IP Fragmentation
Overlapping
ByPassing IDS
$ whois jselvi

- Jose Selvi (jselvi@pentester.es)
- Ethical Hacking & Pentesting
- Telefónica Ingeniería de Seguridad
- Pentester.es (http://www.pentester.es)
Let's Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
• RFC = Request for Comments ([http://www.ietf.org/rfc.html](http://www.ietf.org/rfc.html))

• All Protocols are fully defined by RFCs

• Fully? No!!

• One small set of possible situations still holds out being undefined
3-Way HandShake

Client

Server
3-Way HandShake

Client

Server

Syn
3-Way HandShake

Client → Server
Syn

Server → Client
Syn, Ack
3-Way HandShake

Client

Server

Syn

Syn, Ack

Ack
3-Way HandShake

Client

Syn

Syn , Ack

Ack

Data

Server

[Diagram showing the 3-Way Handshake with Client and Server exchanging packets: Syn, Syn Ack, Ack, Data]
3-Way HandShake

Client

Server
3-Way HandShake

Client

Syn, Rst, Ack

Server
3-Way HandShake

Client -> Server

- Syn
- Rst
- Ack

Server -> WTF?!
3-Way HandShake

Client

Syn, Rst, Ack → Server

WTF?!

¿?

Client → ¿?
Abuse: OS Fingerprinting

• Each coder solves it in a different way
• So... each different TCP/IP Stack response different
• Used for OS fingerprinting

• Different TCP/IP Stacks can work different? That’s sounds evily interesting!
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
To Fit or not to Fit

- MTU = Maximum Transfer Unit
- Depending on Layer 2 Network
  - Ethernet = 1500 bytes

- To Fit or not to Fit. That’s the question.
- What if doesn’t fit?
- IP FRAGMENTATION!
IP Fragmentation

Packet
Packet
Packet
IP Fragmentation
IP Fragmentation

Packet

Packet
IP Fragmentation
IP Fragmentation
IP Fragmentation
IP Fragmentation
IP Fragmentation

Packet
IP Fragmentation

Packet
IP Fragmentation

Fragment
Fragment
Fragment
IP Fragmentation
IP Fragmentation

Fragment

Fragment
IP Fragmentation
IP Fragmentation
IP Fragmentation
IP Fragmentation
IP Header

- IPID = IP Identifier
- MF Flag = More Fragments
- Fragment Offset
Howto Fragment

4000 bytes
MF = 0
Offset = 0
Howto Fragment

4000 bytes
MF = 0
Offset = 0
Howto Fragment

- 1500 bytes
  - MF = 1
  - Offset = 0

- 1500 bytes
  - MF = 1
  - Offset = 1500

- 1000 bytes
  - MF = 0
  - Offset = 3000
How to Defragment

- 1500 bytes, MF = 1, Offset = 1500
- 1000 bytes, MF = 0, Offset = 3000
- 1500 bytes, MF = 1, Offset = 0
Howto Defragment

1500 bytes
MF = 1
Offset = 0

1000 bytes
MF = 0
Offset = 3000

1500 bytes
MF = 1
Offset = 1500

1500 bytes
MF = 1
Offset = 3000
How to Defragment

1500 bytes
MF = 1
Offset = 0

1500 bytes
MF = 1
Offset = 1500

1000 bytes
MF = 0
Offset = 3000
# Howto Defragment

<table>
<thead>
<tr>
<th>Size</th>
<th>MF</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500 bytes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1500 bytes</td>
<td>1</td>
<td>1500</td>
</tr>
<tr>
<td>1000 bytes</td>
<td>0</td>
<td>3000</td>
</tr>
</tbody>
</table>
Howto Defragment

4000 bytes
 MF = 0
 Offset = 0
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
Let’s Go!

