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?

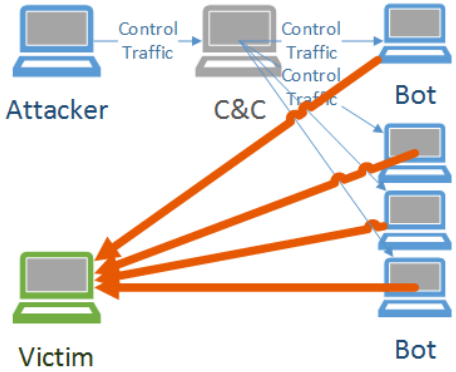
- Financial losses

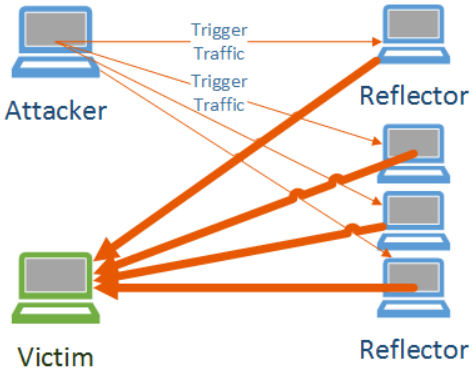
DoS vs. DDoS

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

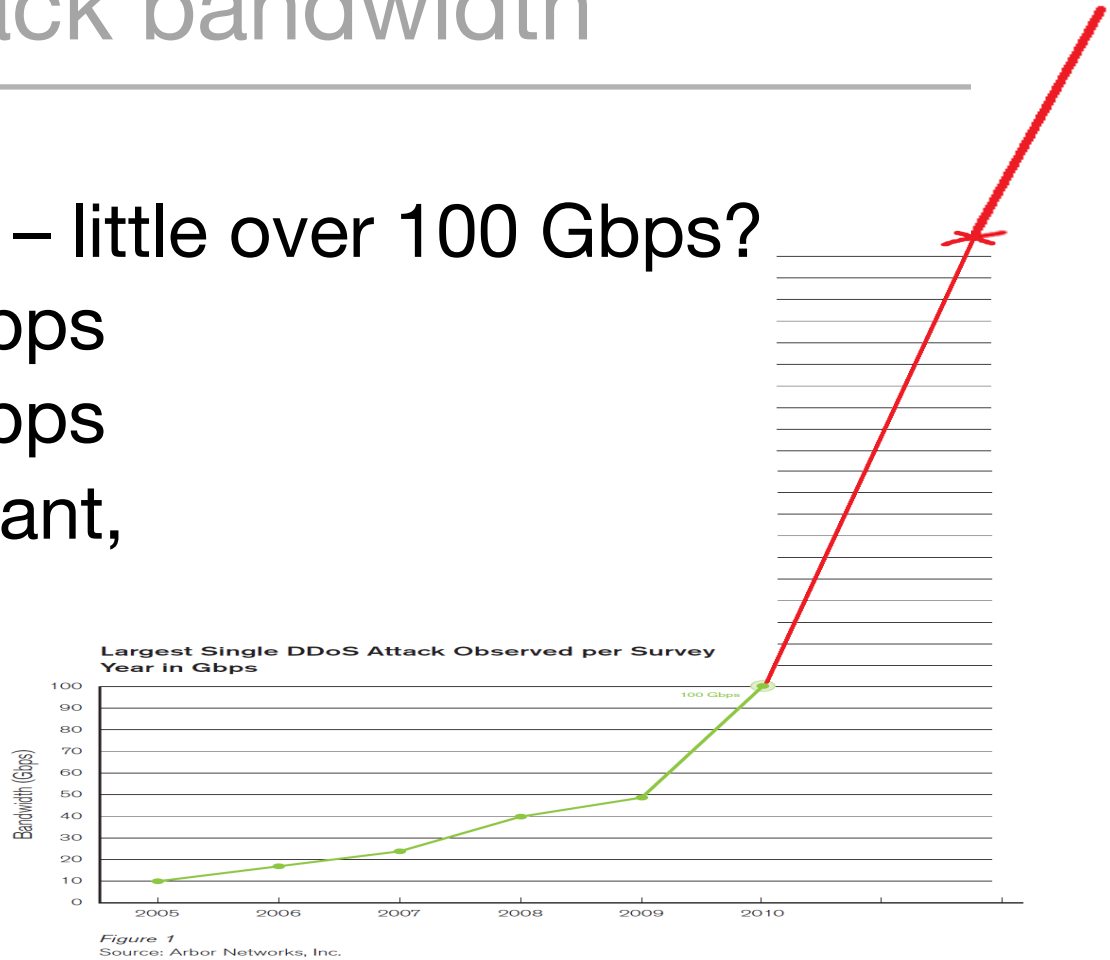




The problem?

