

---

# Ethereal & the Art of Debugging Networks

Gerald (Jerry) Carter  
SAMBA Team  
Centeris

<http://www.plainjoe.org/>  
[jerry@samba.org](mailto:jerry@samba.org)  
<http://www.centeris.com/>



Copyright Gerald Carter, 2005-2006. All rights reserved  
[jerry@samba.org](mailto:jerry@samba.org), Slide 1

## Basic Assumption

---

- We will focus on TCP/IP and Application layer protocols
- We will focus on IPv4
- You are comfortable with TCP/IP host configuration & hardware
- Only dealing with software based network sniffers
- Will be covering debugging techniques based on real examples
  - will not cover every possible network error, bug, or glitch



Copyright Gerald Carter, 2005-2006. All rights reserved  
[jerry@samba.org](mailto:jerry@samba.org), Slide 2

# Justification

---

- Software debugging techniques
  - printf()
  - symbolic debuggers (e.g. gdb)
- Network application debugging techniques
  - log files
  - raw packets & network protocol analyzers
- Sometimes you don't have access to the application logs or they are just wrong
- Don't overlook bad hardware, device drivers, or application bugs
- No magic bullet, just another tool in the utility belt

# References

---

- TCP/IP Illustrated, Volume 1: The Protocols, R. Stevens, 1993, Addison-Wesley.
- Interconnections: Bridges, Routers, Switches, and Internetworking Protocols (2<sup>nd</sup> edition), R. Perlman, 1999, Addison-Wesley.
- Ethereal Packet Sniffing, A. Orebaugh, 2004, Syngress.
- Linux Firewalls (3<sup>rd</sup> edition), R. Ziegler, 2005, Novell Press.

# Outline

---

- Ethereal
  - Overview and Installation
  - Captures & filters
- TCP/IP Networking
- Application Protocols

# Packet Sniffers

---

- Tools which enable record, monitor, or analyze network traffic
- Network Monitoring Tools
  - <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>
- Software based
  - tcpdump, ethereal, snoop, etc...
- Hardware based

# Ethereal

---

- <http://www.ethereal.com/>
- Available for Unix, Windows, and OS X
- Components
  - ❑ *ethereal* – gtk+ based GUI
  - ❑ *tethereal* – command line version of GUI
  - ❑ *editcap*, *mergcap*, *text2pcap*, *capinfos* – command line tools for manipulating capture files
- Many supported file formats including pcap, MS' netmon, & Solaris' snoop

# Dependencies

---

- Required
  - ❑ glib 2.0, gtk+ 2.0
- Optional
  - ❑ GNU ADNS library
  - ❑ Perl Compatible Regular Expression library
  - ❑ Zlib
  - ❑ Net-SNMP libs
  - ❑ Kerberos and OpenSSL