• Having Fun with RFCs
• IP Fragmentation
• Overlapping & Defragmentation
• ByPassing IDS
• Overlapping Defenses
Overlapping

- 200 bytes
  - MF = 1
  - Offset = 0

- 300 bytes
  - MF = 1
  - Offset = 100

- 100 bytes
  - MF = 0
  - Offset = 400
Overlapping
Overlapping
Defragmentation

• Blue or Green?
  • Not defined by RFC

• So... each OS do it by its own

• There are 7 different policies
Policies & OS’s

- First: HP-UX, MacOS, SunOS <5.8
- Last: Cisco
- BSD: AIX, FreeBSD, HP-UX 10.x, IRIX
- BSD-Right: HP Printers (some of them)
- Linux: OpenBSD, Linux
- Windows
- Solaris: Solaris 9 and 10
First Policy

Policy:

1) Always accept the first value received for each byte.
First Policy

Policy:

1) Always accept the first value received for each byte.
First Policy

Policy:

1) Always accept the first value received for each byte.
First Policy

Policy:

1) Always accept the first value received for each byte.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
</table>
First Policy

Policy:

1) Always accept the first value received for each byte.
Linux Policy

Policy:

1) Accept lower offset packet bytes received
2) With same offset, accept last received bytes
Linux Policy

Policy:

1) Accept lower offset packet bytes received
2) With same offset, accept last received bytes
**Linux Policy**

**Policy:**

1) Accept lower offset packet bytes received  
2) With same offset, accept last received bytes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>4</th>
<th>4</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th></th>
<th></th>
</tr>
</thead>
</table>
Linux Policy

Policy:

1) Accept lower offset packet bytes received
2) With same offset, accept last received bytes
Linux Policy

Policy:

1) Accept lower offset packet bytes received
2) With same offset, accept last received bytes
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
IDS & Signatures

• Usually **signature** based IDSs

• Signature = string or regular expression
  • Does it match with packet? => ALERT!

• Evil at Target but not at IDS?
  • Target Policy !== IDS Policy?
  • Possible with **IP Fragmentation**
Overlapping

GET /../../ETC/P

FOOFOOASSWOR

D HTTP/1.1
Target => ../../ETC/PASSWD

GET ../../ETC/P ASSWD HTTP/1.1

IDS => ../../../FOOFOOASSWD

GET ../../../FOOFOOASSWD HTTP/1.1
• “Insertion, Evasion, and Denial of Service: Eluding Networking Intrusion Detection”, January 1998

ip_frag size [old|new]

Fragment each packet in the queue into size-byte IP fragments, preserving the complete transport header in the first fragment. Optional fragment overlap may be specified as old or new, to favor newer or older data.
Windows Frag

- Policy: Always accept the first value received for each byte.
- First value = Older value

```
ip_frag 40 old
order random
print
```

- `fragroute -f ncn.conf 192.168.0.100`
DEMO
ByPassing SNORT with IP Fragmentation (I)
Problems

- Attack String is still there!
- Why not detected?
  - Packet dropped for bad checksum
- What if packet inspected anyway?
  - Bypass doesn’t work!
- Can we improve it with FragRouter?
FragRoute

ip_chaff \textbf{dup}|opt|ttl

Interleave IP packets in the queue with duplicate IP packets containing different payloads, either scheduled for later delivery, carrying invalid IP options, or bearing short time-to-live values.

delay first|\textbf{last}|random ms

Delay the delivery of the first, last, or a randomly selected packet from the queue by ms milliseconds.

drop first|last|random prob-%

Drop the first, last, or a randomly selected packet from the queue with a probability of prob-% percent.
First/BSD Vs Linux

- Policy: With same offset:
  - First/BSD => First Fragment
  - Linux => Last Fragment
- Bypass = First fragments OK, Last fragments garbage

```
ip_frag 40
delay last 1
ip_chaff dup
drop last 100```
DEMO

ByPassing SNORT with IP Fragmentation (II)
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
Let’s Go!

- Having Fun with RFCs
- IP Fragmentation
- Overlapping & Defragmentation
- ByPassing IDS
- Overlapping Defenses
Defenses

• SNORT: Frag3 Preprocessor
  • Others should have something similar

• Makes Snort speak in OS language
• You have to configure for each one
DEMO

Frag3 against IP Fragmentation
Other Defenses

- Force Defragmentation at Perimeter
- Reject Fragmented Packets
- Proxys
- NAT

- Keep out with network design!
Network Design (I)

- Internet
- Firewall / Proxy
- IDS
- Windows
- Linux
Network Design (II)

Diagram:
- Internet
- IDS
- Firewall / Proxy
  - Windows
  - Linux
Other Threats

- Feel safe?
- TCP Overlapping
- TTL ..... 
- Bad Checksum
- ...

Proverb

MORE HUMAN
LESS MACHINE
THANKS!
QUESTIONS?

Jose Selvi
Pentester.es
http://www.pentester.es
jselvi@pentester.es