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- Outdated Content Management Systems (CMSes)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

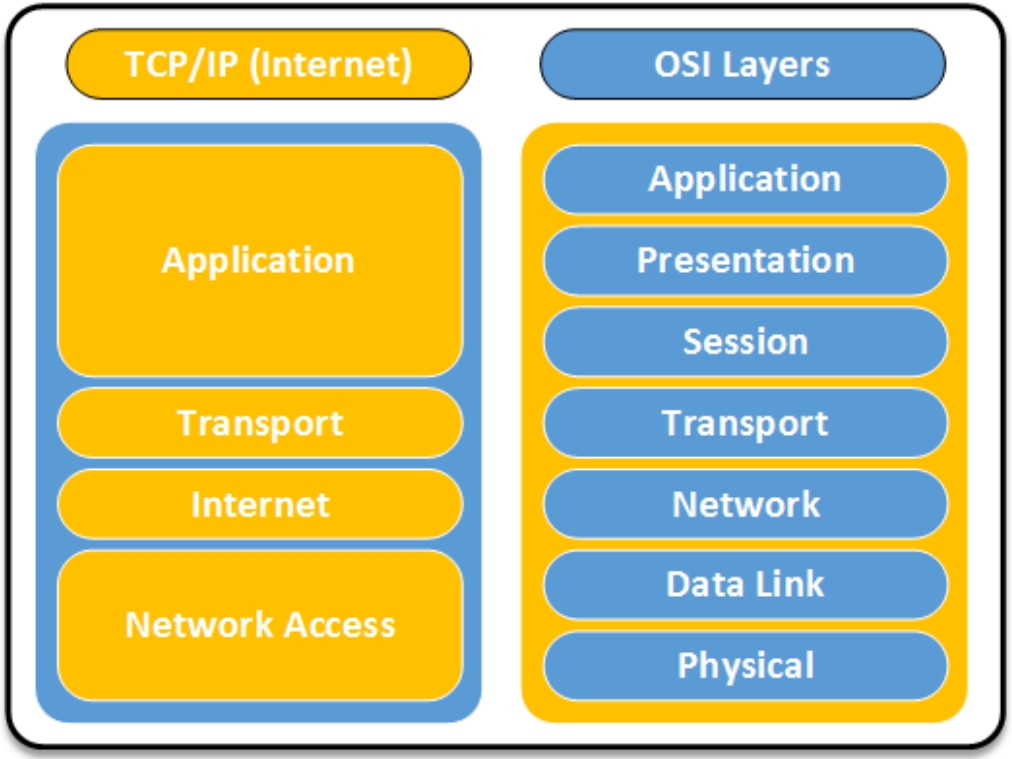
The purpose of this section is to level set

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



Physical and Data-link Layers (L1 and L2)