# Building....

```
$ ./configure --prefix=/opt/ethereal --with-krb5=/usr --with-ssl
The Ethereal package has been configured with the following
options.
```

```
Build ethereal : yes
Build tethereal : yes
.....
Install setuid : no
Use plugins : yes
Use GTK+ v2 library : yes
Use threads : no
Build profile binaries : no
Use pcap library : yes
Use zlib library : yes
Use pcre library : yes
Use kerberos library : yes (MIT)
Use GNU ADNS library : no
Use SSL crypto library : yes
Use IPv6 name resolution : yes
Use UCD SNMP/Net-SNMP library : no
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 9

# Screenshot

*\$ ethereal*

capture

decoded packet

raw data

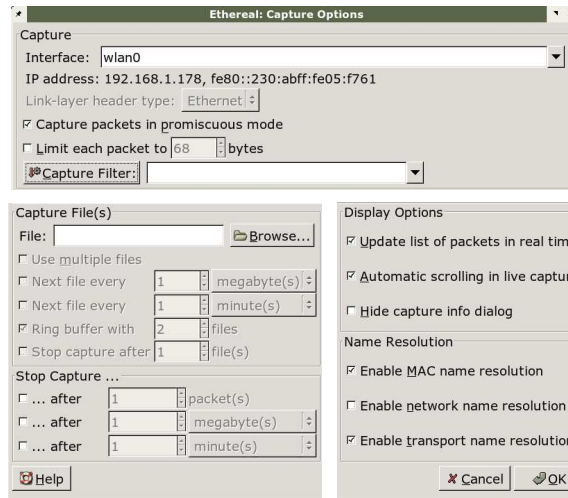
No.	Time	Source	Destination	Protocol	Info
5	7.170828	Vmware_Sb:bf:ef	Broadcast	ARP	Who has 192.168.1.144? Tell 192.168.1.101
6	8.431791	192.168.1.178	130.239.18.172	IRC	Request
7	8.622197	130.239.18.172	192.168.1.178	TCP	6667 > 33147 [ACK] Seq=0 Ack=20 Win=1448 Len=0 TSV=614344665 TSER=1
8	8.657783	130.239.18.172	192.168.1.178	IRC	Response
9	8.657871	192.168.1.178	130.239.18.172	TCP	33147 > 6667 [ACK] Seq=20 Ack=62 Win=16022 Len=0 TSV=19090432 TSER=1
10	9.678331	192.168.1.178	66.70.73.150	TCP	38002 > http [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=19091451
11	9.724711	66.70.73.150	192.168.1.178	TCP	http > 38002 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=321
12	9.724845	192.168.1.178	66.70.73.150	TCP	38002 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=19091499 TSER=321



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 10

# Capture Dialog

*CTRL+K*



interface,  
capture filter,  
snap length,  
etc...

ring  
buffer

DNS lookups,  
automatic  
display, etc...

# Interfaces

- Any configured network interface is available
  - ❑ `ifconfig -a`
  - ❑ `tethereal -D`
  - ❑ `netstat -n -i`
- *any*
  - ❑ view packets on all interfaces
- packet length (a.k.a. snap length)
  - ❑ number of bytes of the packet that should be stored
  - ❑ generally limit this based on the max transmit unit of the link layer
  - ❑ ethernet MTU is 1500 bytes of data + 14 bytes of frame header

# Ring Buffers

---

- A ring buffer is a set of N files of max size Z
  - ❑ Ethereal will write to the first file until a defined condition is met and then move onto the next
  - ❑ When the last file is full, ethereal rolls over to the first file
- End of file conditions
  - ❑ size
  - ❑ time
- Multiple files can be merged (*mergcap*) and then filtered from a command line using tethereal

# Ethereal & Filters

---

- Two types of filters
  - ❑ capture filters
  - ❑ display filters
- Capture filters use the libpcap filter facility
  - ❑ `man tcpdump(1)`
  - ❑ e.g. “port 80 and host 192.168.1.100”
- Display filters are built upon the ethereal protocol dissectors
  - ❑ supports referencing protocol components by name
  - ❑ e.g. “http && ip.addr == 192.168.1.100”

# Capture Filters

---

- expression
  - ❑ type (*host, net, port*)
  - ❑ direction (*src, dst*)
  - ❑ protocol (*ether, arp, tcp, ...*)
- logical operators
  - ❑ not (!)
  - ❑ and (&&)
  - ❑ or (||)
- Name resolution supported by DNS, /etc/services, /etc/networks, etc...
- Enclose capture expression in quotes if using any special shell characters (e.g. > or !)

# Capture Filter Examples

---

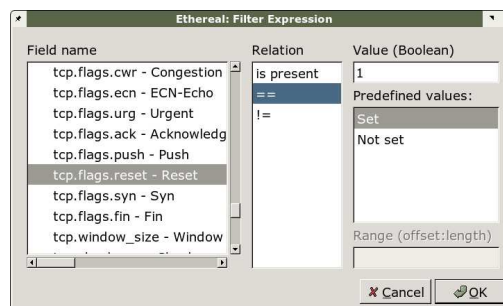
- Capture all packets from the machine at 192.168.1.1
  - ❑ “src host 192.168.1.1”
- Capture all Address Resolution Packets
  - ❑ “arp”
- Capture all packets to and from the host worm.plainjoe.org where the source or destination port is 80
  - ❑ “tcp port 80 and host worm.plainjoe.org”

# Extracting Bytes

- Certain protocols can be referenced as arrays
  - ❑ value returned in network byte order (big endian)
  - ❑ tcp[0] means the first byte in the tcp packet
  - ❑ tcp[0:2] returns a 2 byte value
  - ❑ tcp[0:4] returns a 4 byte value
- Bitwise operators & and | are supported
- Basic set of C relational operators
  - ❑ < <= > >= == !=
- Example: capture ping packets
  - ❑ icmp[0] == 0x8 or icmp[0] == 0x0
  - ❑ icmp[icmptype] == icmp-echo or icmp[icmptype] == icmp-echoreply

# Display Filters

- Display filters allow comparisons based on most fields on the protocol dissector
  - ❑ Help -> Supported Protocols
- Used to filter existing captures, define color rules, or filter statistics
- Can be entered directly or built via a dialog



show all tcp reset packets

# Display Filter Operators

---

<u>Operator</u>	<u>Description</u>
> or gt	greater than
> or ge	than or equal to
< or lt	less than
<= or le	less than or equal to
== or eq	equal
!= or ne	not equal
contains	string (or byte) search
matches	regular expression match (pcre)

# Display Filter Data Types

---

- Numerical
  - int, float, ....
- Strings
  - characters, bytes, ....
- Addresses
  - network, hardware, ....
- Time
  - absolute time and relative time between packets
- Protocol keywords

# Examples

---

- `spoolss && dcerpc.opnum == 0x45`
  - `Win32 OpenPrinterEx()`
- `ip.addr == 192.168.1.1`
  - `ip.src == 192.168.1.1 || ip.dst == 192.168.1.1`
- `ip.flags.df`
  - IP packets with the “Don't Fragment” bit set
- `http contains “POST”`
  - search for the string “POST” in all http packets

# Display Filter Ranges

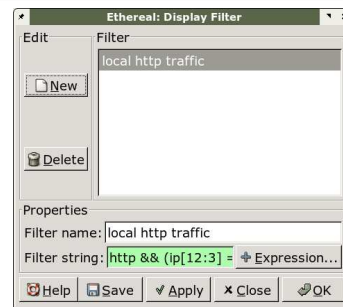
---

- Ranges allow for using slices of byte sequences in filter expressions

<u>Range</u>	<u>Description</u>
<code>[offset]</code>	single byte at <i>offset</i>
<code>[offset:N]</code>	N bytes starting at optional <i>offset</i>
<code>[offset<sub>1</sub>-offset<sub>2</sub>]</code>	bytes from <i>offset<sub>1</sub></i> to <i>offset<sub>2</sub></i> inclusive
<code>[offset:]</code>	bytes in field starting at <i>offset</i>
<code>[range,range]</code>	combine multiple ranges
- Find all HTTP packets on 192.168.1.0/24
  - `http && (ip[12:3] == c0:a8:01 && ip[16:3] == c0:a8:01)`

## Saving Filters

- Both capture and display filters can be saved in the user preferences
- Stored in `~/.ethereal/`
  - ❑ `cfilters`
  - ❑ `dfilters`
  - ❑ `colorfilters`



## Command line captures

- `tethereal`
  - ❑ `-h` (help text)
  - ❑ `-w outfile` (write to file)
  - ❑ `-s snaplen` (capture *snaplen* bytes of packet)
  - ❑ `-i interface` (name of NIC to watch)
  - ❑ `-n` (disable name resolution)
  - ❑ `-b` (multiple file mode)
    - ✓ `files:numfiles` (number of files in ring buffer)
    - ✓ `filesize:Kb` (swap to next file after N Kb)
    - ✓ `duration:seconds` (swap to next file after M seconds)
  - ❑ `-a` (criterion to move to next file; same options as `-b`)
  - ❑ `-f "capture_filter"`
    - ✓ can also pass mult-expression filters as last arguments

## Multiple Capture Files

---

- Multiple files (until disk is full)
  - ❑ `tethereal -b -a {duration,filesize,files}:value -w x.pcap`
- Ring buffer
  - ❑ `tethereal -b files:num -a test:value -w x.pcap`
- Merging and filtering files
  - ❑ `mergcap -w newfile.pcap x*.pcap`
  - ❑ `tethereal -r newfile.pcap -R dfilter -w filtered.pcap`

## Popular CLI Sniffers

---

- Operating systems often come with a command line packet capture tool installed
  - ❑ frequently easier to use an installed tool than deploying a new package
  - ❑ no ring buffer support in general
- Solaris' snoop
  - ❑ `snoop -r -o /tmp/dump.snoop -d hme0 cfilter`
- tcpdump
  - ❑ similar arguments as tethereal
  - ❑ `tcpdump -w /tmp/dump.pcap -s 0 -i eth0 cfilter`

---

This slide intentionally left blank.

--anonymous



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 27

---

This slide intentionally left blank.

--anonymous



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 28

---

# TCP/IP Protocol

---

## OSI vs. TCP/IP

### OSI

Application
Presentation
Session
Transport
Network
Data Link
Physical

### TCP/IP

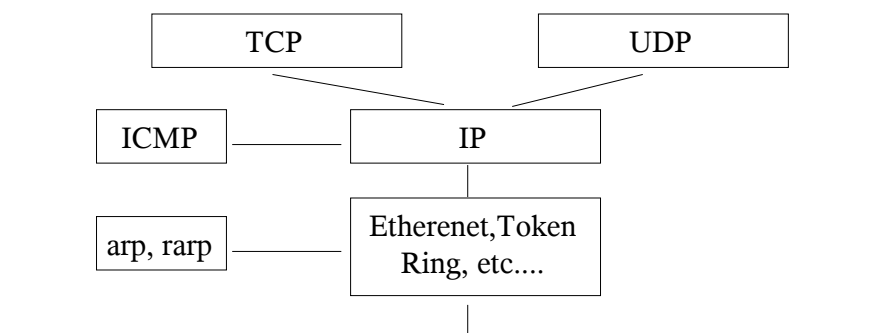
Application
Transport
Network
Link

# TCP/IP Protocol Suite

---

Applications (Samba, nfsd, Apache, Postfix, BGP ,...)

---



## Address Resolution Protocol

---

- Arp (“What's the ethernet address for aa.bb.cc.dd?”)
  - Matches network addresses to link layer addresses using the link broadcast address
- Reverse Arp (“What's the IP address for this ethernet address?”)
  - commonly used for diskless booting
- Gratuitous Arp (“Who has aa.bb.cc.dd? Tell aa.bb.cc.dd.”)
  - Frequently used by hosts when booting to detect a duplicate IP address on the network
- Proxy Arp
  - Used by routers to answer Arp requests for remote hosts

# Failed Ping

