DDoS Mitigation Tutorial
NANOG 69

Krassimir Tzvetanov
Introduction and overview
Introduction

Who am I?

What is the target audience of this tutorial?

Let me know if I speak too fast!

Let’s make it interactive!
Overview

- Discuss what DDoS is, general concepts, adversaries, etc.
- What is currently fashionable?
- Go through a networking technology overview, in particular the OSI layers, sockets and their states
- Look at popular attack types at the different layers - DNS, NTP, SSDP reflection, SYN Flood
- Discuss reflection and amplification
- Mitigations
What is DoS/DDoS?
What is Denial of Service?

Resource exhaustion… which leads to lack of availability

Consider:
• How is it different from CNN pointing to somebody’s web site?
• How is that different from company’s primary Internet connection going down?
What is Denial of Service?

From security point of view?
- Decreased availability

From operations point of view?
- An outage

From business point of view?
- Financial losses
DoS vs. DDoS

One system is sending the traffic vs many systems are sending the traffic

In the past it _usually_ meant difference in volume

Over the past 3 years, due to reflective attacks, this has been changing rapidly
The problem?
Let’s look at attack bandwidth

• Bandwidth in 2010 – little over 100 Gbps?
• 2013 – over 300 Gbps
• 2014 – over 400 Gbps
• Nowadays – irrelevant, it is all about bragging rights

Source: Arbor Networks Yearly Report
Contributing factors

- Embedded devices (mostly home routers)
- Available reflectors (DNS, NTP, SSDP) …with ability to amplify
- Outdated Content Management Systems (CMSes)
- Hosting providers allowing reflection
- More overall bandwidth available
Embedded Devices *(aka IoT)*

- Home routers – increasing threat
  - Default passwords
  - Other vulnerabilities (NetUSB)
  - XBOX – attacks – Krebs’ blog (re: 2014 XBOX/Play Station attacks)
  - Some do not allow the user to turn off DNS resolution
  - Network diagnostic tools
Compromised CMSes

- Most targeted Content Management Systems:
  - WordPress
  - Joomla

- Started in early 2013

- Started with a particular group of people abusing it

- Now it is an easy way to build a botnet and other groups abuse it as well
Economics Considerations

How much does an attack really cost?
How much does the attacker pay per system?
Consider lack of other illegal activities profits

The life of a drone
- Financials related
- Spam related
- DDoS

How about IoT?
- Cost of ownership is low
- No financial gain (at this point)
Who is the adversary?
Adversary

Wide range of attackers
Gamers – on the rise!!! 😊
Professional DDoS operators and booters/stressors
Nation states
Hacktivists – not recently
…and more
Motivation

Wide range of motivating factors as well

Financial gain

Extortion (Stealth Ravens/DD4BC/Armada Collective/copy cats)

Taking the competition offline during high-gain events
  (online betting, Superbowl, etc).

Political statement

Divert attention (seen in cases with data exfiltration)

Immature behavior
Skill level

Wide range of skills:
- Depending on the role in the underground community
- Mostly segmented between operators and tool smiths
- Tool-smiths are not that sophisticated (at this point) and there is a large reuse of code and services
- This leads to clear signatures for some of the tools

Increasing complexity:
- DirtJumper
- xnote.1
- XOR Botnet
- Mirai
What is new(-ish)?
What is new?

- Booters/Stressors (3 years)
- Embedded home and SOHO devices (3-4 years), Mirai added a new spin to it
- Content management systems – (5 years)
Booters/Stressors

• Inexpensive

• Tools are sold for cheap on the black market (forums)

• Range 5-10 Gbps and up to 40GBps
  – over the past years there were mentions of 80GBps (but not conclusive)

• Usually short duration

• Popular among gamers
Booters/Stressors

• A picture is worth a thousand words:
  – Think about the audience they are trying to attract

• Google: “Gwapo’s Professional DDOS”
Home routers

• Embedded home and SOHO devices
  – Default username/password
  – Open DNS recursive resolvers
  – NetUSB bug
  – Network diagnostic tools
  – Some do not allow the user to turn off DNS

• XBOX and Sony attacks over Christmas (2014)
  – Krebs on security:
    http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/

• Is that intentional?
Technology and Terminology Overview
Technology Overview

The purpose of this section is to level set

Topics we’ll cover

• OSI and Internet models
• TCP and sockets
• DNS operation and terminology
• NTP, SNMP, SSDP operation
• Some terminology and metrics
Network Layers

