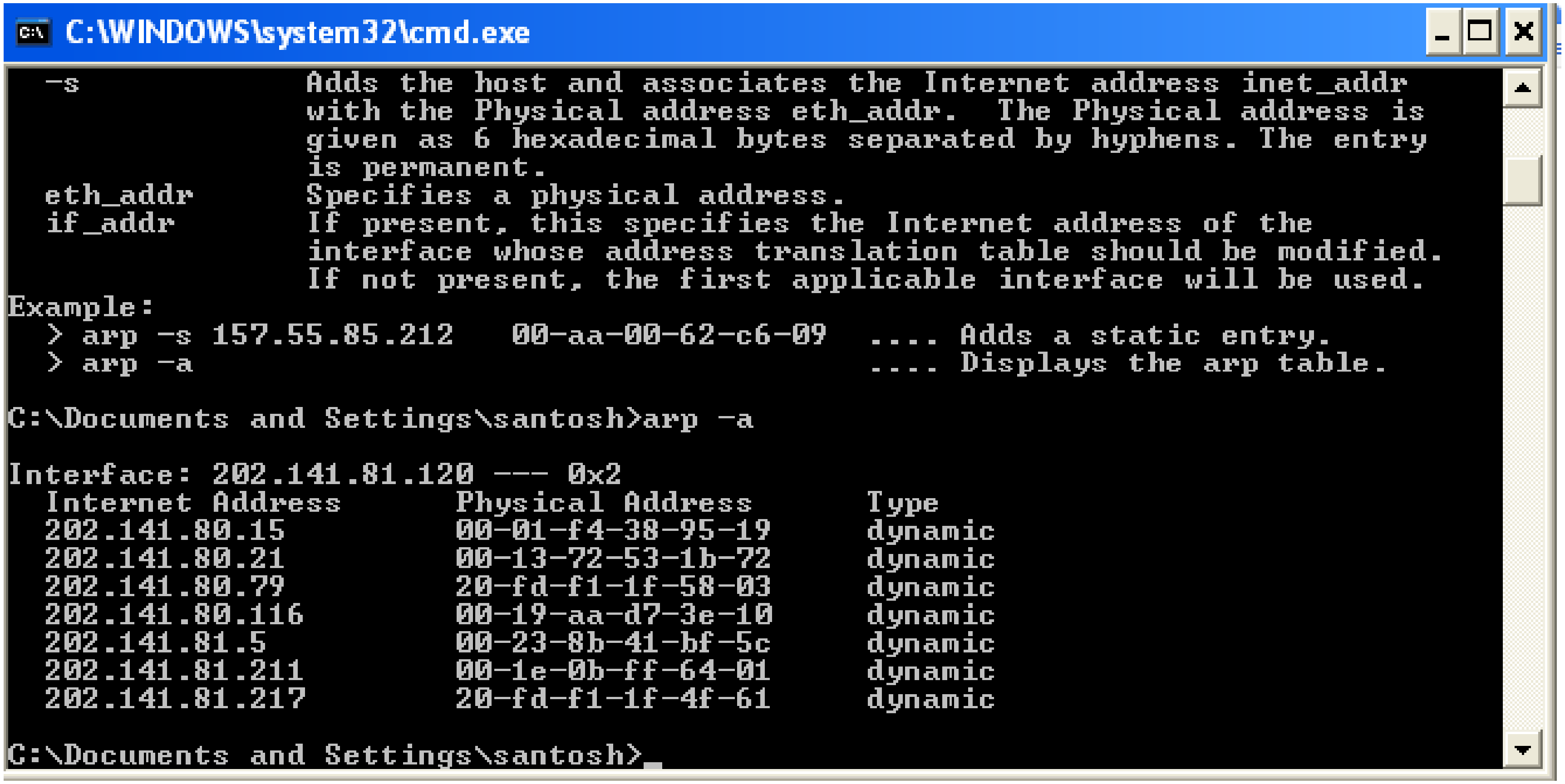
(B) Response Spoofing

Demonstration by Screen Captures



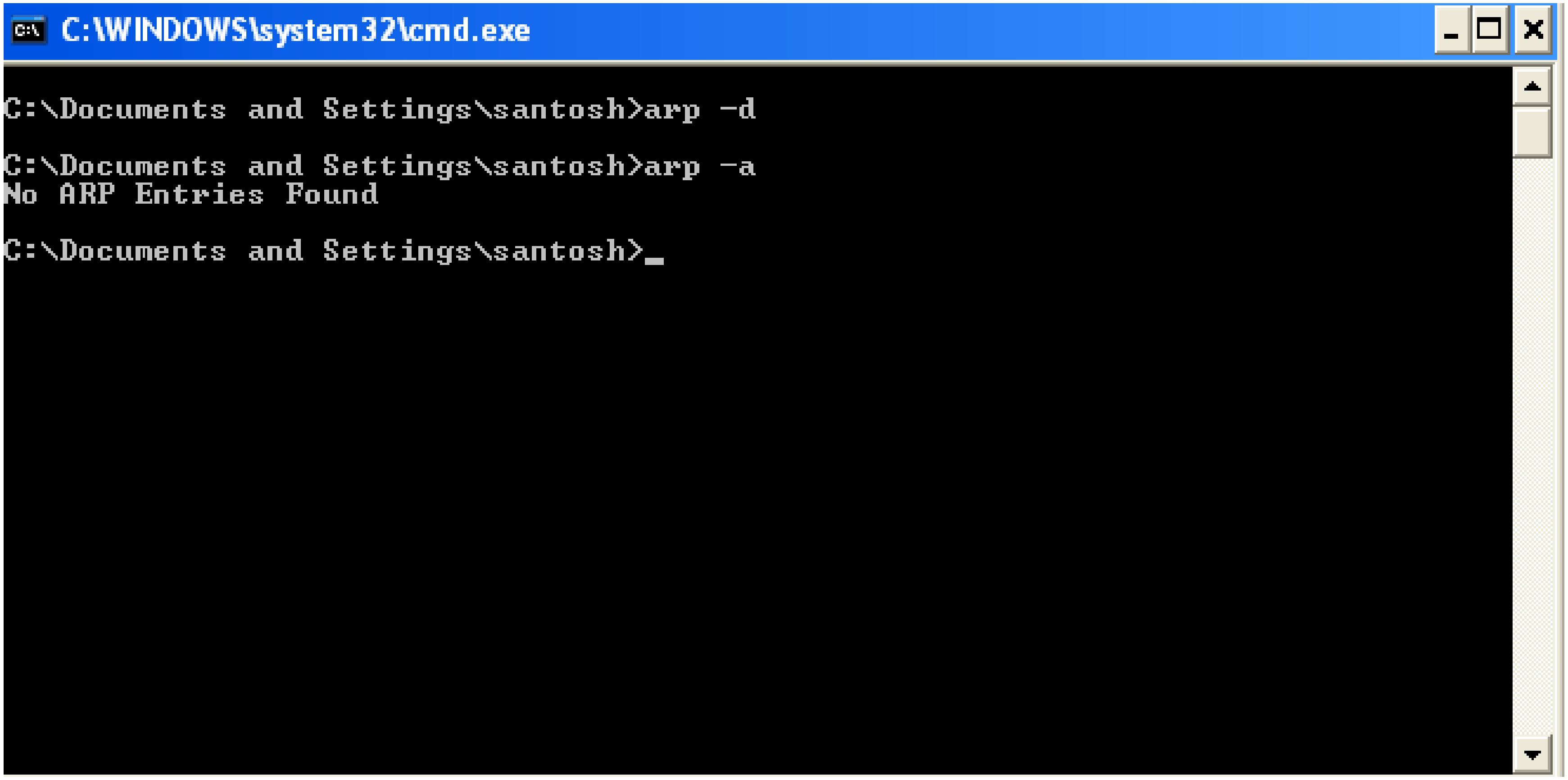
```
                                          "Local Area Connection 2"

C:\Documents and Settings\santosh>ipconfig -all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : ramakrishna
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Realtek RTL8169/8110 Family Gigabit
Ethernet NIC
        Physical Address. . . . . . . . . : 90-FB-A6-34-01-47
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 202.141.81.120
        Subnet Mask . . . . . . . . . . . : 255.255.248.0
        Default Gateway . . . . . . . . . : 202.141.80.15
        DNS Servers . . . . . . . . . . . : 202.141.80.9

C:\Documents and Settings\santosh>_
```