```
rain$ ping 192.168.1.1 2>&1 > /tmp/ping.log &
```

```
rain$ tethereal -i wlan0 \  
ether host 00:30:AB:05:F7:61 and arp  
Capturing on wlan0  
0.000000 00:30:ab:05:f7:61 -> ff:ff:ff:ff:ff:ff  
      ARP Who has 192.168.1.1? Tell 192.168.1.178  
0.999845 00:30:ab:05:f7:61 -> ff:ff:ff:ff:ff:ff  
      ARP Who has 192.168.1.1? Tell 192.168.1.178  
.....
```

```
snow$ tethereal -n -i eth0 \  
ether host 00:30:AB:05:F7:61 and arp  
Capturing on eth0  
51.655889 00:30:ab:05:f7:61 -> ff:ff:ff:ff:ff:ff  
      ARP Who has 192.168.1.1? Tell 192.168.1.178  
51.655958 00:00:f4:d8:37:d0 -> 00:30:ab:05:f7:61  
      ARP 192.168.1.1 is at 00:00:f4:d8:37:d0  
.....
```

# Duplicate IP

```
rain$ # tethereal -i wlan0 \  
ether host 00:30:AB:05:F7:61 and arp  
Capturing on wlan0  
1.514118 00:30:ab:05:f7:61 -> ff:ff:ff:ff:ff:ff  
      ARP Who has 192.168.1.71? Tell 192.168.1.178  
1.516633 00:0c:29:a9:b5:2f -> 00:30:ab:05:f7:61  
      ARP 192.168.1.71 is at 00:0c:29:a9:b5:2f  
1.517694 00:0c:29:63:18:bd -> 00:30:ab:05:f7:61  
      ARP 192.168.1.71 is at 00:0c:29:63:18:bd  
.....
```

# IP Datagrams

- Delivered host to host
- Destination host kernel demultiplexes to TCP/IP applications based on port numbers specified in the transport layer

0 ← - - - - - ▶ 15 16 ← - - - - - ▶ 31

4-bit version	4-bit hdr len	TOS	total length	
identification			3-bit flags	fragment offset
TTL	protocol		checksum	
source IP address				
destination IP address				

IPv4 Header



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 35

# UDP

- Connectionless, no ACK, retransmits up to application
- Single packet payload
  - theoretically limited by 16-bit length
  - generally  $\leq 8200$  bytes (8192 + 8 byte header)

0 ← - - - - - ▶ 15 16 ← - - - - - ▶ 31

source port	destination port
length	checksum

UDP Header



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 36

# Unserviced UDP Ports

---

```
$ host foo 192.168.1.1
;; connection timed out; no servers could be reached
```