OSI – Open Systems Interconnect

TCP/IP (Internet)

Application

Transport

Internet

Network Access

OSI Layers

Application

Presentation

Session

Transport

Network

Data Link

Physical
Physical and Data-link Layers (L1 and L2)

Aka: Network Access Layer

Physical
Media changes that carry information: voltage, phase
Line coding: Manchester, NRZ, NRZ-I
Data unit: bit

Data-link
Data unit: frame (organizes bits in a frame)
Provides physical addressing on a local network segment
Separate in two:
   Media Access Control Layer (MAC): 802.3, 802.4, 802.5, 802.11abe
   Logical-link Control: 802.2
Network Layer (layer 3)

Aka Internet Layer

Provides transport of data units between two points in the network
Provides logical (and global) addressing of machines in the network
Data unit: Packet
Examples: Internet Protocol (IP)
Does not guarantee delivery
Allows for fragmentation
Transport Layer (layer 4)

Aka Transport (hey, this one matches the Internet model!!) 😊

Provides logical connection between applications
Provides addressing of applications on a single system (via port numbers)
Data unit: segment
In some modalities like TCP provides virtual circuit and ensures data ordering and no loss of packets
Typical for TCP is the 3-way handshake
Examples: TCP, UDP
Session Layer (layer 5)

Part of the Internet model Application layer

Managing sessions between application (think state, like authentication)
Examples: HTTP, SMTP, NetBIOS
Addressing: some protocols provide logical endpoint
Presentation Layer (layer 6)

Part of the Internet model Application layer

Provides uniform data representation across multiple architectures and platforms
Examples: images, file encryption
Application Layer (layer 7)

This is where the application lives

Part of the Internet model
Application layer
Questions?
Attack surface
Physical and Data-link Layers

- Cut cables
- Jamming
- Power surge
- EMP

- MAC Spoofing
- MAC flood
Network Layer

- Floods (ICMP)
- Teardrop (overlapping IP segments)
Transport Layer

• SYN Flood
• RST Flood
• FIN Flood
• You name it…

• Window size 0
  (looks like Slowloris)
• Connect attack
• LAND (same IP as src/dst)
Session Layer

• Slowloris
• Sending data to a port with no NL in it (long headers, long request lines)
• Send data to the server with no CR
Presentation Layer

- Expensive queries (repeated many times)
- XML Attacks

```
<!DOCTYPE lolz [
  <!ENTITY lol1 "&lol2;">
  <!ENTITY lol2 "&lol1;">
]> 
<lolz>&lol1;</lolz>
```
Application Layer

• Depends on the application
• Black fax
Attack summary by layer

- Note the dependency between layer and compute power needed to mitigate.
Questions?
Transmission Control Protocol (TCP) and sockets
Introduction to TCP

- Provides end-to-end virtual circuit
- Manages data loss detection and retransmission
- Deals with datagram ordering

RFC: 793 / September 1981

TRANSMISSION CONTROL PROTOCOL

NANOG 69: DDoS Tutorial
Socket is an abstraction allowing an application to bind to a transport layer address (aka network port)

It is described by a state machine

Throughout its life time it goes through a number of states
Socket States

Here are some of the socket states of importance:

CLOSED – start state
LISTEN – waiting for a connection request
SYN_SENT – initiated a connection
SYN_RECV – received request still negotiating
ESTABLISHED – connection working OK
CLOSE_WAIT – waiting for the application to wrap up
FIN-WAIT1/2, CLOSING, LAST_ACK – one side closed the connection
TIME-WAIT – waiting for 2 x MSL
Opening a TCP connection

Let’s review the sequence for opening a connection

- Server side opens a port by changing to LISTEN state
- Client sends a SYN packet and changes state to SYN_SENT
- Server responds with SYN/ACK and changes state to SYN_RECV. For the client this is ESTABLISHED connection
- Client has to ACK and this completes the handshake for the server
- Packet exchange continues; both parties are in ESTABLISHED state
Sequence for closing a connection

- Both parties are in ESTABLISHED state
- One side initiates closing by sending a FIN packet and changes state to FIN_WAIT1; this changes the other side to CLOSE_WAIT
- It responds with ACK and this closes one side of the connection
- We are observing a half closed connection
- The other side closes the connection by sending FIN
- And the first side ACKs
- The first side goes into a wait for 2 times the MSL time (by default 60 seconds)
Use of netstat for troubleshooting