Demonstration by Screen Captures



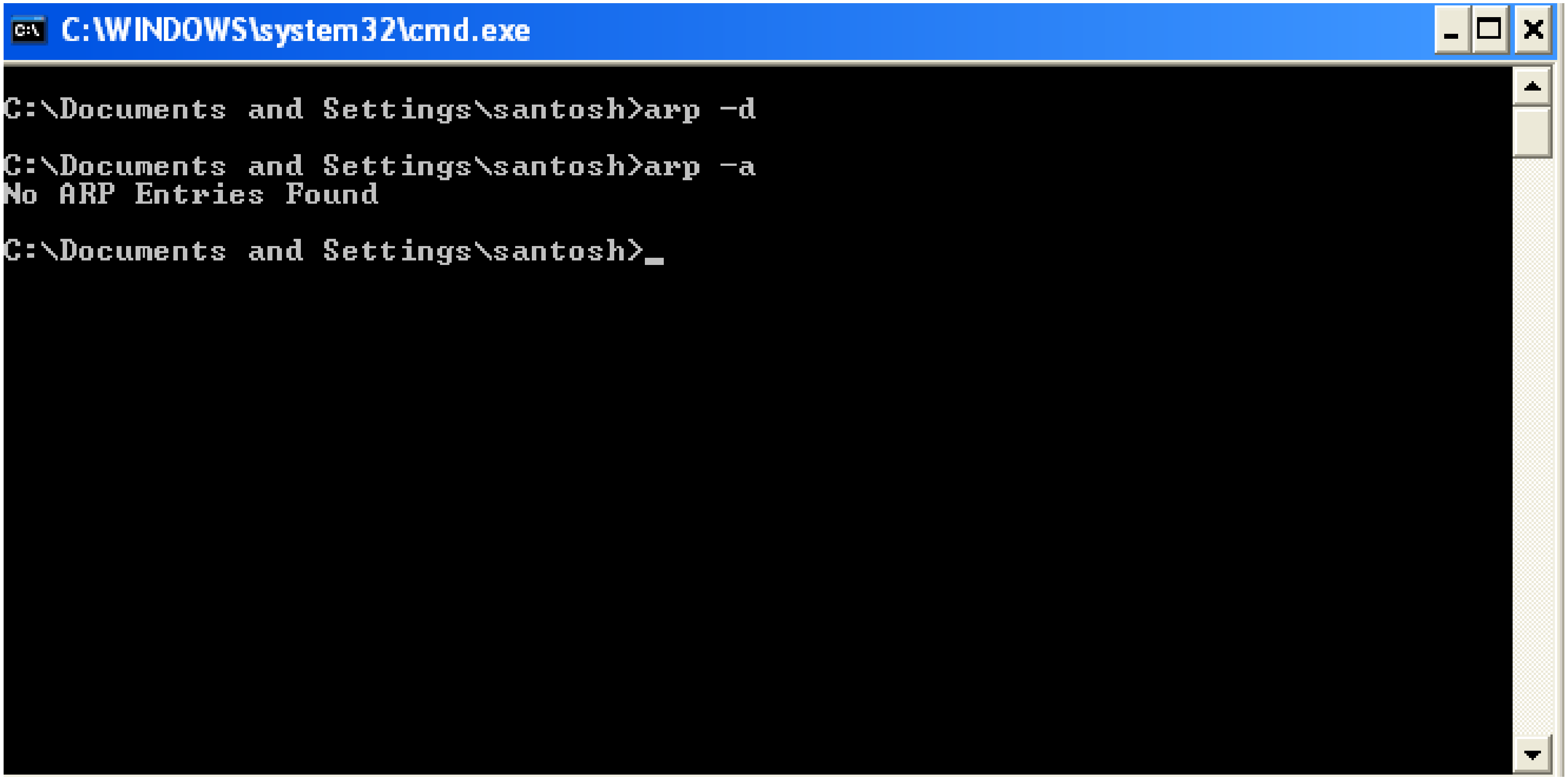
```
  -s              Adds the host and associates the Internet address inet_addr
                  with the Physical address eth_addr.  The Physical address is
                  given as 6 hexadecimal bytes separated by hyphens. The entry
                  is permanent.
  eth_addr        Specifies a physical address.
  if_addr         If present, this specifies the Internet address of the
                  interface whose address translation table should be modified.
                  If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.

C:\Documents and Settings\santosh>arp -a

Interface: 202.141.81.120 --- 0x2
  Internet Address      Physical Address      Type
  202.141.80.15         00-01-f4-38-95-19     dynamic
  202.141.80.21         00-13-72-53-1b-72     dynamic
  202.141.80.79         20-fd-f1-1f-58-03     dynamic
  202.141.80.116        00-19-aa-d7-3e-10     dynamic
  202.141.81.5          00-23-8b-41-bf-5c     dynamic
  202.141.81.211        00-1e-0b-ff-64-01     dynamic
  202.141.81.217        20-fd-f1-1f-4f-61     dynamic

C:\Documents and Settings\santosh>_
```

Demonstration by Screen Captures



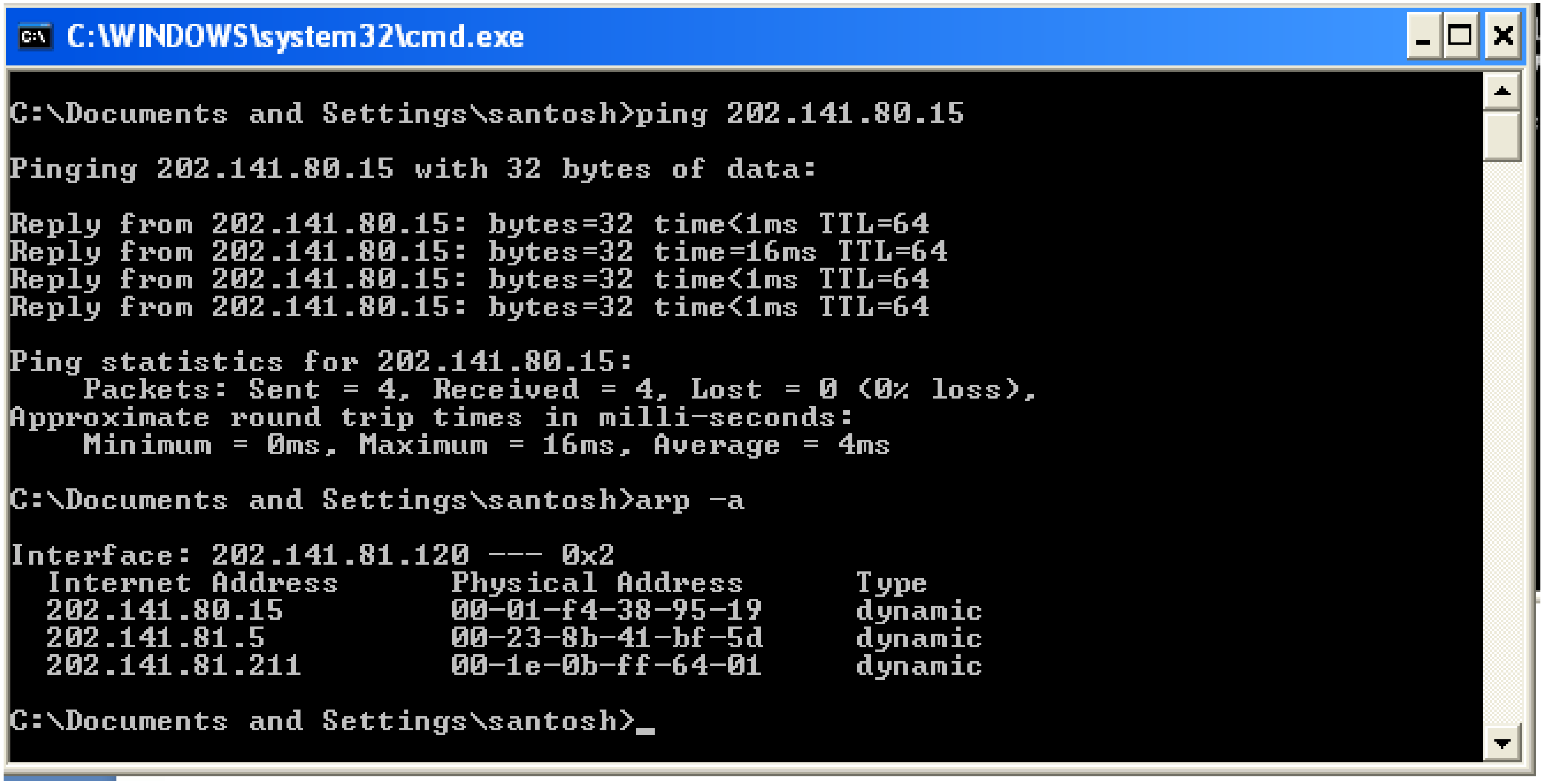
```
C:\Documents and Settings\santosh>arp -d

C:\Documents and Settings\santosh>arp -a
No ARP Entries Found

C:\Documents and Settings\santosh>_
```