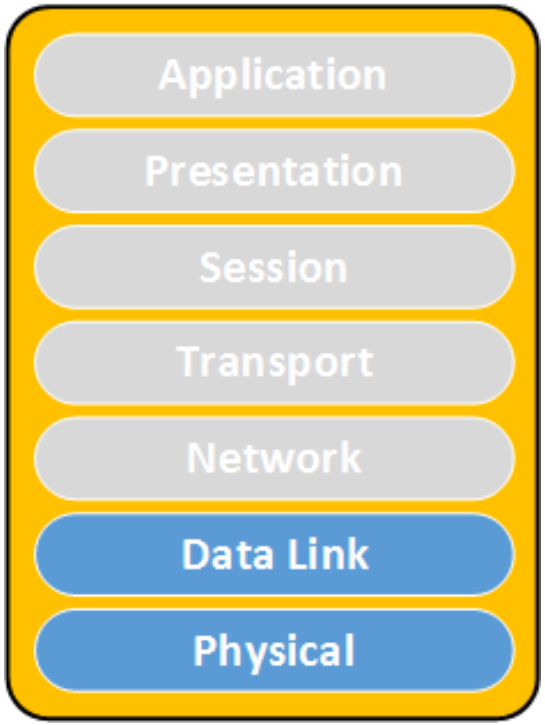
Aka: Network Access Layer

Physical

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

Data-link

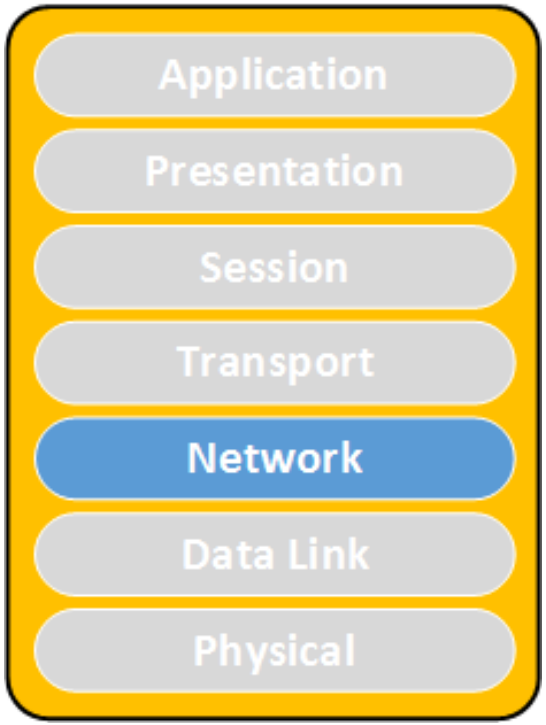
Data unit: frame (organizes bits in a frame)
Provides physical addressing on a local network segment
Separate in two:
Media Access Control Layer (MAC): 802.3, 802.4, 802.5, 802.11abe
Logical-link Control: 802.2



Network Layer (layer 3)

Aka Internet Layer

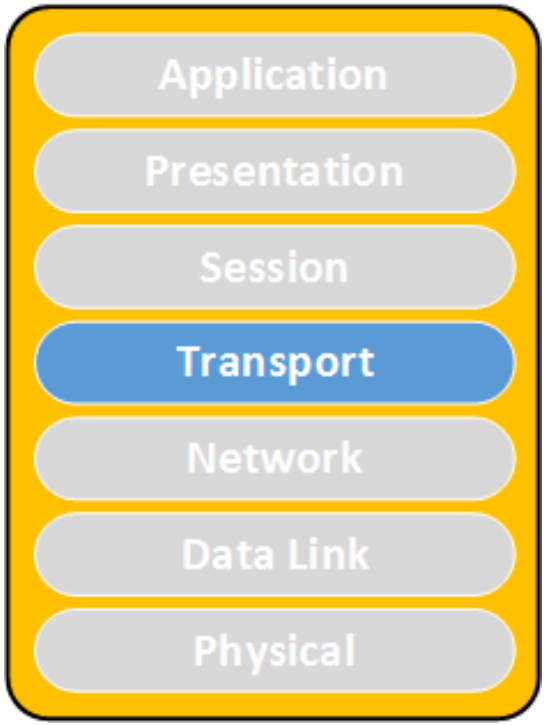
Provides transport of data units between two points in the network
Provides logical (and global) addressing of machines in the network
Data unit: Packet
Examples: Internet Protocol (IP)
Does not guarantee delivery
Allows for fragmentation



Transport Layer (layer 4)

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

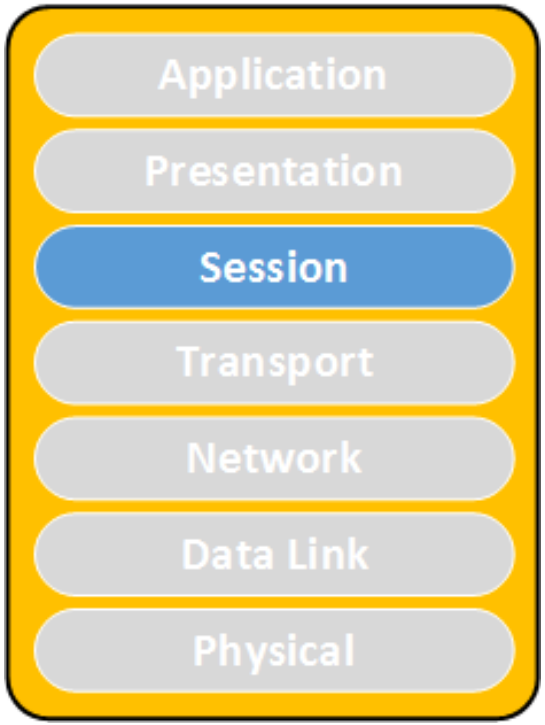
Provides logical connection between applications
Provides addressing of applications on a single system (via port numbers)
Data unit: segment
In some modalities like TCP provides virtual circuit and ensures data ordering and no loss of packets
Typical for TCP is the 3-way handshake
Examples: TCP, UDP