[root@knight ghost]# netstat -nap | grep 12345
TCP        0      0 0.0.0.0:12345               0.0.0.0:* LISTEN   2903/nc
[root@knight ghost]# netstat -nap | grep 12345
TCP        0      0 127.0.0.1:12345             127.0.0.1:49188 ESTABLISHED 2903/nc
[root@knight ghost]# netstat -nap | grep 12345
TCP        0      0 127.0.0.1:49188             127.0.0.1:12345 TIME_WAIT -
[root@knight ghost]# netstat -nap | grep 12345
[root@knight ghost]#
Attack types and terminology
Attack classification classifications
(pun intended) ;)

- By volume
  - Volumetric
  - Logic/Application
- Symmetry
  - Asymmetric
  - Symmetric
- Direction
  - Direct
  - Reflected

- Source
  - Single source
  - Distributed
- State change
  - Permanent
  - Recoverable

- Based on network layer
Important metrics and what to report

– Bandwidth (Kbps, Gbps)
– Latency
– PPS
– QPS
– Storage
– CPU
– Application specific – usually latency
– Protocol
Attack type details
SYN Flood
What is a SYN flood?

What is a 3-way handshake?

“I want to talk to you”
Flags: SYN
SEQ: 101; ACK: <not used>

“Are you real?”
Flags: SYN, ACK
SEQ: 550; ACK: 101+1

“Of course I am!”
Flags: ACK, ACK
SEQ: 101+1; ACK: 550+1
SYN flood

- Exploits the limited slots for pending connections
- Overloads them

Flags:
- SYN
- SEQ
- ACK

“Want to talk to you”

10.1.1.10

3.3.3.3

101
431
583
392
938

783

SYN flood through the eyes of netstat

netstat –anp

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
<th>PID/Program name</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:111</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1339/rpcbind</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:33586</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1395/rpc.statd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>192.168.122.1:53</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1962/dnsmasq</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:631</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1586/cupsd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>2703/sendmail: acce</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49718</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49717</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49722</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49720</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49719</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49721</td>
<td>SYN_RECV</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:49716</td>
<td>SYN_RECV</td>
<td></td>
</tr>
</tbody>
</table>
**SYN on the wire**

<table>
<thead>
<tr>
<th>Attacker</th>
<th>Random IP address/port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>127.0.0.1:80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1:80</td>
</tr>
</tbody>
</table>

Pay attention to the SYN flag!
SYN flood mitigation

• Technology
• SYN Cookies
• Whitelists
• TCP Proxy (TCP Intercept – active mode)
• TCP Resets (TCP Intercept – passive)
• Nowadays – volumetric
What is a SYN cookie?

Hiding information in ISN (initial sequence number)

SYN Cookie:

\[ \text{Timestamp} \% 32 + \text{MSS} + \text{24-bit hash} \]

Components of 24-bit hash:
- server IP address
- server port number
- client IP address
- client port
- timestamp >> 6 (64 sec resolution)
Enabling SYN-cookies

- To enable SYN cookies:
  ```
  echo 1 > /proc/sys/net/ipv4/tcp_syncookies
  ```

- All TCP related settings are located in `/proc/sys/net/ipv4/`
  - `tcp_max_syn_backlog`
  - `tcp_synack_retries`
  - `tcp_syn_retries`
Socket Exhaustion
Socket Exhaustion

What is a socket?

What is Maximum Segment Lifetime (MSL)?
How old is the Internet?
What is Time To Live (TTL) measured in?

What is socket exhaustion?
Socket exhaustion would look like this:
Active Internet connections (servers and established)

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
<th>PID/Program name</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:111</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1339/rpcbind</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:33586</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1395/rpc.statd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>192.168.122.1:53</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1962/dnsmasq</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:53</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1586/cupsd</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>2703/sendmail: acce</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0:1241</td>
<td>0.0.0.0:*</td>
<td>LISTEN</td>
<td>1851/nessusd: waiti</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60365</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60240</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60861</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60483</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60618</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60407</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60423</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60211</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60467</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1:25</td>
<td>127.0.0.1:60213</td>
<td>TIME_WAIT</td>
<td>-</td>
</tr>
</tbody>
</table>
How to enable socket reuse (IoT issue)