Demonstration by Screen Captures



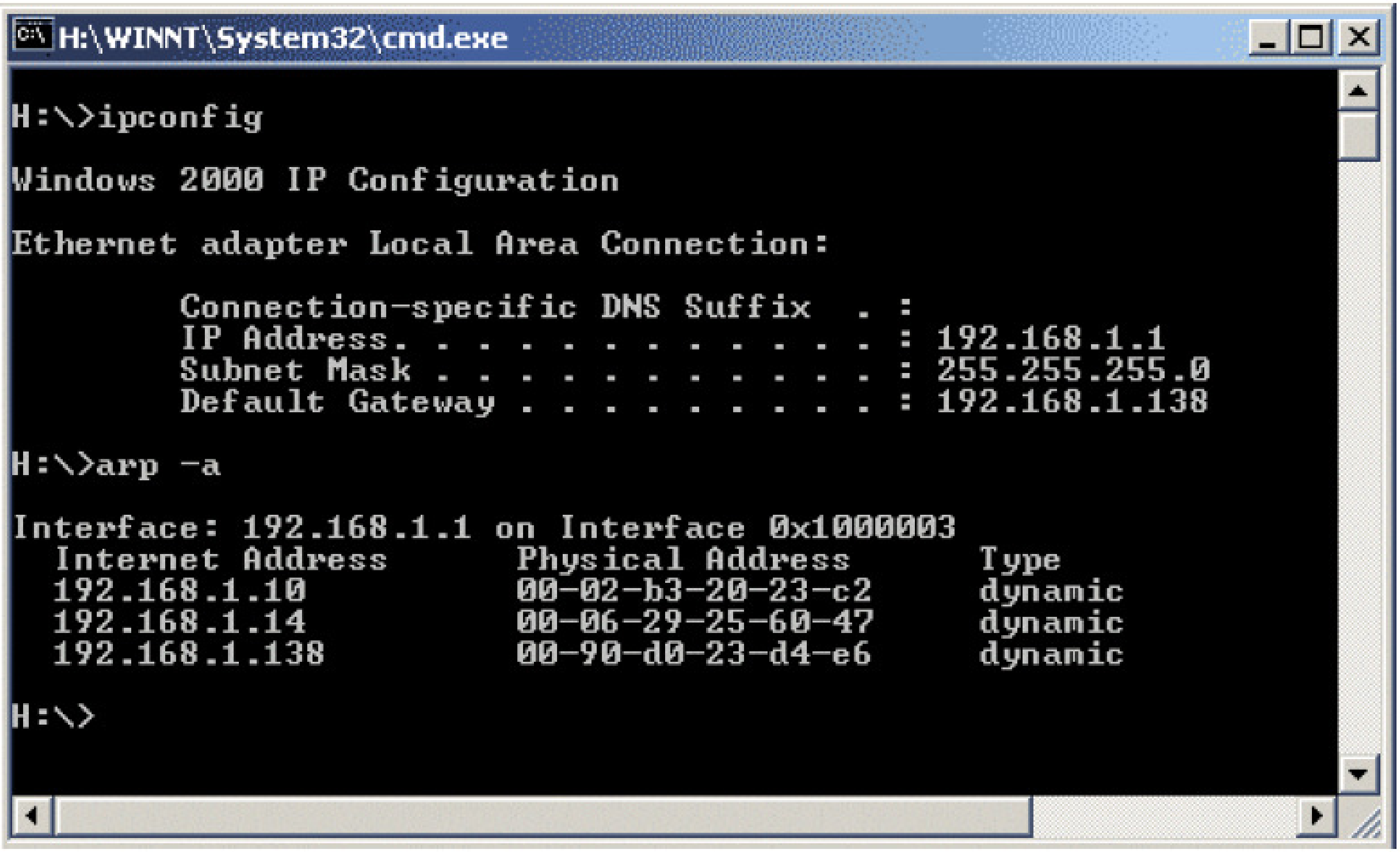
```
C:\Documents and Settings\santosh>arp -d

C:\Documents and Settings\santosh>arp -a
No ARP Entries Found

C:\Documents and Settings\santosh>_
```

Demonstration by Screen Captures



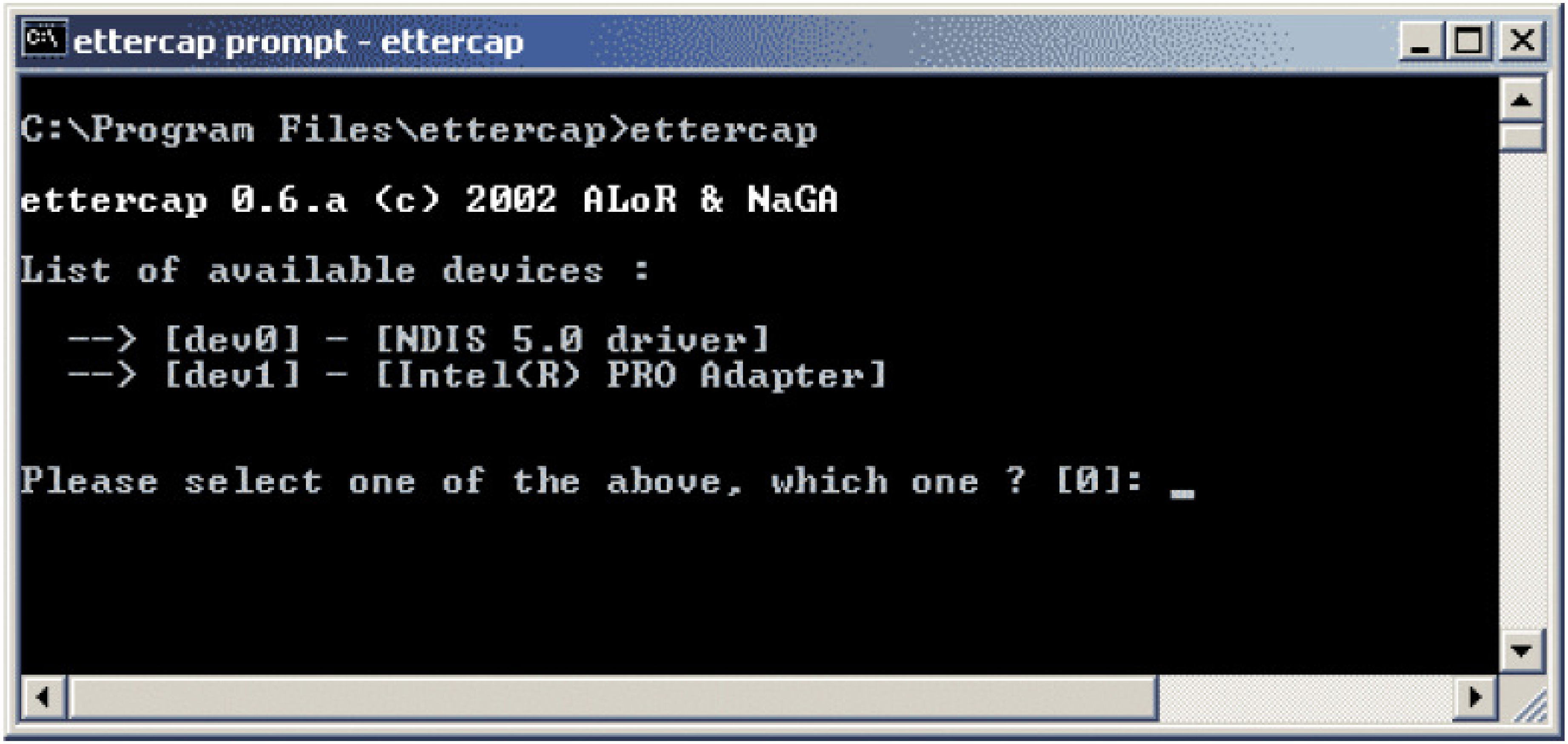
```
C:\Documents and Settings\santosh>ping 202.141.80.15

Pinging 202.141.80.15 with 32 bytes of data:

Reply from 202.141.80.15: bytes=32 time<1ms TTL=64
Reply from 202.141.80.15: bytes=32 time=16ms TTL=64
Reply from 202.141.80.15: bytes=32 time<1ms TTL=64
Reply from 202.141.80.15: bytes=32 time<1ms TTL=64

Ping statistics for 202.141.80.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 16ms, Average = 4ms

C:\Documents and Settings\santosh>arp -a

Interface: 202.141.81.120 --- 0x2
  Internet Address      Physical Address      Type
  202.141.80.15         00-01-f4-38-95-19     dynamic
  202.141.81.5          00-23-8b-41-bf-5d     dynamic
  202.141.81.211        00-1e-0b-ff-64-01     dynamic

C:\Documents and Settings\santosh>_
```