Session Layer (layer 5)

Part of the Internet model Application layer

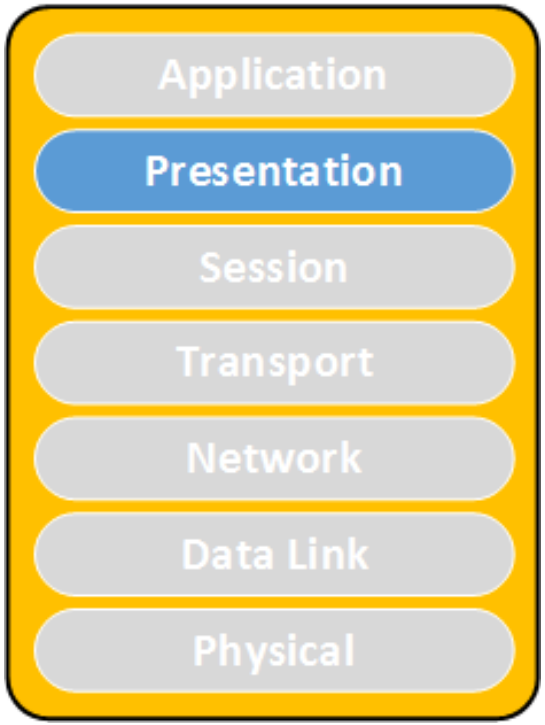
Managing sessions between application (think state, like authentication)
Examples: HTTP, SMTP, NetBIOS
Addressing: some protocols provide logical endpoint



Presentation Layer (layer 6)

Part of the Internet model Application layer

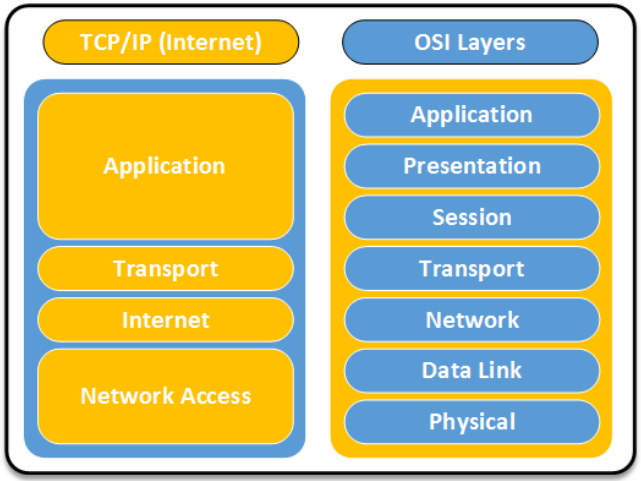
Provides uniform data representation across multiple architectures and platforms
Examples: images, file encryption

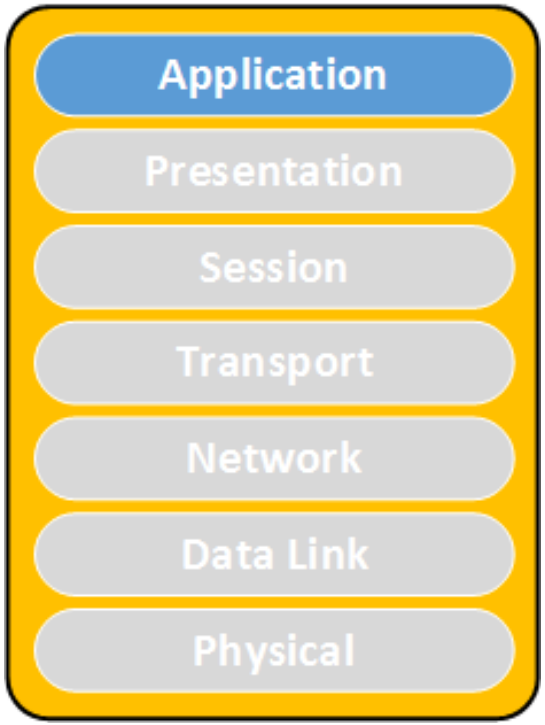


Application Layer (layer 7)