```
$ tethereal -p -i wlan0 arp or icmp or port 53
Capturing on wlan0
0.000000 192.168.1.178 -> 192.168.1.1
          DNS Standard query A foo.plainjoe.org
0.002487 192.168.1.1 -> 192.168.1.178
          ICMP Destination unreachable (Port unreachable)
5.001193 192.168.1.178 -> 192.168.1.1
          DNS Standard query A foo.plainjoe.org
5.003729 192.168.1.1 -> 192.168.1.178
          ICMP Destination unreachable (Port unreachable)
```

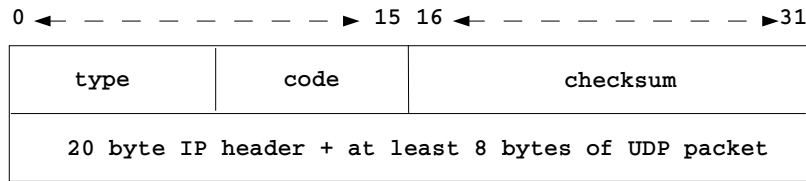
# Internet Control Message Protocol

---

- ICMP is used to communicate
  - ❑ error message (e.g. host unreachable)
  - ❑ status information (e.g. ping)
  - ❑ network information (e.g. redirect for network)
- Carried in IP packets

# ICMP: Port Unreachable

- Returned when a UDP packet is sent to a port which has no listening process



ICMP: Port Unreachable

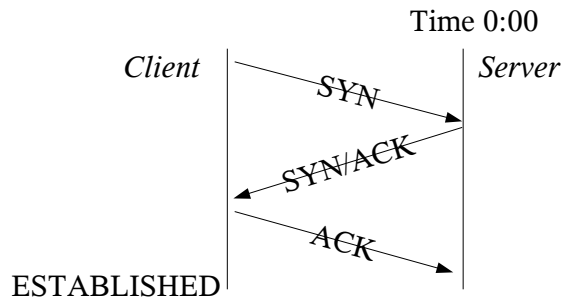
# Traceroute

- Use TTL-exceeded ICMP error to map path from source to destination
- Starts with a UDP packet using a TTL of 1 and increments monotonically
- Router returns time-to-live exceeded ICMP message when the TTL (after decrementing) reaches 0
  - TTL of 1 should map 1<sup>st</sup> hop, TTL of 2 should map 2<sup>nd</sup> hop, etc...
- Linux traceroute sends out 3 UDP probes per hop to log round trip time (RTT)
- Beware of firewalls



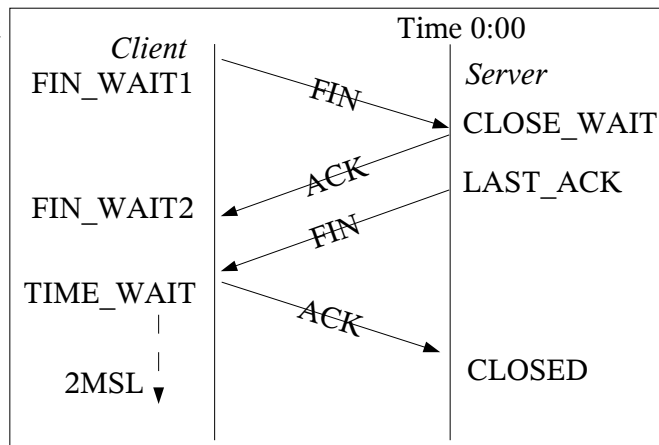
# TCP 3-way Handshake

- Client sends SYN packet with initial sequence number
- Server responds with a SYN/ACK
  - ACK number is ISN+1
- Client responds with ACK



# 4 Way Close

- FIN bit is for normal TCP session termination
- TIME\_WAIT lasts for 2 x MSL
  - max segment lifetime



# netstat

```
$ # netstat -n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.1:35400        127.0.0.1:4993        CLOSE_WAIT
tcp      0      0 192.168.1.178:35221    66.70.73.150:22       ESTABLISHED
tcp      0      0 127.0.0.1:4993        127.0.0.1:35399       FIN_WAIT2
tcp      0      0 127.0.0.1:4993        127.0.0.1:35401       FIN_WAIT2
tcp      0      0 127.0.0.1:4993        127.0.0.1:35400       FIN_WAIT2
tcp      0      0 192.168.1.178:39701    192.168.1.56:143     ESTABLISHED
tcp      0      0 192.168.1.178:34259    192.168.1.56:143     CLOSE_WAIT
tcp      0      0 192.168.1.178:34260    192.168.1.56:143     CLOSE_WAIT
tcp      0      0 192.168.1.178:36959    192.168.1.84:22      TIME_WAIT
tcp      0      0 192.168.1.178:38788    192.168.1.84:22      ESTABLISHED
tcp      0      0 127.0.0.1:6010        127.0.0.1:35979       ESTABLISHED
tcp      0      0 127.0.0.1:49677       127.0.0.1:631        CLOSE_WAIT
tcp      0      0 192.168.1.178:46140    192.168.1.84:22      ESTABLISHED
tcp      0      0 192.168.1.178:51979    192.168.1.44:22      ESTABLISHED
tcp      896    0 127.0.0.1:59245        127.0.0.1:22         ESTABLISHED
tcp      0      0 127.0.0.1:35979       127.0.0.1:6010       ESTABLISHED
....
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 45

## Unserviced TCP Ports

```
$ telnet 192.168.1.1 53
Trying 192.168.1.1...
telnet: connect to address 192.168.1.1: Connection refused
```

```
$ tethereal -p -n -i wlan0 tcp and port 53
Capturing on wlan0
7.787969 192.168.1.178 -> 192.168.1.1  TCP 53329 > 53 [SYN]
      Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=31781951
      TSER=0 WS=2
7.790345 192.168.1.1 -> 192.168.1.178  TCP 53 > 53329 [RST, ACK]
      Seq=0 Ack=0 Win=0 Len=0
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 46

# TCP SYN Tricks

---

- SYN Scan
  - ❑ send SYN packets looking for open ports
  - ❑ e.g.
    - ✓ “closed” ports return [RST,ACK]
    - ✓ “open” ports return [SYN,ACK]
    - ✓ “filtered” ports are protected by a firewall
- SYN Attacks
  - ❑ Client sends a SYN but does not send the final ACK leaving a half open connection
  - ❑ consumes resources in the server's connection queue
  - ❑ syn cookies use a special ISN to protect against
    - ✓ <http://cr.yip.to/syncookies.html>

---

This slide intentionally left blank.

--anonymous

---

# Application Protocols

---

## Domain Name Service

- Utilizes both UDP and TCP port 53
- Normal queries are via UDP
- Client should retry using TCP when the UDP reply contains the “truncated” bit
- Already seen on instance of DNS timeouts caused by misconfigured `/etc/resolv.conf`
  - ❑ client resolver continued to send name query even when the UDP packet elicited a “port unreachable” ICMP message

# Scenario

```
$ grep software /etc/auto.misc
software      -rw,hard,intr  worm:/export/u2/software
$ cd /misc/software
cd: /misc/software: No such file or directory
```

```
Oct 29 14:39:16 rain automount[6315]:
    attempting to mount entry /misc/software
Oct 29 14:39:16 rain automount[13018]:
    mount(nfs): entry software: host worm: lookup failure
```

```
# tethereal -n -i wlan0 icmp or port 53
Capturing on wlan0
0.000000 192.168.1.178 -> 192.168.1.56
        DNS Standard query A worm.painjoe.org
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 51