Demonstration by Screen Captures



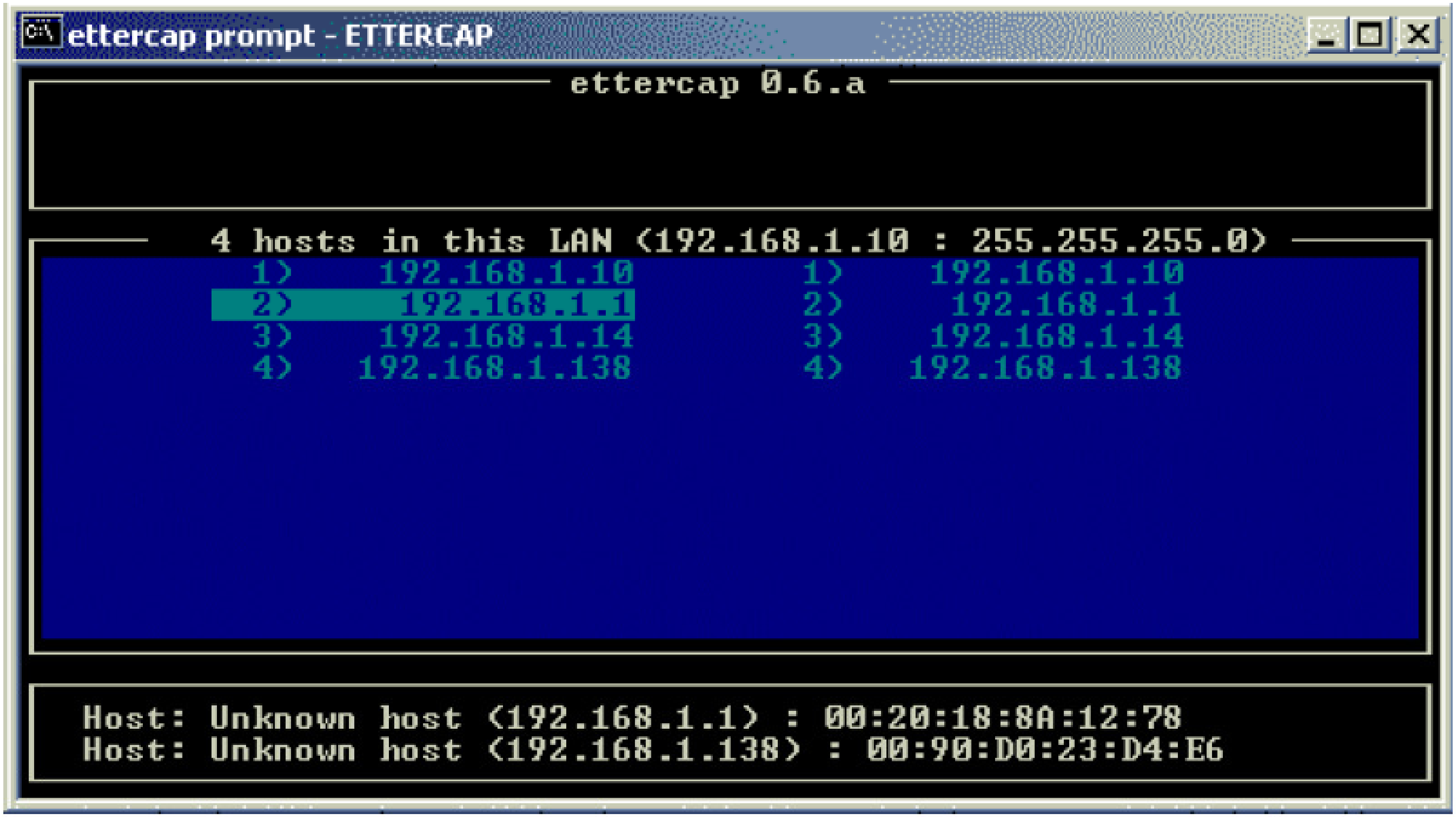
```
H:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.138

H:\>arp -a

Interface: 192.168.1.1 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.1.10          00-02-b3-20-23-c2     dynamic
  192.168.1.14          00-06-29-25-60-47     dynamic
  192.168.1.138         00-90-d0-23-d4-e6     dynamic

H:\>
```