This is where the application lives

Part of the Internet model
Application layer

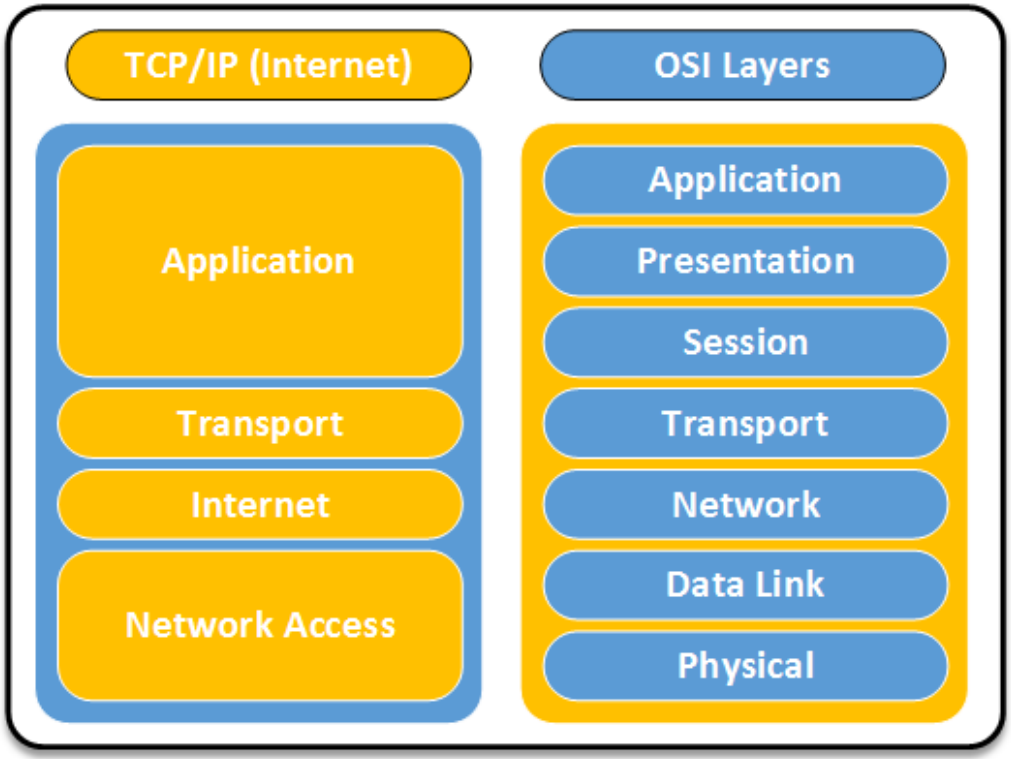




Questions?

