IP Spoofing
(what is it, how it allows largest attacks and how to fix it)

Marek Majkowski
What is it?
IP Spoofing

(source: DaPuglet)
IP Spoofing

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Header Length</td>
<td>Service Type</td>
<td>Total Length</td>
<td>Identification</td>
<td>Flags</td>
</tr>
<tr>
<td>TTI</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.8.8.8</td>
<td>Source IP Addr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.6.7.8</td>
<td>Destination IP Addr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Padding</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enables impersonation

Real
8.8.8.8

Spoofed
8.8.8.8

Destination
5.6.7.8
IP Spoofing is still a problem
May 2000: BCP38
spoofer.caida.org

Measured Autonomic Systems

- Spoofable: 27.8%
- Inconsistent: 15.8%
- UnSpoofable: 56.4%
It allows largest attacks
Global network
Daily attacks
Some are super large
Direct: SYN Flood
"winter of attacks"
Amplification: SSDP
IP Spoofing

(source: DaPuglet)
IP Spoofing

1. Tracing back is impossible
2. Allows sophisticated attacks
Direct: SYN Flood
2. “Winter of attacks”
Tracing the attack

Attack starts
Tcpdump

```
$ tcpdump -ni eth0 -c 100

IP 94.242.250.109.47330 > 1.2.3.4:80: Flags [S], seq 1444613291, win 63243
IP 188.138.1.240.61454 > 1.2.3.4:80: Flags [S], seq 1995637287, win 60551
IP 207.244.90.205.17572 > 1.2.3.4:80: Flags [S], seq 1523683071, win 61607
IP 94.242.250.224.65127 > 1.2.3.4:80: Flags [S], seq 928944042, win 61778
IP 207.244.90.205.43074 > 1.2.3.4:80: Flags [S], seq 137074667, win 63891
IP 64.22.81.44.23865 > 1.2.3.4:80: Flags [S], seq 838596928, win 63808
IP 188.138.1.137.23373 > 1.2.3.4:80: Flags [S], seq 593106072, win 60272
IP 207.244.90.205.39653 > 1.2.3.4:80: Flags [S], seq 47289666, win 63210
IP 208.66.78.204.64197 > 1.2.3.4:80: Flags [S], seq 1850809890, win 62714
IP 207.244.90.205.33108 > 1.2.3.4:80: Flags [S], seq 319707959, win 63351
IP 207.244.90.205.6937 > 1.2.3.4:80: Flags [S], seq 1591500126, win 63902
IP 213.152.180.151.60560 > 1.2.3.4:80: Flags [S], seq 1902119375, win 62511
IP 64.22.79.127.11061 > 1.2.3.4:80: Flags [S], seq 1456438676, win 62148
```
Which router interface is it from?
Identifying interface

Attacks
Identifying the interface
Other side of the cable

- Local Internet Exchange
- Direct Peering
- Router
- Internet Carrier
- Server
1. Direct Peering
2. Internet Exchange

Local Internet Exchange → Router

3. Internet Carrier

Internet Carrier → Router
2. Internet Exchanges
2. Internet Exchanges

Local ISP #1

Local ISP #2

Local ISP #3

Internet Exchange L2 SWITCH

Router
3. Internet Carriers

Internet Carrier → Router
“Winter of attacks”
“Winter of attacks”

src IP= Hurricane Electric

Internet Carrier

LAX router
“Winter of attacks”
Lack of attribution

==

impossible to fight
Blocked with BPF

```iptables -A INPUT \
   --dst 1.2.3.4 \ 
   -p udp --dport 53 \ 
   -m bpf --bytecode "14,0 0 0 20,177 0 0 0,12 0 0 0,7 0 0 0,64 0 0 0,21 0 7 124090465,64 0 0 4,21 0 5 1836084325,64 0 0 8,21 0 3 56848237,80 0 0 12,21 0 1 0,6 0 0 1,6 0 0 0" \ 
   -j DROP
```
ldx 4*([14]&0xf)
ld #34
add x
tax

lb_0:
ldb [x + 0]
add x
add #1
tax
ld [x + 0]
jneq #0x07657861, lb_1
ld [x + 4]
jneq #0x6d706c65, lb_1
ld [x + 8]
jneq #0x03636f6d, lb_1
ldb [x + 12]
jneq #0x00, lb_1
ret #1

lb_1:
ret #0
Introducing the BPF Tools

03 Jul 2014 by Marek Majkowski.

In a recent article I described the basic concepts behind the use of Berkeley Packet Filter (aka BSD Packet filter or BPF) bytecode for high performance packet filtering, and the xt_bpf iptables module. In this post I'll explain how we use BPF and xt_bpf as one tool to deal with large scale DDoS attacks.

And, today, CloudFlare is open sourcing the tools we've created to generate and deploy BPF rules.