Demonstration by Screen Captures



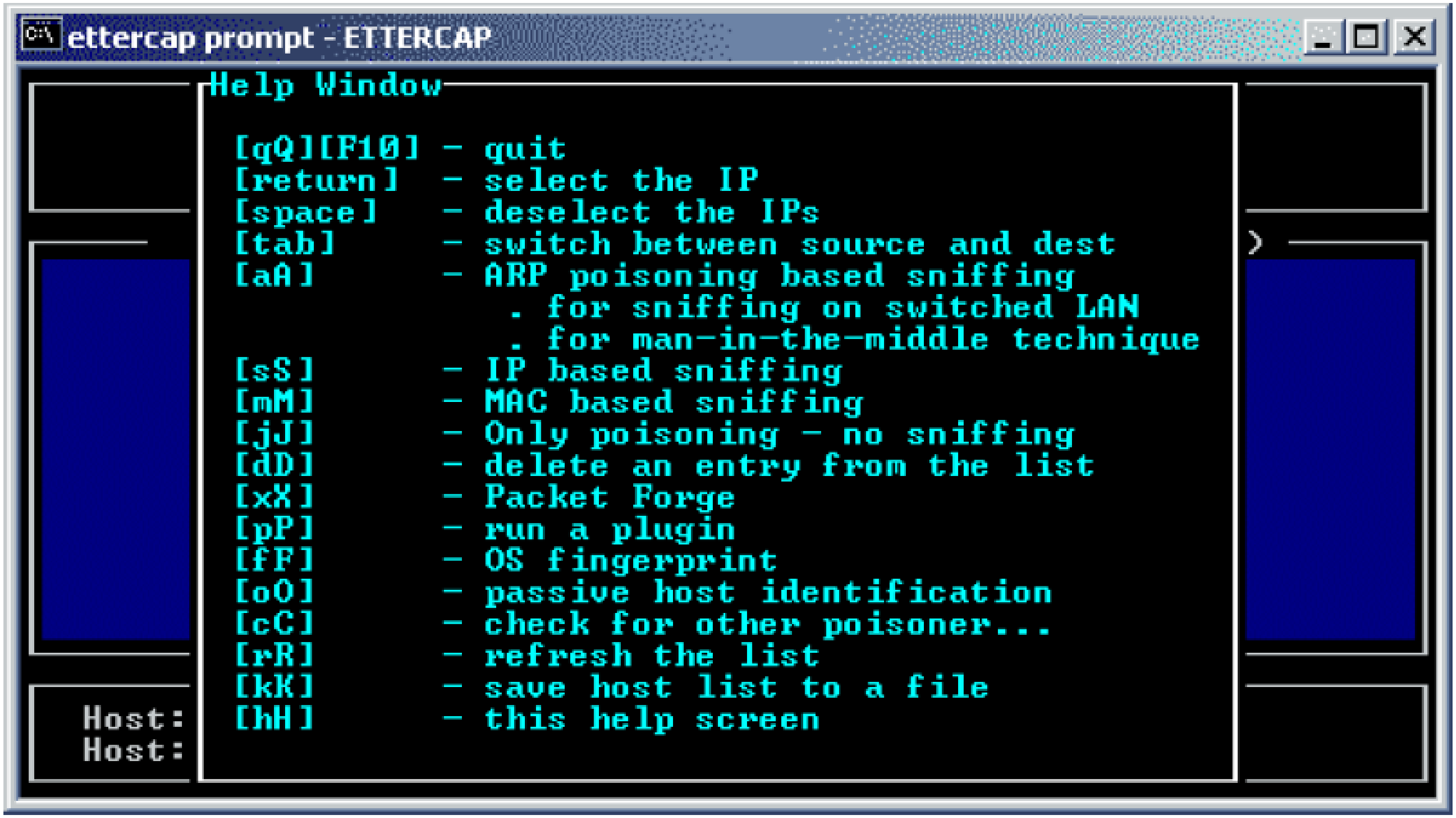
```
C:\Program Files\ettercap>ettercap

ettercap 0.6.a (c) 2002 ALoR & NaGA

List of available devices :

  --> [dev0] - [NDIS 5.0 driver]
  --> [dev1] - [Intel(R) PRO Adapter]


Please select one of the above, which one ? [0]: _
```

Demonstration by Screen Captures



```
ettercap 0.6.a

4 hosts in this LAN (192.168.1.10 : 255.255.255.0)
   1)     192.168.1.10          1)     192.168.1.10
   2)      192.168.1.1          2)      192.168.1.1
   3)     192.168.1.14          3)     192.168.1.14
   4)    192.168.1.138          4)    192.168.1.138

Host: Unknown host (192.168.1.1) : 00:20:18:8A:12:78
Host: Unknown host (192.168.1.138) : 00:90:D0:23:D4:E6
```

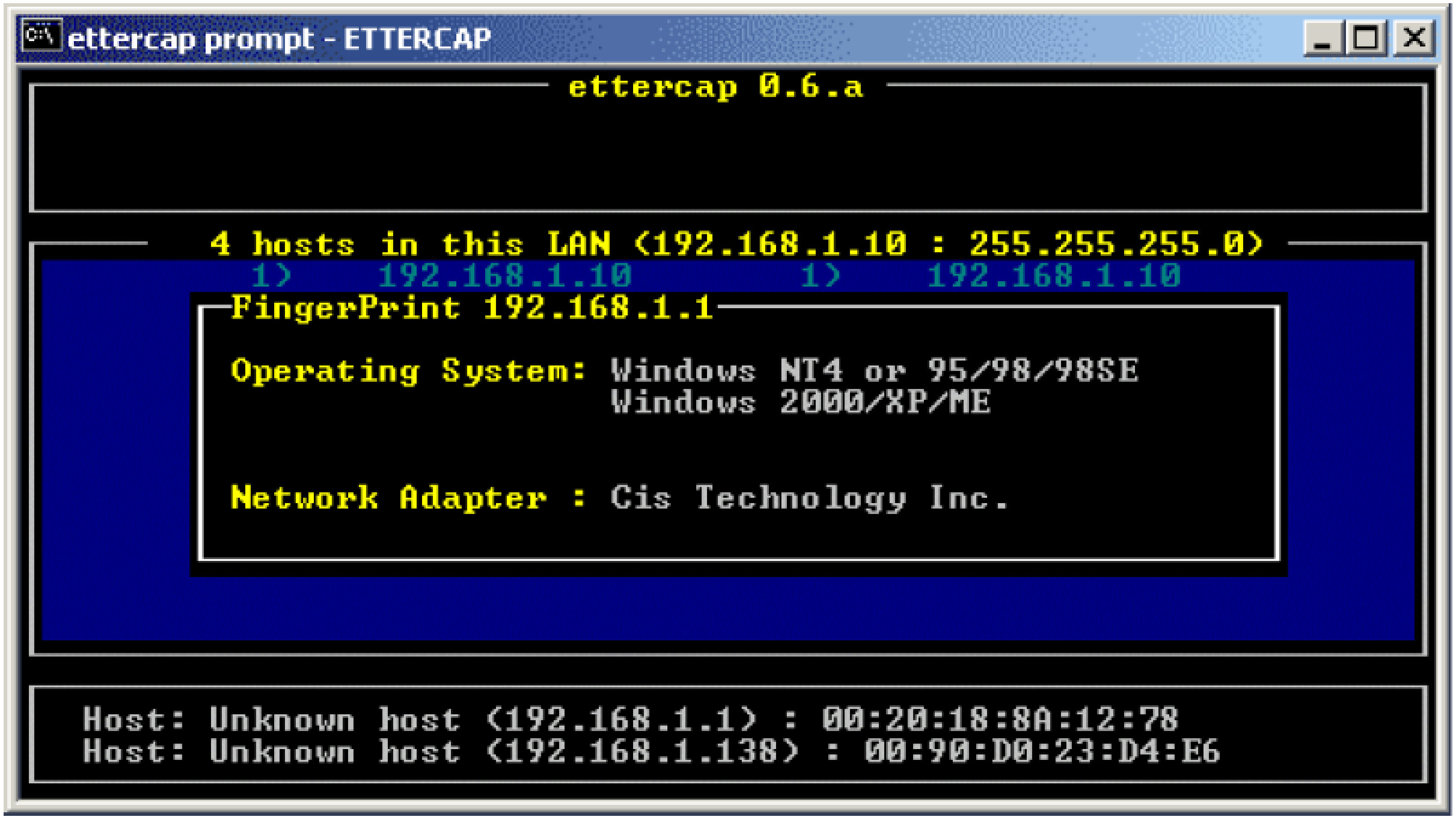
Demonstration by Screen Captures



```
Help Window

  [qQ][F10] - quit
  [return]  - select the IP
  [space]   - deselect the IPs
  [tab]     - switch between source and dest
  [aA]      - ARP poisoning based sniffing
                . for sniffing on switched LAN
                . for man-in-the-middle technique
  [sS]      - IP based sniffing
  [mM]      - MAC based sniffing
  [jJ]      - Only poisoning - no sniffing
  [dD]      - delete an entry from the list
  [xX]      - Packet Forge
  [pP]      - run a plugin
  [fF]      - OS fingerprint
  [oO]      - passive host identification
  [cC]      - check for other poisoner...
  [rR]      - refresh the list
  [kK]      - save host list to a file
  [hH]      - this help screen
```