# Scenario

```
$ ping sleet.everest.painjoe.org
ping: unknown host sleet.everest.painjoe.org
```

```
; zone file for everest.painjoe.org
....
; Addresses for local hosts
localhost      IN      A       127.0.0.1
abydos         IN      A       192.168.1.54
sleet          IN      A       192.168.1.41
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 52

# Trace

---

```
# tethereal -n -i wlan0 icmp or port 53
Capturing on wlan0
0.000000 192.168.1.178 -> 192.168.1.56
    DNS Standard query A sleet.everest.plainjoe.org
0.003927 192.168.1.56 -> 192.168.1.178
    DNS Standard query response A 192.168.1.41
5.002181 192.168.1.178 -> 192.168.1.56
    DNS Standard query A sleet.everest.plainjoe.org
5.006624 192.168.1.56 -> 192.168.1.178
    DNS Standard query response A 192.168.1.41
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 53

# DNS Reply Packet

---

```
Domain Name System (response)
Transaction ID: 0xde76
Flags: 0x8780 (Standard query response, No error)
    .... ..1. .... = Truncated: Message is truncated
    .... .... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 21
Additional RRs: 0
Queries
    sleet.everest.plainjoe.org: type A, class IN
Answers
    sleet.everest.plainjoe.org:
        type A, class IN, addr 192.168.1.41
        Name: sleet.everest.plainjoe.org
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 1 day
        Data length: 4
        Addr: 192.168.1.41
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 54

# DNS Delays?

```
$ cat /etc/resolv.conf
domain ad.plainjoe.org
nameserver 192.168.1.101

$ host worm.plainjoe.org
;; connection timed out; no servers could be reached
```

```
$ tethereal -n -i wlan0 port 53
Capturing on wlan0
0.000000 192.168.1.178 -> 192.168.1.101
      DNS Standard query A worm.plainjoe.org
5.001037 192.168.1.178 -> 192.168.1.101
      DNS Standard query A worm.plainjoe.org
15.191964 192.168.1.101 -> 192.168.1.178
      DNS Standard query response, Server failure
20.200041 192.168.1.101 -> 192.168.1.178
      DNS Standard query response, Server failure
```

# Follow the Bread Crumbs

```
$ host -r worm.plainjoe.org 192.168.1.101
;; nothing

$ host -r -t NS plainjoe.org 192.168.1.101
plainjoe.org name server thorn.plainjoe.org.

$ host -r thorn.plainjoe.org 192.168.1.101
thorn.plainjoe.org has address 192.168.1.56

$ host worm.plainjoe.org 192.168.1.56
worm.plainjoe.org has address 192.168.1.79
```

# Tracing Windows DNS

```
c:\> tethereal -r dns.pcap -R "ip.addr == 192.168.1.56"
 4  0.011988 192.168.1.101 -> 192.168.1.56
    DNS Standard query A worm.plainjoe.org
 6  5.002069 192.168.1.101 -> 192.168.1.56
    DNS Standard query A worm.plainjoe.org
 7  5.424241 192.168.1.101 -> 192.168.1.56
    DNS Standard query A worm.plainjoe.org
13 10.424791 192.168.1.101 -> 192.168.1.56
    DNS Standard query A worm.plainjoe.org
15 10.979990 192.168.1.101 -> 192.168.1.56
    DNS Standard query AAAA phzzbt.plainjoe.org
```

## FTP

- Separate streams for control and data
- Originally the server performed an active open back to the client
  - ❑ client sent PORT command to notify the server regarding which port to connect to
- Newer passive mode requires the client to issue the active open
  - ❑ server sends the port number in the PASV reply

# FTP PORT Command

```
$ tethereal -n -i wlan0 host 192.168.1.56
0.003808 192.168.1.178 -> 192.168.1.56 TCP 59398 > 21 [SYN]
0.006091 192.168.1.56 -> 192.168.1.178 TCP 21 > 59398 [SYN, ACK]
0.006128 192.168.1.178 -> 192.168.1.56 TCP 59398 > 21 [ACK]
0.100249 192.168.1.56 -> 192.168.1.178 TCP 38309 > 113 [SYN]
0.100285 192.168.1.178 -> 192.168.1.56 TCP 113 > 38309 [RST, ACK]
0.102590 192.168.1.56 -> 192.168.1.178 FTP Response: 220 ProFTPD
0.105328 192.168.1.178 -> 192.168.1.56 FTP Request: USER anonymous
.....
1.139595 192.168.1.178 -> 192.168.1.56 FTP Request: PORT
....
1.142317 192.168.1.178 -> 192.168.1.56 FTP Request: LIST
1.146569 192.168.1.56 -> 192.168.1.178 TCP 20 > 32777 [SYN]
1.146617 192.168.1.178 -> 192.168.1.56 TCP 32777 > 20 [SYN, ACK]
1.148892 192.168.1.56 -> 192.168.1.178 TCP 20 > 32777 [ACK]
1.150032 192.168.1.56 -> 192.168.1.178 FTP Response:
    150 Opening ASCII mode data connection for file list
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 59

# FTP PASV Command

```
$ tethereal -n -i wlan0 host 192.168.1.56
0.100943 192.168.1.56 -> 192.168.1.178 FTP Response: 220 ProFTPD
...
1.196822 192.168.1.178 -> 192.168.1.56 FTP Request: PASV
1.199359 192.168.1.56 -> 192.168.1.178 FTP Response:
    227 Entering Passive Mode (192,168,1,56,149,176).
1.200114 192.168.1.178 -> 192.168.1.56 TCP 32779 > 38320 [SYN]
1.202345 192.168.1.56 -> 192.168.1.178 TCP 38320 > 32779 [SYN, ACK]
1.202381 192.168.1.178 -> 192.168.1.56 TCP 32779 > 38320 [ACK]
1.203063 192.168.1.178 -> 192.168.1.56 FTP Request: LIST
1.206525 192.168.1.56 -> 192.168.1.178 FTP Response:
    150 Opening ASCII mode data connection for file list
1.210180 192.168.1.56 -> 192.168.1.178 FTP-DATA FTP Data: 681 bytes
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 60

# Passive FTP and Firewalls

```
$ ncftp ftp.plainjoe.org
Connecting to 192.168.1.56...
ProFTPD 1.2.10 Server (ProFTPD) [192.168.1.56]
Logging in...
Anonymous access granted, restrictions apply.
Logged in to ftp.plainjoe.org.
ncftp / > ls
Data connection timed out.
```