The Code

Our BPF Tools are now available on the CloudFlare Github:
https://github.com/cloudflare/bpftools

For installation instructions review the README, but typing make should do most of the work:

$ git clone https://github.com/cloudflare/bpftools.git
$ cd bpftools
$ make
Source IP addresses

Internet Carrier

LAX router

Hurricane Electric

???
IP Spoofing

1. Tracing back is impossible
2. Allows sophisticated attacks
Sophistication
1. UDP request-response

UDP Server

request

response

UDP Client
1. Amplification

Attacker

request

UDP Server

response

Target
1. Amplification factor

- **Attacker**
  - Request: 10 bytes
- **UDP Server**
  - Response: 100 bytes
- **Target**
1. Scale up!
June 2017: SSDP

- 5 Gbps of spoofing
- Exposed SSDP Devices
- 112 Gbps of traffic
Amplification easy to block

- Easy to block on firewall
  - udp and src port 1900

- Nicely dispersed geographically

- But tracing is impossible
The only way to keep online is to absorb the attack
Receive and process
Centralisation
Solution
Technical solutions

• BGP Flowspec
• Prevent IP spoofing - BCP38
• Netflow
BGP Flowspec is awesome
Flowspec

Router under attack
Adoption = nil
Prevent IP spoofing - BCP38
Failed! Live with it!

<table>
<thead>
<tr>
<th>Field</th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fragment Offset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header Checksum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source IP Addr</td>
<td>X.X.X.X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination IP Addr</td>
<td></td>
<td>5.6.7.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We're left with incompetent
Don't solve the IP spoofing!

Solve the attribution!
Netflow
Netflow
Netflow

• Open source netflow toolchain is great

• Scales well

• To avoid privacy issues
  • Rotate logs often
  • Set high sampling rate - 1/64k connections
Internet Carriers

Customer #1

Customer #2

Customer #3

Internet Carrier

Router
(netops)# nfdump -M db/waw01:1hr01 -R . -n2 -t -300 -s dstip/packets "in if 731"
Top 2 Dst IP Addr ordered by packets:
<table>
<thead>
<tr>
<th>Dst IP Addr</th>
<th>Flows(%)</th>
<th>Packets(%)</th>
<th>Bytes(%)</th>
<th>pps</th>
<th>bps</th>
<th>bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>173.245.58.40</td>
<td>1.0 M(77.0)</td>
<td>17.6 G(75.8)</td>
<td>1.1 T(22.6)</td>
<td>59.0 M</td>
<td>30.7 G</td>
<td>65</td>
</tr>
<tr>
<td>173.245.59.15</td>
<td>54962(4.0)</td>
<td>910.3 M(3.9)</td>
<td>75.5 G(1.5)</td>
<td>3.1 M</td>
<td>2.0 G</td>
<td>82</td>
</tr>
</tbody>
</table>

Summary: total flows: 1361108, total bytes: 5087980650496, total packets: 23271079936, avg bps: 135599480319, avg pps: 77524526, avg bpp: 218
Total flows processed: 2457140, Blocks skipped: 0, Bytes read: 177251772
Sys: 0.210s flows/second: 11700666.7 Wall: 0.210s flows/second: 11654603.2
Recap

• **BGP flowspec firewall**
  - A stop gap, useful for amplification attacks

• **Prevent IP spoofing - BCP38**
  - The root of all evil, but unfixable in short time

• **Netflow sampling**
  - Required for attack attribution
  - Supported by some competent ISP's / tier 1
  - Still not supported by majority!
Attribution allows informed discussion
The internet will be better for everyone.
How to help?
Help: report IP spoofing

- From spoofer.caida.org
Help: close NTP and DNS

- Scan your network for open NTP and DNS servers
- http://openntpproject.org/
- http://openresolverproject.org
- https://www.shodan.io/
Help: press for attribution

• When under attack
• Collect evidence
• Ask where the traffic came from!
Is amplification in decline?
Is amplification in decline?

- Very easy to block on firewall
  - udp and src port 123 == NTP attack
  - udp and src port 53 == DNS attack
- DDoS mitigation vendors have FAT pipes
- Amplification is bouncing off *real* servers
- Therefore geographically distributed
- Not effective against anycast
Why IP Filtering must be on the edge
Filtering is hard
Filtering is hard

Source 1.2.3.4 → ISP 1 1.2.3.0/24 → Internet Carrier A → Destination 5.6.7.8
Filtering is hard

Source 1.2.3.4 → ISP 1 → Internet Carrier A → Destination 5.6.7.8

1.2.3.0/24

Internet Carrier B
Filtering is hard

Source 1.2.3.4

ISP 1
1.2.3.0/24
4.3.2.0/24

Internet Carrier A

Destination 5.6.7.8

Source 4.3.2.1

ISP 2

Internet Carrier B
Internet is asymmetric
Filter close to the source

Source → ISP 1 → Internet Carrier A → Destination

ISP 1

Internet Carrier B