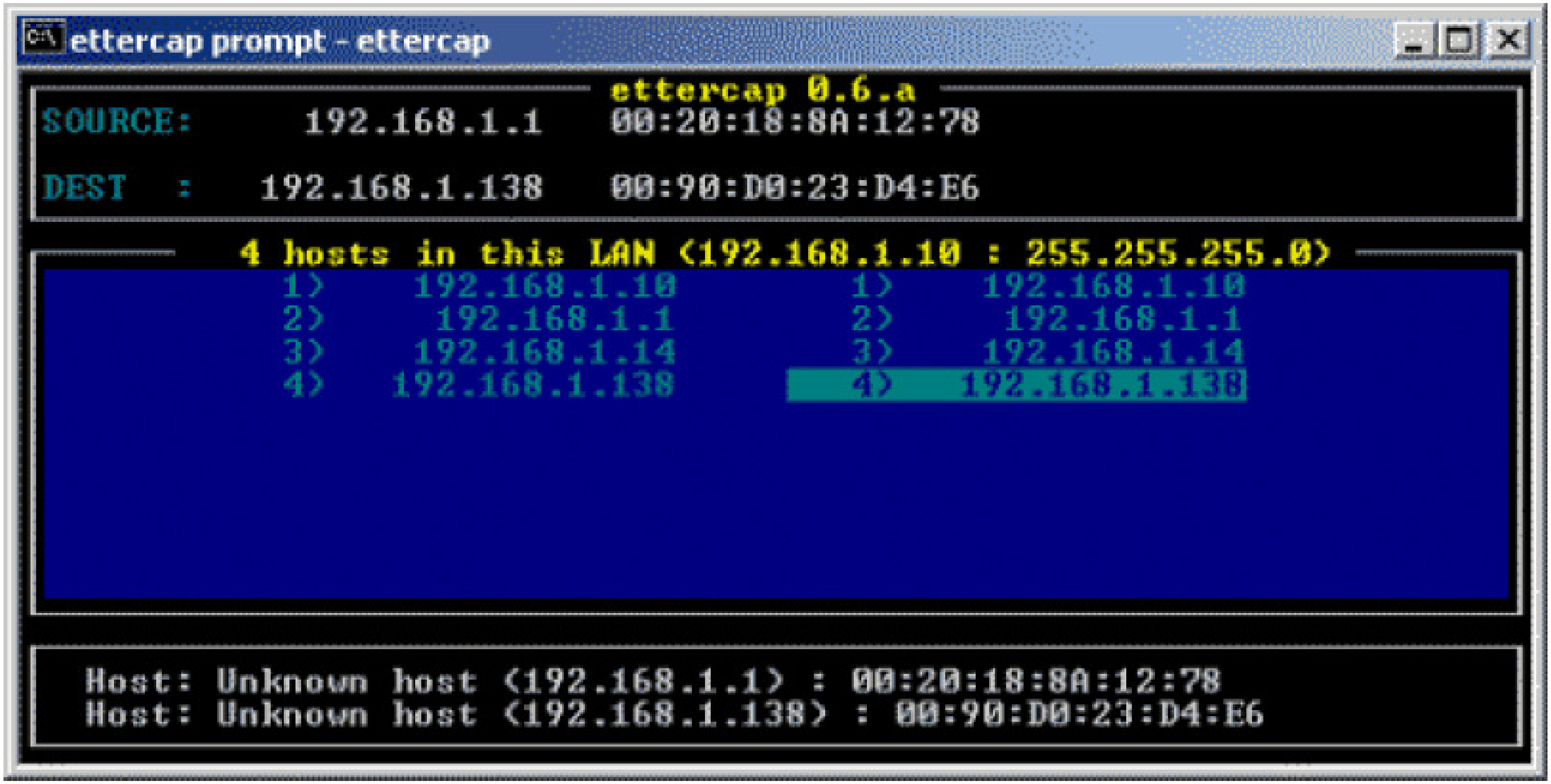
```
Host:
Host:
```

Demonstration by Screen Captures



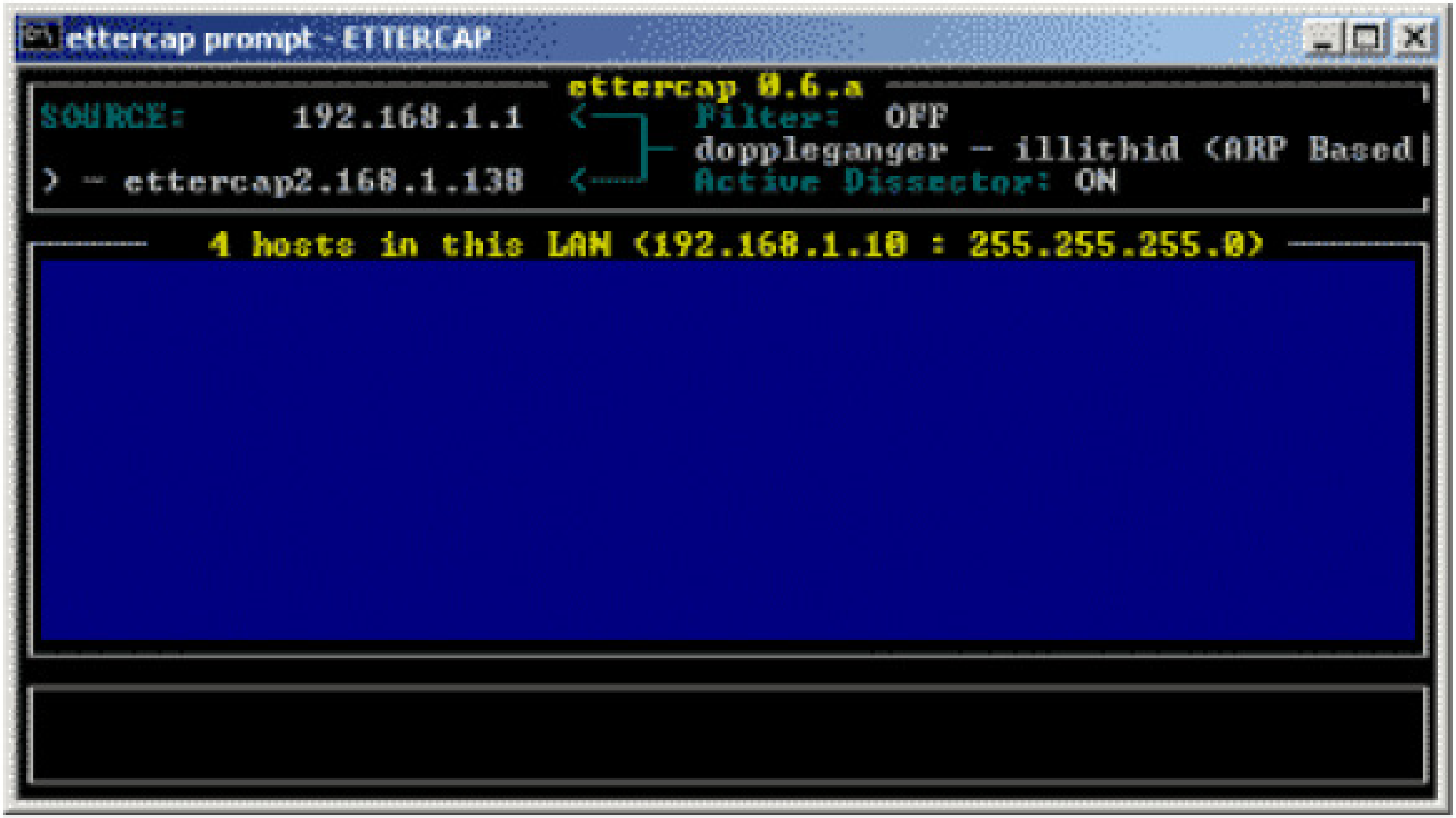
```
                         ettercap 0.6.a

       4 hosts in this LAN (192.168.1.10 : 255.255.255.0)
        1)      192.168.1.10            1)      192.168.1.10
          FingerPrint 192.168.1.1

          Operating System: Windows NT4 or 95/98/98SE
                            Windows 2000/XP/ME


          Network Adapter : Cis Technology Inc.


   Host: Unknown host (192.168.1.1) : 00:20:18:8A:12:78
   Host: Unknown host (192.168.1.138) : 00:90:D0:23:D4:E6
```