```
$ tethereal -n -i wlan0 host 192.168.1.56
0.100943 192.168.1.56 -> 192.168.1.178 FTP Response: 220 ProFTPD
...
2.148186 192.168.1.178 -> 192.168.1.56 FTP Request: PASV
2.150579 192.168.1.56 -> 192.168.1.178 FTP Response:
      227 Entering Passive Mode (192,168,1,56,149,170).
2.150916 192.168.1.178 -> 192.168.1.56 TCP 32778 > 38314 [SYN]
5.149642 192.168.1.178 -> 192.168.1.56 TCP 32778 > 38314 [SYN]
11.148730 192.168.1.178 -> 192.168.1.56 TCP 32778 > 38314 [SYN]
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 61

## DHCP

- Protocol for assigning client IP addresses and related parameters (DNS servers, netmask, etc...)
  - bootpc (68/udp)
  - bootps (67/udp)
- DHCP/BOOTP gateways forward broadcast requests (255.255.255.255) to a central server
- Simple protocol. What could possibly go wrong?



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 62

# Example DHCP Session

```
# tethereal -n -i eth0 port 67 or port 68
Capturing on eth0
0.000000      0.0.0.0 -> 255.255.255.255
                DHCP DHCP Discover - Transaction ID 0x7ae74002
0.090627 192.168.1.56 -> 255.255.255.255
                DHCP DHCP Offer   - Transaction ID 0x7ae74002
0.092209      0.0.0.0 -> 255.255.255.255
                DHCP DHCP Request - Transaction ID 0x7ae74002
0.217145 192.168.1.56 -> 255.255.255.255
                DHCP DHCP ACK     - Transaction ID 0x7ae74002
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 63

## No Lease?

```
# dhclient eth0
Listening on LPF/eth0/00:0c:29:3c:69:d2
Sending on   LPF/eth0/00:0c:29:3c:69:d2
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 10
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 14
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 19
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 11
No DHCPOFFERS received.
```

```
# tethereal -n -i wlan0 port 67 or port 68
0.000000      0.0.0.0 -> 255.255.255.255 DHCP
                DHCP Discover - Transaction ID 0xc065646a
7.428451      0.0.0.0 -> 255.255.255.255 DHCP
                DHCP Discover - Transaction ID 0xc065646a
19.376876     0.0.0.0 -> 255.255.255.255 DHCP
                DHCP Discover - Transaction ID 0xc065646a
...
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 64

# Dead DHCP Server

```
# nmap -sU -p 67 -PI -PT 192.168.1.56

Starting nmap 3.81 ( http://www.insecure.org/nmap/ )
Interesting ports on thorn.plainjoe.org (192.168.1.56):
PORT      STATE SERVICE
67/udp    closed dhcpserver
MAC Address: 00:02:E3:14:D7:91 (Lite-on Communications)

Nmap finished: 1 IP address (1 host up) scanned in 0.394 seconds
```

```
# tethereal -n -i wlan0 host 192.168.1.56
0.532889 192.168.1.178 -> 192.168.1.56
        BOOTP [Malformed Packet]
0.534637 192.168.1.56 -> 192.168.1.178
        ICMP Destination unreachable (Port unreachable)
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 65

# No Available Leases

```
# nmap -sU -p 67 -PI -PT 192.168.1.56

Starting nmap 3.81 ( http://www.insecure.org/nmap/ )
Interesting ports on thorn.plainjoe.org (192.168.1.56):
PORT      STATE SERVICE
67/udp    open|filtered dhcpserver
MAC Address: 00:02:E3:14:D7:91 (Lite-on Communications)