- How to determine
  
  cat /proc/sys/net/ipv4/tcp_fin_timeout
  sysctl net.ipv4.tcp_fin_timeout

- Enable socket reuse
  
  echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle
  echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse
Slowloris
Connection handling architectures

Process based connection handling?
Think “Apache”

Event based connection handling?
Think “nginx”
Process oriented explained

- Listener opens sockets
- New connection comes in
- Process forks; separate process handles the connection
- New connection comes in
- Process forks; separate process handles the connection
- ...and so on...
- ...usually with up to 500-600 concurrent process copies
Apartment web server (simplified)

- Few child processes listen on a socket
- A new connection comes in...
- ...and one of them takes it
- Another new connection comes in...
- ...and the next one takes it.
- Pool is exhausted; new processes are spawned (forked)
- ...and so on...
- Up to about 500-600
- The initial set is defined by StartServers
Nginx (simplified)

- Master Process controls logistics
- Support processes (cache management)
- Worker processes process connections
  - One or more…
    - …one per core
  - Each worker can handle many sockets concurrently
  - A new connection comes in
    - …and is established; no dup()
    - …and so on…
Slowloris

• Exploits the process based model but opening a number of concurrent connections and holds them open for as long as possible with the least amount of bandwidth possible
Slowloris request

Request:
send: GET /pki/crl/products/WinPCA.crl HTTP/1.1
wait...
send: Cache-Control: max-age = 900
wait...
send: Connection: Keep-Alive
wait...
send: Accept: */*
wait...
send: If-Modified-Since: Thu, 06 Aug 2015 05:00:26 GMT
wait...
send: User-Agent: Microsoft-CryptoAPI/6.1
wait...
send: Host: crl.microsoft.com
The client opens a connection and sends a request…
...then another…
...and another…
...and so on.

...and waits some time...
...and sends the next header
...and so for each connection
...and so on…
Slowloris mitigation

• Change of the software architecture

• Use of event driven reverse proxy to protect the server (like nginx)

• Dedicated hardware devices
Questions?
Reflection and amplification attacks
Two different terms

- Reflection using an intermediary to deliver the attack traffic
- Amplification ability to deliver larger response than the trigger traffic
Reflection
Reflective attacks

- Attacks where an unwilling intermediary is used to deliver the attack traffic.

- The attacker would normally send a packet with a forged source IP address to the intermediary. The forged address is going to be the one of the target. The intermediary will deliver a response which will go to the target instead of the attacker.

Note to audience: think what protocols we can use for that?
What is reflection(ed) attack

Attacks where an unwilling intermediary is used to deliver the attack traffic

Attacker sends a packet with a spoofed source IP set to the victim’s Reflectors respond to the victim

```
S: 191.236.103.221 D: 3.3.3.3
Size: 64 bytes

S: 3.3.3.3 D: 191.236.103.221
Size: 512 bytes
```
The ones that are of interest are:

- DNS
- NTP
- SSDP
- SNMP
- RPC (reported lately but not really large)
Amplification
What is amplification attack?

- Asymmetric attack where response is much larger than the original query
Amplifiers types

The ones that are of interest and provide amplifications are:

- DNS
- SSDP
- NTP
- SNMP

Amplification factors:
https://www.us-cert.gov/ncas/alerts/TA14-017A
## Amplification quotients

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bandwidth Amplification Factor</th>
<th>Vulnerable Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS</td>
<td>28 to 54</td>
<td>Multiple</td>
</tr>
<tr>
<td>NTP</td>
<td>556.9</td>
<td>Multiple</td>
</tr>
<tr>
<td>SNMPv2</td>
<td>6.3</td>
<td>GetBulk request</td>
</tr>
<tr>
<td>NetBIOS</td>
<td>3.8</td>
<td>Name resolution</td>
</tr>
<tr>
<td>SSDP</td>
<td>30.8</td>
<td>SEARCH request</td>
</tr>
<tr>
<td>CharGEN</td>
<td>358.8</td>
<td>Character generation request</td>
</tr>
<tr>
<td>QOTD</td>
<td>140.3</td>
<td>Quote request</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>3.8</td>
<td>File search</td>
</tr>
<tr>
<td>Kad</td>
<td>16.3</td>
<td>Peer list exchange</td>
</tr>
<tr>
<td>Quake Network Protocol</td>
<td>63.9</td>
<td>Server info exchange</td>
</tr>
<tr>
<td>Steam Protocol</td>
<td>5.5</td>
<td>Server info exchange</td>
</tr>
</tbody>
</table>

Source: US-CERT: https://www.us-cert.gov/ncas/alerts/TA14-017A
Questions?
DNS Resolution
DNS server types

• Authoritative
  The source of truth for a particular domain name
  Example: Root DNS servers, .com DNS server, .google.com DNS server, etc.