Demonstration by Screen Captures



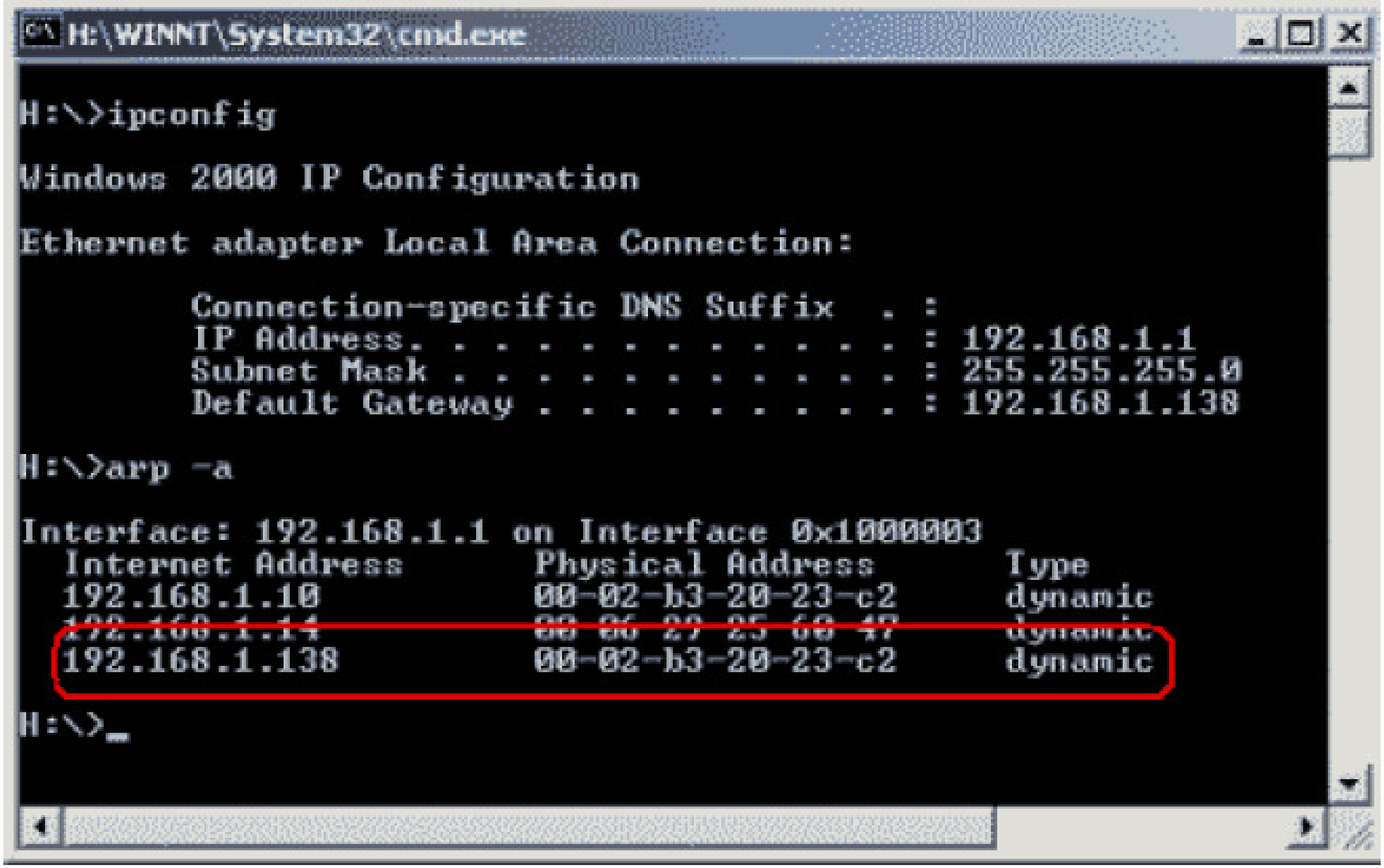
```
                                    ettercap 0.6.a
SOURCE:      192.168.1.1      00:20:18:8A:12:78

DEST   :    192.168.1.138     00:90:D0:23:D4:E6

          4 hosts in this LAN (192.168.1.10 : 255.255.255.0)
          1)     192.168.1.10             1)     192.168.1.10
          2)      192.168.1.1             2)      192.168.1.1
          3)     192.168.1.14             3)     192.168.1.14
          4)    192.168.1.138             4)    192.168.1.138

   Host: Unknown host (192.168.1.1) : 00:20:18:8A:12:78
   Host: Unknown host (192.168.1.138) : 00:90:D0:23:D4:E6
```

Demonstration by Screen Captures



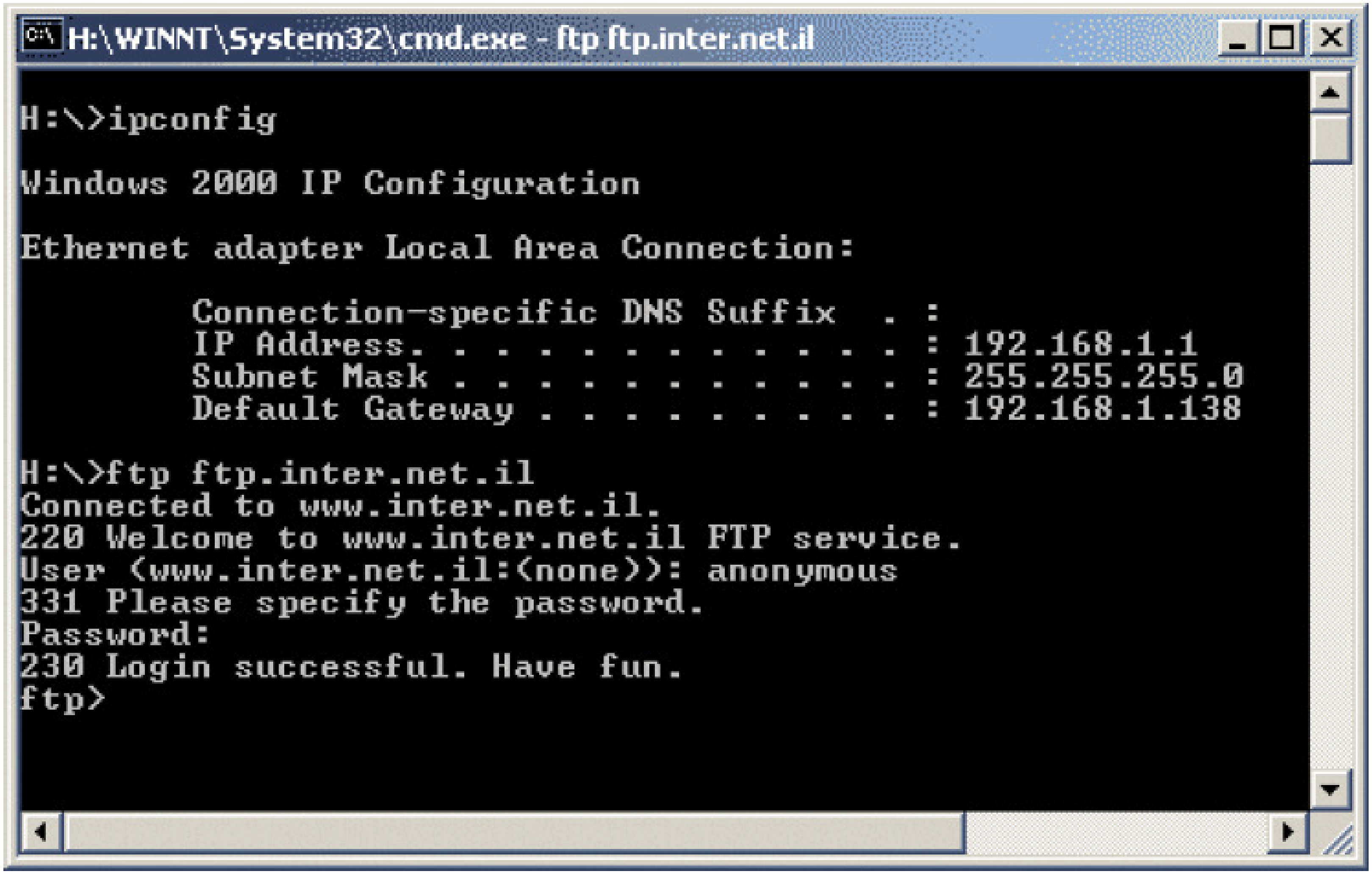
```
                              ettercap 0.6.a
SOURCE:        192.168.1.1    <--┐      Filter:  OFF
                                 ├─  doppleganger - illithid (ARP Based
> - ettercap2.168.1.138       <--┘      Active Dissector: ON

         4 hosts in this LAN (192.168.1.10 : 255.255.255.0)
```