Nmap finished: 1 IP address (1 host up) scanned in 0.394 seconds
```

```
# tethereal -n -i wlan0 host 192.168.1.56
0.114482 192.168.1.178 -> 192.168.1.56
        BOOTP [Malformed Packet]
0.215411 192.168.1.178 -> 192.168.1.56
        BOOTP [Malformed Packet]
```

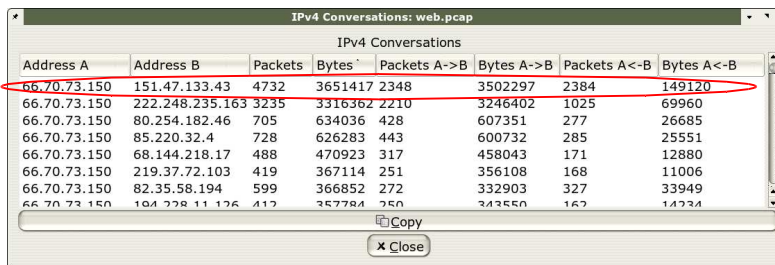


Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 66

# HTTP

- TCP based (port 80)
  - SSL enabled servers usually configured on port 443/tcp
- Generally simple text based command language
  - GET, PUT, POST, AUTH, etc....
- Three digit error codes similar to FTP and SMTP
  - 200 OK, 404 page not found, etc...
- Heavily used servers can deal with thousands of clients

## HTTP Statistics

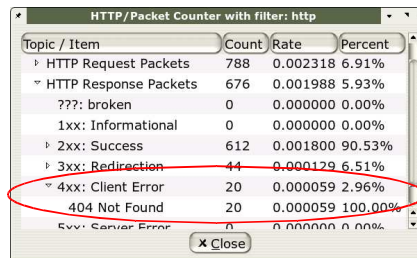


IPv4 Conversations: web.pcap

Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
66.70.73.150	151.47.133.43	4732	3651417	2348	3502297	2384	149120
66.70.73.150	222.248.235.163	3235	3316362	2210	3246402	1025	69960
66.70.73.150	80.254.182.46	705	634036	428	607351	277	26685
66.70.73.150	85.220.32.4	728	626283	443	600732	285	25551
66.70.73.150	68.144.218.17	488	470923	317	458043	171	12880
66.70.73.150	219.37.72.103	419	367114	251	356108	168	11006
66.70.73.150	82.35.58.194	599	366852	272	332903	327	33949
66.70.73.150	194.228.11.126	412	357794	250	243550	162	14234

chatty clients

404 Errors



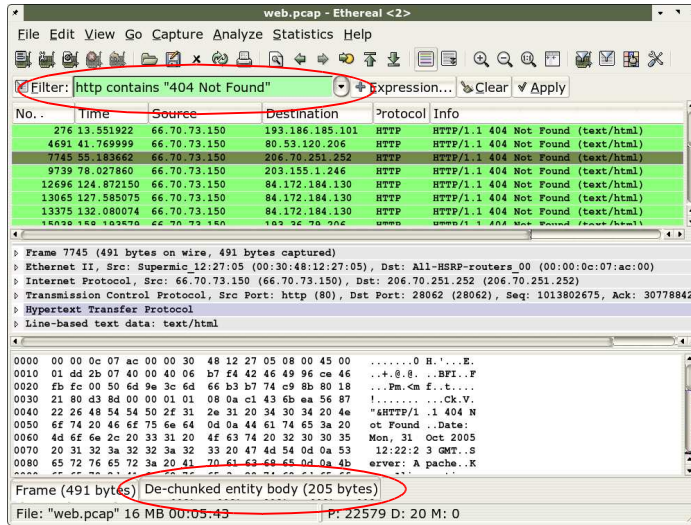
HTTP/ Packet Counter with filter: http

Topic / Item	Count	Rate	Percent
HTTP Request Packets	788	0.002318	6.91%
HTTP Response Packets	676	0.001988	5.93%
???: broken	0	0.000000	0.00%
1xx: Informational	0	0.000000	0.00%
2xx: Success	612	0.001800	90.53%
3xx: Redirection	44	0.000129	6.51%
4xx: Client Error	20	0.000059	2.96%
404 Not Found	20	0.000059	100.00%
5xx: Server Error	0	0.000000	0.00%

# HTTP 404

Filter all HTTP responses for "page not found"

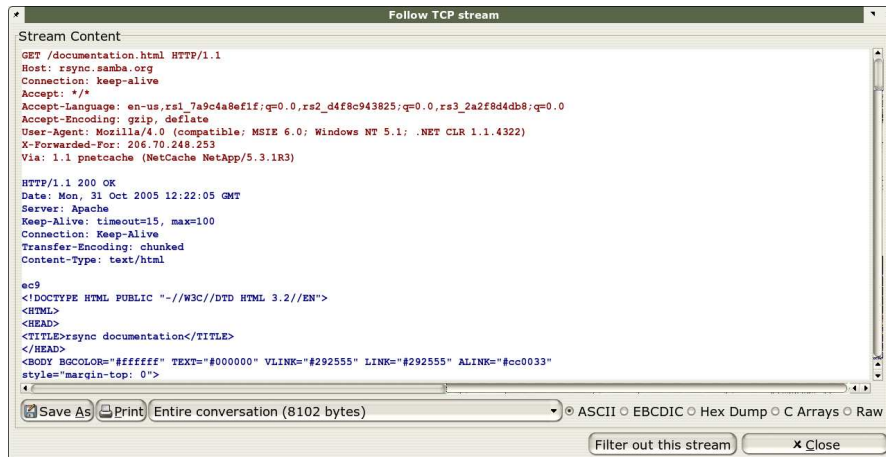
Reassemble fragmented HTML response



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 69

# TCP Streams

Analyze -> Follow TCP Stream



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 70

# Dealing with Encryption

- `ssldump` – protocol analyzer for SSL/TLS traffic
  - <http://www.rtfm.com/ssldump/>

```
$ ssldump -n -q -r /tmp/dump.pcap host 192.168.1.178

New TCP connection #1: 192.168.1.178(48688) <-> 66.70.73.153(443)
1 1 0.0356 (0.0356) C>S SSLv2 compatible client hello
1 2 0.1018 (0.0662) S>C Handshake ServerHello
1 3 0.1018 (0.0000) S>C Handshake Certificate
1 4 0.1018 (0.0000) S>C Handshake ServerKeyExchange
1 5 0.1018 (0.0000) S>C Handshake ServerHelloDone
...
New TCP connection #2: 192.168.1.178(48689) <-> 66.70.73.153(443)
2 1 0.1837 (0.1837) C>S Handshake ClientHello
1 28 1.9144 (0.2089) S>C application_data
1 29 1.9168 (0.0024) C>S application_data
2 2 0.2391 (0.0553) S>C Handshake ServerHello
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 71

# Decoding SSL/TLS

- `ssldump` can decode application data if provided with the appropriate key

```
$ ssldump -k ssl.key -dqn -r /tmp/dump.pcap host 192.168.1.178

New TCP connection #1: 192.168.1.178(46915) <-> 66.70.73.153(443)
....
1 10 0.2637 (0.1237) C>S application_data
-----
GET / HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.4; Linux)
          KHTML/3.4.0 (like Gecko)
Accept: text/html, image/jpeg, image/png, text/*, image/*, */*
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: bugzilla.samba.org
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 72

# NFS

---

- Several versions to deal with
  - ❑ NFS v1 and v2 used UDP
  - ❑ NFSv3 adds TCP support
  - ❑ NFSv4 latest incarnation
- Based on ONC-RPC
  - ❑ services register their port with the RPC portmapper
- eXternal Data Representation (XDR)

## Why does “mount -t nfs” hang?

---

```
# mount -t nfs worm.plainjoe.org:/export/u2/suse /mnt
....
```

```
# tethereal -n -i lo
Capturing on lo
0.000000 127.0.0.1 -> 127.0.0.1
    Portmap V2 UNSET Call
0.000694 127.0.0.1 -> 127.0.0.1
    ICMP Destination unreachable (Port unreachable)
4.998372 127.0.0.1 -> 127.0.0.1
    Portmap [RPC retransmission of #1]V2 UNSET Call
4.998405 127.0.0.1 -> 127.0.0.1
    ICMP Destination unreachable (Port unreachable)
....
```

# Mismatched NFS Versions

---

```
$ grep NFS /var/log/messages
Oct 31 09:23:41 fc4 automount[2241]:
    >> mount to NFS server 'ahab.plainjoe.org' failed:
    possible invalid protocol.
```

```
$ tethereal -r dump.pcap -n \
-R "portmap && ip.addr == 192.168.1.113"