• Recursive
  Service endpoints; optimize the DNS queries
  Example: corporate DNS server, home router DNS server
DNS resolution

• How does DNS work?

• User talks to recursive resolver

• The recursive goes on the Internet and talks to the authoritative servers

• When an answer is obtained (or not) it reports back to the user
DNS resolution at the packet level

• The process of mapping:
  www.fastly.com => 151.101.117.57

...if the answer was cached
DNS Reflection
What is DNS reflection attack?

- What happens if an attacker forges the victim address as its source?

S: \textbf{191.236.103.221} D: 3.3.3.3
---
What is the IP for www.cisco.com?
S: 3.3.3.3 D: \textbf{191.236.103.221}
---
www.cisco.com is at 5.5.5.5

...the reflected traffic goes to the target server

S: X.X.Y.Z D: \textbf{191.236.103.221}
---
S: X.X.Y.Z D: \textbf{191.236.103.221}
---
S: X.X.Y.Z D: \textbf{191.236.103.221}

... and what if hundreds of misconfigured open DNS resolvers are used?

NANOG 69: DDoS Tutorial
Consider this query

• Triggered by something like:
  • `dig ANY isc.org @3.3.3.3`

• Example:~$ `dig ANY isc.org @172.20.1.1` # My home lab
  - Flip over for answer
Consider this (cont’d)

ghostwood@sgw:~$ dig ANY isc.org @172.20.1.1

;; ANSWER SECTION:

isc.org.                481     IN      RRSIG   DS 7 2 86400 20130517145725 20130517145725 42353 org. KHMs09DaFmx416/7xHaxaD9By0NrqClO4kbBnqi6oq2VocZRRERaUbUHRrAY
KydolgKO5vOaw56lfY86/oOlDk3y3hHypciwvdjufelu4PktldUn1iQXW7 7919/jWgHBL51QiQgBYvZ75Zf y1En+6fIPoCghAyWgEYdqcW8pzOjz zlU=

isc.org.                481 IN   DS 12892 5 2 F1E184C0E1D615D20EB3C232ACED3B03C773DD952D5F0EB5C777586D E18A6D5B

isc.org.                5725 IN   RRSIG   A 5 2 7200 20130620134150 20130521134150 50012 isc.org. icBy1j9P6mXYjaSc62JCIrZw+hvYAUHGoh7WwRmxGRapSpI9L+lClvRI
2erglmomkBP79mahnFOxWEauea6UlHCIGxOkgr3hBIMFJUB9hrvkmuxO2D8gc1DJDL5egfpJCFCf2ffThFeMVteM8tQGYNwicWmXsFHCxM7Fms D8I=

isc.org.                5725 IN   A 149.20.64.42

isc.org.                5725 IN   RRSIG   DNSKEY 5 2 7200 20130620130130 20130521130130 12892 isc.org. dfxTGA/f6vdhuqojp+Konkd8c4y3WUIrVs5TjzvhdEyh14qph/cHh
+y1A6gAwTH14X+GpzcTnxlEwaSwvU3m9Nocniw/AZQol/SyDgsEsl.J/M+X+ZXY5qQq6rV2grOcKAAA91Bus3beHYZTsa2h2TSTAKAjIINEgvm
yQ5KwEo6e3Epl0pgQ4eMN04f9T9U9DHR3D3v3ztxFINNVksmWZ9GBH05fQcbOxnyGyn1bBptJEEGKhCBG01ncJ1TMCyFE98VGHKJFeowOiirDQ3
cjJRFPtCTCkA8n4JvsnimUJP/TGl+yMg4AUZP96j6j9vFbdScCl/0O61XKQVIA==