Attack surface

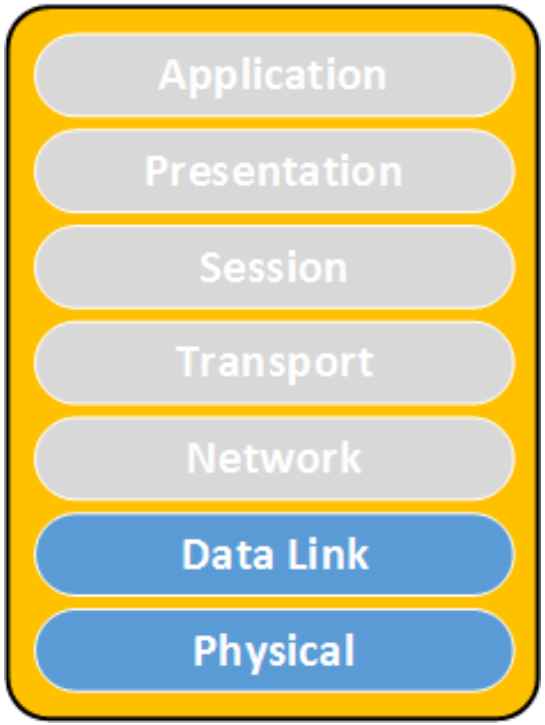
Network Layers – OSI vs Internet Model



Physical and Data-link Layers

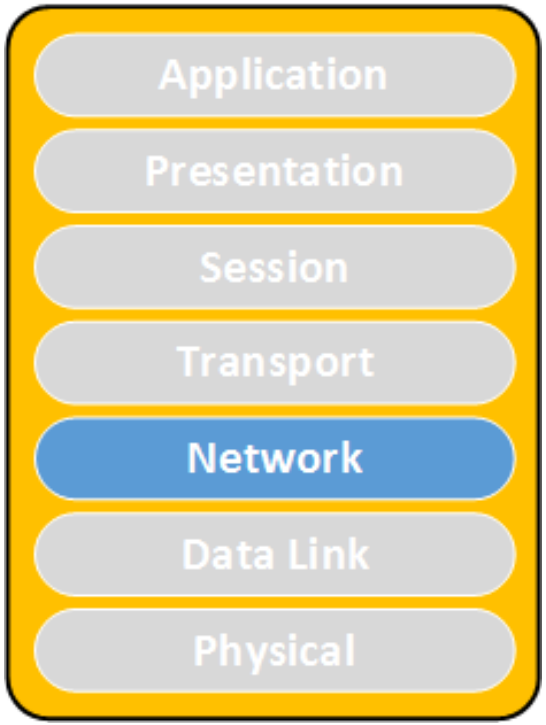
- Cut cables
- Jamming
- Power surge
- EMP

- MAC Spoofing
- MAC flood



Network Layer

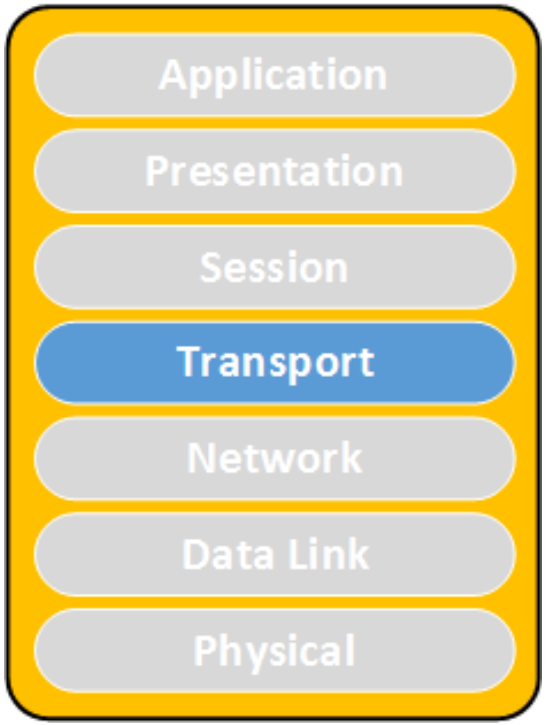
- Floods (ICMP)
- Teardrop
  (overlapping IP segments)



Transport Layer

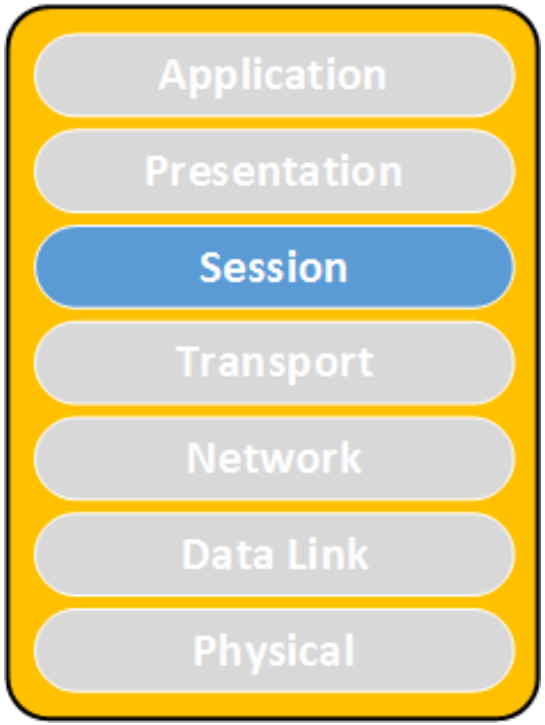
- SYN Flood
- RST Flood
- FIN Flood
- You name it…