13  0.045103 192.168.1.113 -> 192.168.1.79
    Portmap V2 GETPORT Call NFS(100003) V:3 TCP
15  0.045585 192.168.1.79 -> 192.168.1.113
    Portmap V2 GETPORT Reply (Call In 13)
    PROGRAM_NOT_AVAILABLE
```

# CIFS/SMB

---

- Utilized by Windows as well as non-Microsoft platforms
- NetBIOS services vs. TCP+UDP+DNS
- Active Directory domains utilize many protocols
  - CIFS, DCE-RPC, LDAP, DNS, Kerberos 5, etc...
- More changes coming in Windows Vista
  - symbolic links, transactions, etc...
- Many features are negotiated
  - e.g. password encryption, 32-bit status codes, extended security (SPNEGO), and DFS support
- Resources listed at <http://devel.samba.org>

# Samba

---

- Windows 2000 client unable to logon to a Samba domain
- Answer: nmbd is running, but smbд has died
  - connection to port 139 (netbios-ssn) receives a RST

```
0.000000 172.16.29.128 -> 172.16.29.255
      NBNS Name query NB RAIN<20>
0.003109 172.16.29.1 -> 172.16.29.128
      NBNS Name query response NB 172.16.29.1

0.003883 172.16.29.128 -> 172.16.29.1
      TCP danf-ak2 > netbios-ssn [SYN]
0.003911 172.16.29.1 -> 172.16.29.128
      TCP netbios-ssn > danf-ak2 [RST, ACK]
```

# Network Path Not Found

---

```
C:\> net view \\lettuce
System error 53 has occurred.
The network path was not found.
```

```
$ tethereal -r xp_firewall.pcap -R 'nbns'
17  10.846761 172.16.29.128 -> 172.16.29.255
      NBNS Name query NB LETTUCE<20>
23  11.594905 172.16.29.128 -> 172.16.29.255
      NBNS Name query NB LETTUCE<20>
24  12.346497 172.16.29.128 -> 172.16.29.255
      NBNS Name query NB LETTUCE<20>
```

# Normal Krb5 SMB Session

```
Transmission Control Protocol, Src Port: 1091, Dst Port: 445
NetBIOS Session Service
SMB (Server Message Block Protocol)
  SMB Header
  Session Setup AndX Request (0x73)
  ....
  Security Blob: 608204FB06062B0601050502A08204EF308204EBA030302E...
  GSS-API Generic Security Service Application Program Interface
  OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
  SPNEGO
  negTokenInit
  mechTypes: 4 items
    Item: 1.2.840.48018.1.2.2 (MS KRB5 - Microsoft Kerberos 5)
    Item: 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
    Item: 1.2.840.113554.1.2.2.3 (KRB5 - Kerberos 5 - User to User)
    Item: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP)
  mechToken: 608204AD06092A864886F71201020201006E82049C308204...
  krb5_blob: 608204AD06092A864886F71201020201006E82049C308204...
  Native OS: Windows 2000 2195
  Native LAN Manager: Windows 2000 5.0
  Primary Domain:
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 79

# AD and NTLMSSP Fallbacks

- Windows client do not obtain a service ticket when accessing a server using an IP in the UNC path

```
Transmission Control Protocol, Src Port: 1106, Dst Port: 445
NetBIOS Session Service
SMB (Server Message Block Protocol)
  SMB Header
  Session Setup AndX Request (0x73)
  ....
  Security Blob: 604806062B0601050502A03E303CA00E300C060A2B060104...
  GSS-API Generic Security Service Application Program Interface
  OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
  SPNEGO
  negTokenInit
  mechTypes: 1 item
    Item: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP)
  mechToken: 4E544C4D5353500001000000978208E200000000000000...
  NTLMSSP
  ....
```



Copyright Gerald Carter, 2005-2006. All rights reserved  
jerry@samba.org, Slide 80

# LDAP

---

- Nine core operations
  - bind, unbind, abandon
  - search, compare
  - modify, add, delete, moddn
  - plus extended operations and controls
- Generally string based with some LBER encoding
- Does include data privacy via
  - LDAPS protocol
  - StartTLS extended operation
- Distributed directories are linked together via referrals and references

## Where can things go wrong?

---

- Invalid authentication
  - clear text binds, Kerberos ticket failures, etc...
- Referrals to unknown servers
- Misconfigured search filters
- Failures in extended operations or controls when marked as critical
- Limits on search result sizes or durations
- Missing attributes or object classes
- Misconfigured access control lists
- Corrupted indexes
- blah, blah, blah, ...

# LDAP Search

- Each result is returned in a single PDU

```
$ ldapsearch -h ldap.plainjoe.org -x \  
-b 'dc=plainjoe,dc=org' '(uid=jerry)' objectclass  
  
dn: uid=jerry,ou=people,dc=plainjoe,dc=org  
objectClass: inetOrgPerson  
objectClass: posixAccount  
objectClass: sambaSamAccount  
objectClass: sambaIdmapEntry
```

```
# tethereal -n -r ldap_search.pcap -R "ldap.message_id == 2"  
8 0.010113 192.168.1.178 -> 192.168.1.56  
   LDAP MsgId=2 Search Request, Base DN=dc=plainjoe,dc=org  
9 0.014480 192.168.1.56 -> 192.168.1.178  
   LDAP MsgId=2 Search Entry  
10 0.014981 192.168.1.56 -> 192.168.1.178  
   LDAP MsgId=2 Search Result
```

# LDAP Search Result

```
Transmission Control Protocol, Src Port: 389 (389),  
  Dst Port: 56763 (56763), Seq: 1088281708,  
  Ack: 2698029667, Len: 129  
Lightweight Directory Access Protocol  
LDAP Message, Search Entry  
  Message Id: 2  
  Message Type: Search Entry (0x04)  
  Message Length: 122  
  Response To: 8  
  Time: 0.003292000 seconds  
  Distinguished Name: uid=jerry,ou=people,dc=plainjoe,dc=org  
  Attribute: objectClass  
    Value: inetOrgPerson  
    Value: posixAccount  
    Value: sambaSamAccount  
    Value: sambaIdmapEntry
```

^D

---

## Ethereal & the Art of Debugging Networks

Gerald (Jerry) Carter  
SAMBA Team  
Centeris

<http://www.plainjoe.org/>  
[jerry@samba.org](mailto:jerry@samba.org)  
<http://www.centeris.com/>



Copyright Gerald Carter, 2005-2006. All rights reserved  
[jerry@samba.org](mailto:jerry@samba.org), Slide 85