isc.org.                5725 IN   RRSIG   DNSKEY 5 2 7200 20130620130130 20130521130130 50012 isc.org. o18F3KlFkYedFRw1e5MP4QDo3wSg0XK9f5WCYD75aGhs9Ri5eyc/6KEW Se4lZXRh6f6d77xXlemYMCrshf/GHJdPrE6x1Ln/zH/h7TBJA9XDbC5I/ EUpFiGVLvdQy43Ktywmoq2ybc5MdGa2VeLko+hHTmH3St3pQRWpj2kJk5Z0=

isc.org.                5725 IN   DNSKEY 257 3 5 BEAAAAdAoHQBdrhQbthqg2wQUpEQ5t4DTU1xQmMVFlu2hWLDmvoOMRXGrhreeFvAZh7yHf8ZGWM6h388XG/xyiYCO6Krbpbd0w6w6XMYX5s/Va/A u50WI/8LZ1R16KTbsYVMF+x5FlRiNbcPWv+vT+U8ueEJmo2J0s1ULgy3 47cBCBi1mznz4/LJpA0a9cbKj3A254T515sNIMcwsBSB2+2E63/zzRQz6ilkbN89BjExiipk3sJzhZaTmEx3rdT47T09UxI5WcJ+txqZ7+yusl_KOOoedS327ZSDmns2eA0FktqPWa6LXEg2w+jojxwmo3A8iVUEf/rzeC/cBbYBns070aEFTd

isc.org.                5725 IN   DNSKEY 256 3 5 BQEAADABwhu9Z9emOBjUjQTO7C/a3McR68Maufljy1df/onajjPy7v3XTrAOtoMeKrK+/x6eT4QRLnu0KzvZJnQnTIyjJaFTw2OM/ItBFh/hL2mCf2O7n3SMeqYtvyPhY7dWghYW4sFh7VVEGM958o9rnf7932Qekkxh x8pXWDeAARU=
Reflection and Amplification

S: 191.236.103.221 D: 3.3.3.3
What is ANY isc.org

S: 3.3.3.3 D: 191.236.103.221

NANOG 69: DDoS Tutorial
### On the wire

<table>
<thead>
<tr>
<th>Attack traffic</th>
<th>127.0.0.1</th>
<th>DNS</th>
<th>70 Standard query 0x4918 A test.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
<tr>
<td>127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
<tr>
<td>127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
<tr>
<td>127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>153 Standard query response 0x4918 A 192.168.1.1</td>
</tr>
<tr>
<td>Reflector</td>
<td>127.5.5.5</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
</tbody>
</table>

| Target         | 127.0.0.1 | ICMP | 181 Destination unreachable (Port unreachable) |

- **Victim is 127.5.5.5**
- **Attacker spoofs traffic as if it comes from 127.5.5.5**
- **Reflector (127.0.0.1) responds to the query to the victim**

**BACK SCATTER**

Notice the victim is responding with port unreachable because there is nothing running on that UDP port. This is called backscatter
### On the wire (details)

<table>
<thead>
<tr>
<th>Frame 35820</th>
<th>128.1479010127.5.5.5</th>
<th>127.0.0.1</th>
<th>DNS</th>
<th>70 Standard query 0x4918 A test.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>35821</td>
<td>128.1479080127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
<tr>
<td>35822</td>
<td>128.1479150127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>70 Standard query 0x4918 A test.com</td>
</tr>
<tr>
<td>35823</td>
<td>128.1479410127.5.5.5</td>
<td>127.0.0.1</td>
<td>DNS</td>
<td>153 Standard query response 0x4918 A 192.168.1.1</td>
</tr>
<tr>
<td>35824</td>
<td>128.1479440127.5.5.5</td>
<td>127.0.0.1</td>
<td>ICMP</td>
<td>181 Destination unreachable (Port unreachable)</td>
</tr>
</tbody>
</table>

- **Victim is 127.5.5.5**
- **Attack traffic from 127.5.5.5; port 49249**
- **To reflector 127.0.0.1; port 53**
On the wire (details)

- Reflector (127.0.0.1) responds to the query to the victim (127.5.5.5)

- Note the number of records in the answer
Cache busting
DNS resolution (rehash)

• Let’s focus on the number of requests per second

• User talks to recursive resolver, which:
  – Caches answers
  – Answers a large number of requests

• The recursive talks to different level of authoritative servers, which:
  – Do not cache answers (they are auths)
  – Relatively lower number of queries

• Consider caching and authoritative capacity
What is cache busting?