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



Session Layer

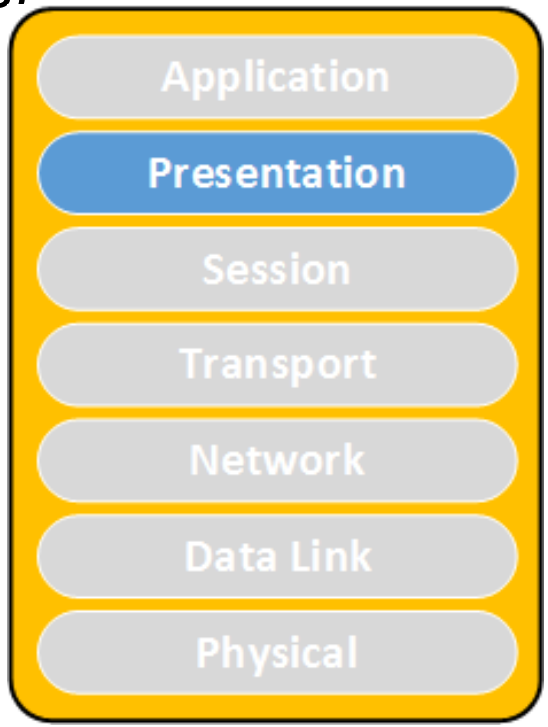
- Slowloris
- Sending data to a port with no NL in it (long headers, long request lines)
- Send data to the server with no CR



Presentation Layer

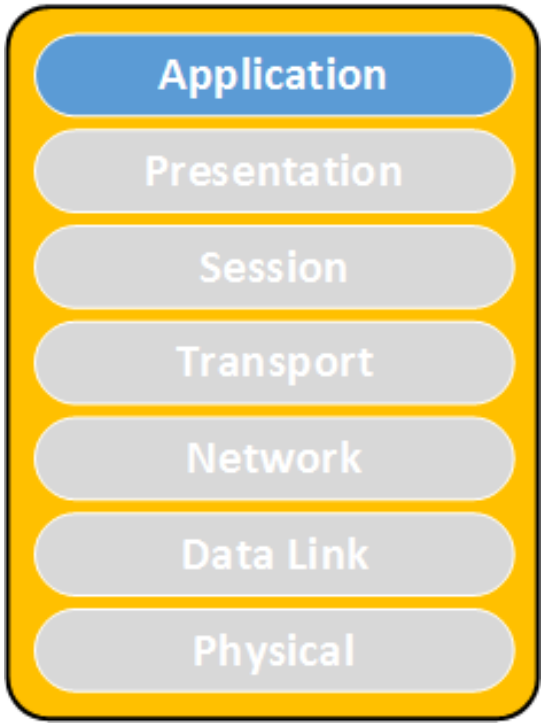
- Expensive queries (repeated many times)
- XML Attacks