Demonstration by Screen Captures



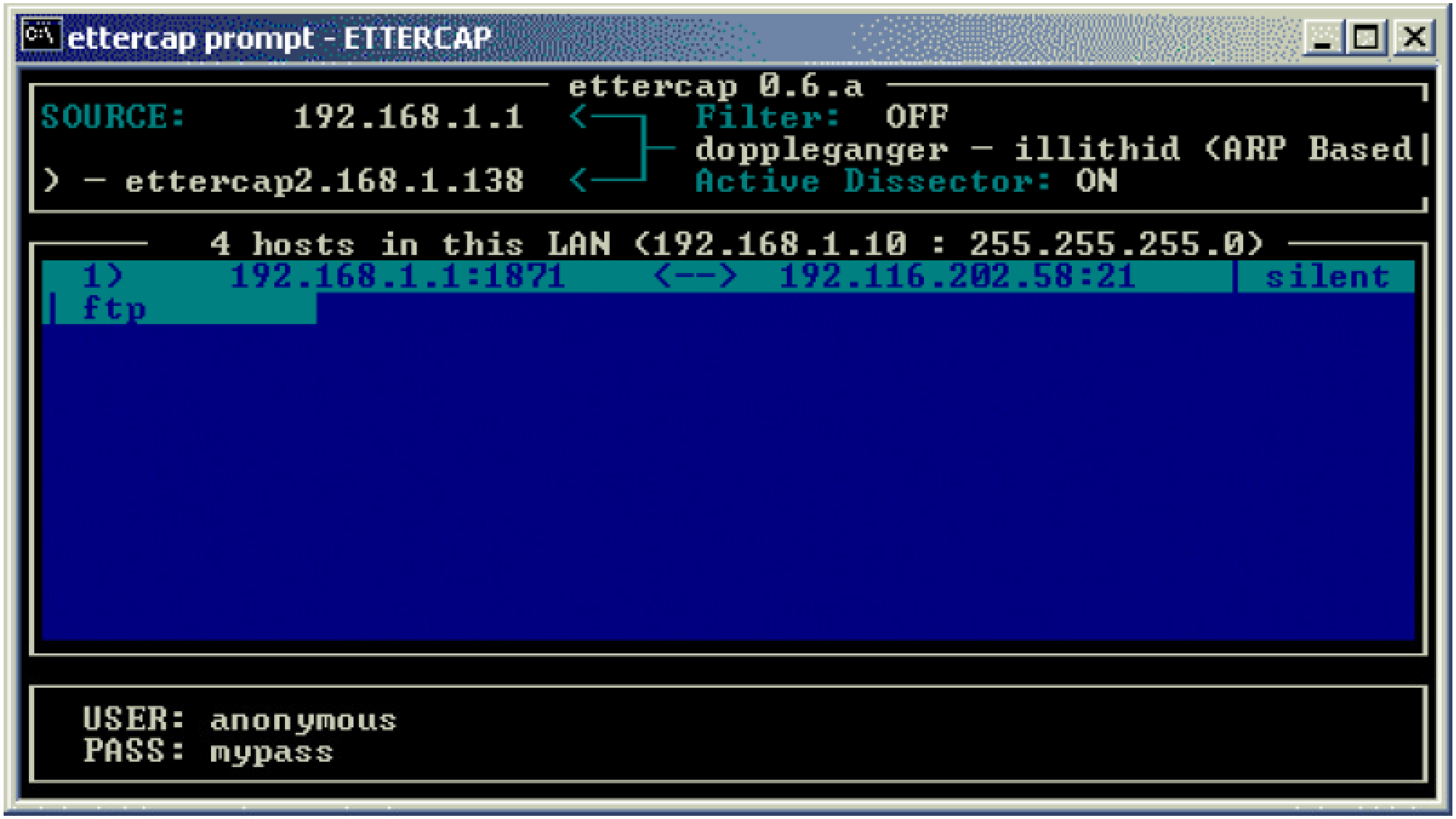
```
H:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.138

H:\>arp -a

Interface: 192.168.1.1 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.1.10          00-02-b3-20-23-c2     dynamic
  192.168.1.14          00-06-29-25-60-47     dynamic
  192.168.1.138         00-02-b3-20-23-c2     dynamic

H:\>_
```

Demonstration by Screen Captures



```
H:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.138

H:\>ftp ftp.inter.net.il
Connected to www.inter.net.il.
220 Welcome to www.inter.net.il FTP service.
User (www.inter.net.il:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful. Have fun.
ftp>
```

Demonstration by Screen Captures



```
                              ettercap 0.6.a
SOURCE:      192.168.1.1    <──┐   Filter:   OFF
                               ├─ doppleganger - illithid (ARP Based|
> - ettercap2.168.1.138     <──┘   Active Dissector: ON

      4 hosts in this LAN (192.168.1.10 : 255.255.255.0)
  1)      192.168.1.1:1871      <-->  192.116.202.58:21      | silent
| ftp

  USER:  anonymous
  PASS:  mypass
```

References

- [ARD] *https://www.arpdefender.com.*
- [ARG] *https://www.arp-guard.com.*
- [Bar03] M. Barnaba. Anticap. 2003. *http://cvs.antifork.org/cvsweb.cgi/anticap.*
- [BOR03] D. Bruschi, A. Ornaghi, and E. Rosti. S-arp: a secure address resolution protocol. In *ACSAC '03: Proceedings of the 19th Annual Computer Security Applications Conference,* page 66,, 2003. IEEE Computer Society.
- [CIS] *http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/swcg.html.*
- [Hun] F. Hunleth. Secure link layer. h*ttp://www.cs.wustl.edu/fifhunleth/projects/projects.html.*
- [Lab] Lawrence Berkeley National Laboratory. Arpwatch tool: Arp spoofing detector.
- *ftp://ftp.ee.lbl.gov/ARPwatch.tar.gz.*
- [LEM05] Wesam Lootah, William Enck, and Patrick McDaniel. Tarp: Ticket-based address resolution protocol. *Computer Security Applications Conference, 2005.*
- [Plu82] David C. Plummer. An ethernet address resolution protocol. 1982. RFC826.
- [RN05] Vivek Ramachandran and Sukumar Nandi. Detecting arp spoofing: An actuve technique. *Computer Security Applications Conference, Annual, 0:116-126, 2005.*

References

- [SM06] Khamphao Sisaat and Daisuke Miyamoto. Source address validation support for network forensics. In *The 1st Joint Workshop on Information security,* 2006.
- [Tet03] I. Teterin. Antidote. 2003. *http://online.securityfocus.com/archive/1/299929*.
- C. G. Cassandras and S. Lafortune, Introduction to discrete event systems, Kluwer Academic Publishers, 1 edition, 1999.
- R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," in Symposium on Security and Privacy. 2001, pp. 144–155, IEEE.
- S.-J. Whittaker, M. Zulkernine, and K. Rudie, "Towards incorporating discrete-event systems in secure software development," in International Conference on Availability, Reliability and Security. 2008, pp. 1188–1195, IEEE.
- R. Alur and D. L. Dill, "A theory of timed automata," in Theoretical Computer Science. 1993, pp. 183–235, Elsevier.
- K-T. Cheng and A.S. Krishnakumar, "Automatic functional test generation using the extended finite state machine model," in International Design Automation Conference. 1993, pp. 86–91, IEEE-ACM.

THANK YOU