Attacker sends a query to recursive/reflector
Recursive forwards the query
And so on…
Imagine one more recursive resolver
Rinse and repeat…
Questions?
Large scale mitigation and load distribution: Anycast

- Multiple points of presence advertise the same address space
- Network ensures user is routed to the “closest” instance
Network Time Protocol (NTP)
NTP reflection attack

• Stratum servers
• NTP queries

• MONLIST command
  – provides a list of clients that have time readings
NTP server configuration

Access lists

NTP authentication

Disable the MONLIST command

Useful hints:  
http://www.team-cymru.org/secure-ntp-template.html

List of open NTP reflectors:  
http://openntpproject.org/
Simple Network Management Protocol (SNMP)
• Different researchers claim amplification factors larger than the ones provided by NTP

• Tools floating in the wild

• Amplification 6 times according to US-CERT
Simple Service Discovery Protocol (SSDP)
SSDP

• Spoofed MSEARCH query with the source of the victim

• Amplification is up to 30 times (US-CERT)
Reflection attacks summary and resources

• Summary
  – Protocols that allow spoofing of the source of a query
  – Protocols that provide amplification – the query is much smaller than the response

• SSDP: http://openssdpproject.org/
• DNS: http://openresolverproject.org/
• NTP: http://openntpproject.org/
Questions?
Thank you!

krassi@fastly.com
Mitigation
Risk Pyramid

Number of attackers/attacks

Intensity

Resources
The cost of a minute?

• How much does a minute of outage cost to your business?
• Are there other costs associated with it? Reputation?
• Are you in a risk category?
• How much is executive management willing to spend to stay up?

• Are there reasons you need to mitigate on-site vs offsite? Latency?
On-site / DIY

• Bandwidth
• Equipment
• Qualified personnel
• More expensive overall but cheaper per MB
• Need for a backup plan
On Premise DDoS Mitigation
Outsource / scrubbing center

• Limited protocol support (usually HTTP/S)
• Added latency
• May lose visibility to source IP of the client
• Pay per MB of clean traffic (usually)
• Fast setup/Lower overhead
• More expensive per MB
On Demand DDoS Mitigation

“Um, Hi! This is the scrubbing service, We think you’re under attack”

“Yep, we’re under attack and we’re DOWN! Start Scrubbing!!”
On Demand DDoS Mitigation

All Traffic Redirected

"Scrubbing!!"

"Yeah!!, We're not down anymore"

Response / Return Traffic Asymmetric

"Yes, but you only have "blah" Capacity left, “N” Number of Incidents or time Left - Pay us MORE!"

"You tell me? Is the attack still happening?..."
Always On DDoS Mitigation

Enforcement Points

Symmetrical Traffic Flows
Questions?
Thank you!

krassi@fastly.com
Good Internet citizenship
Mitigations

• Defend yourself
  – Anycast
  – Some form of IPS/DDoS mitigation gear
  – Overall network architecture

• Defend the Internet
  – Rate-limiting
  – BCP38/140 (outbound filtering) source address validation
  – Securely configured DNS, NTP and SNMP servers
  – No open resolvers

• Talk to the professionals
Are you noticing the imbalance?

• Defend yourself
  - Anycast (DNS)
  - Some form of IPS/DDoS mitigation gear

Defend the Internet
  - Rate-limiting
  - BCP38/140 (outbound filtering) source address validation
  - Securely configured authoritative DNS servers
  - No open resolvers

• Lots of money
• Somewhat cheap
What’s the point I’m trying to make?

• It’s not feasible to mitigate those attacks single handedly
• We need cooperation
• Companies need to start including “defending the Internet from themselves” as a part of their budget – not only “defending themselves from the Internet”
What can I do about it?

- RFC 2827/BCP 38 – Paul Ferguson
- If possible filter all outgoing traffic and use proxy
- uRPF

- BCP 140: “Preventing Use of Recursive Nameservers in Reflector Attacks”
- Aka RFC 5358
Resources

- DNS

- NTP

- If you see your IP space in the lists provided by those sites – resolve it
Summary

- Discuss what DDoS is, general concepts, adversaries, etc.
- Went through a networking technology overview, in particular the OSI layers, sockets and their states, tools to inquire system state or capture and review network traffic
- Dove into specifics what attack surface the different layers offer
- Discussed different attack types
- Terminology
- Tools
Thank you!

krassi@fastly.com