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



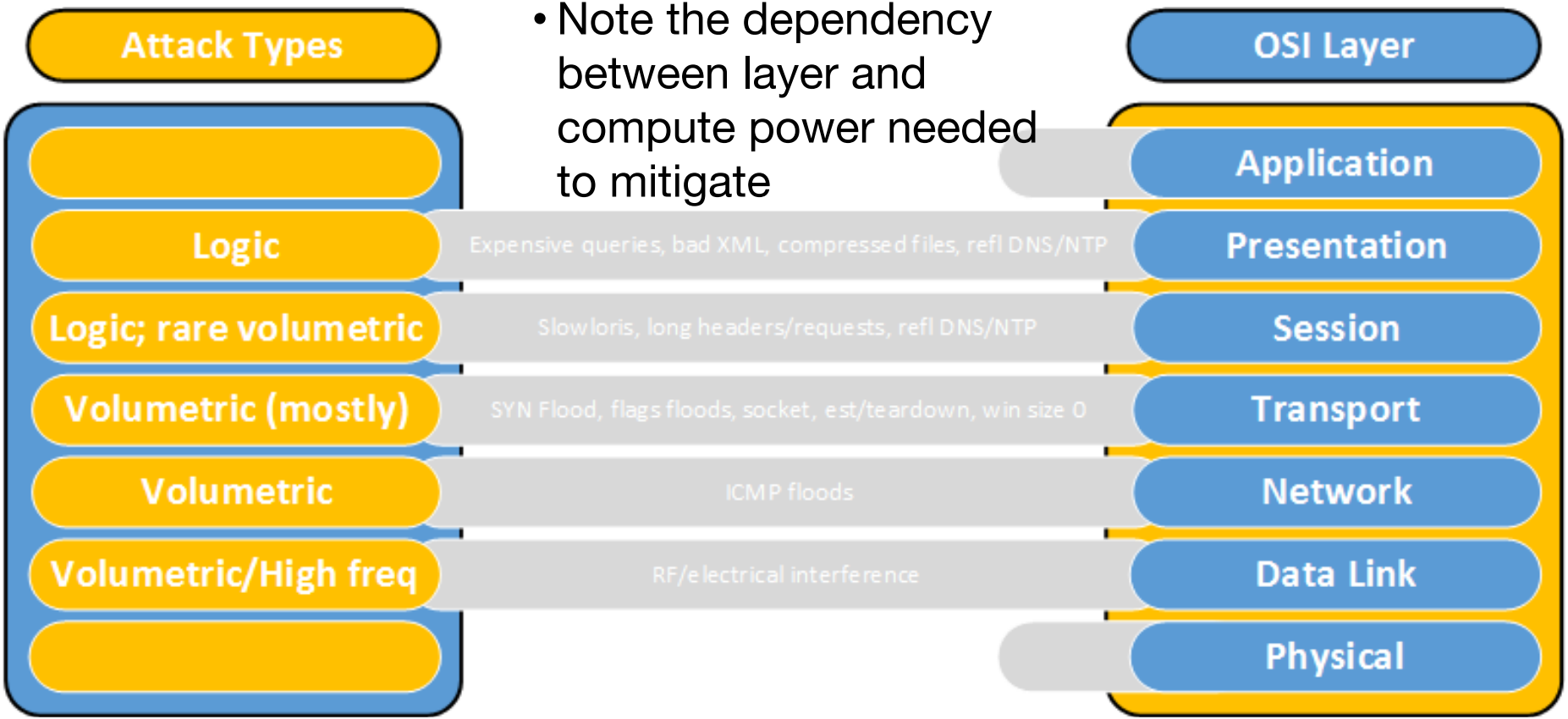
Application Layer

- Depends on the application
- Black fax



Attack summary by layer

- Note the dependency between layer and compute power needed to mitigate



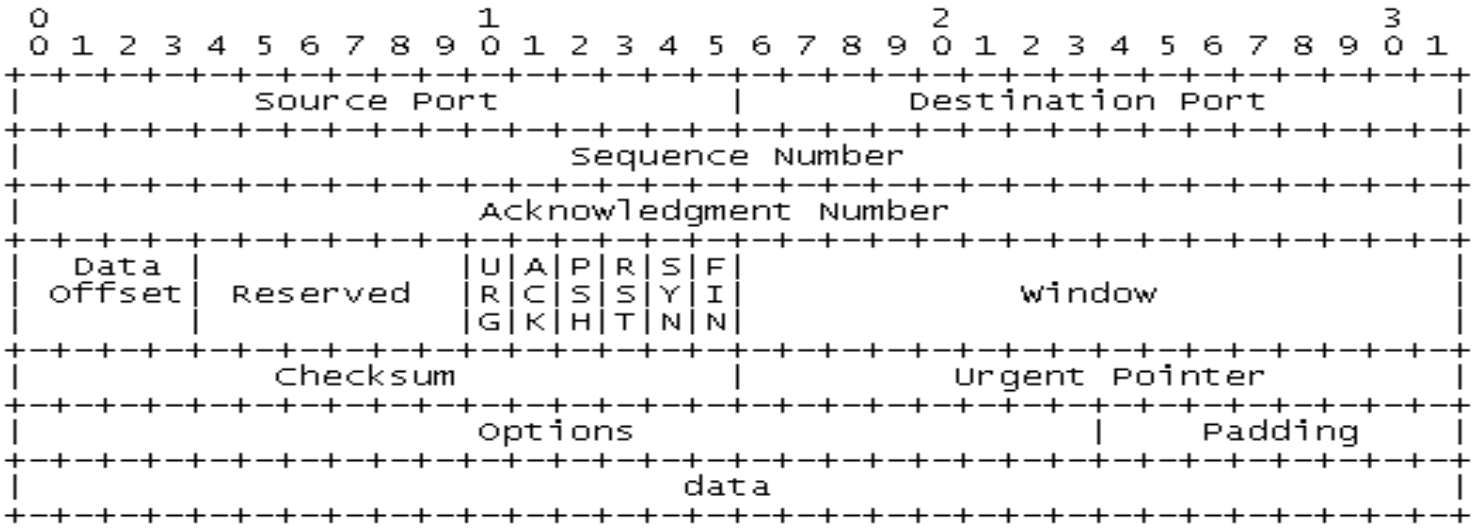
Questions?

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



Sockets

Socket is an abstraction allowing an application to bind to a transport layer address (aka network port)

It is described by a state machine

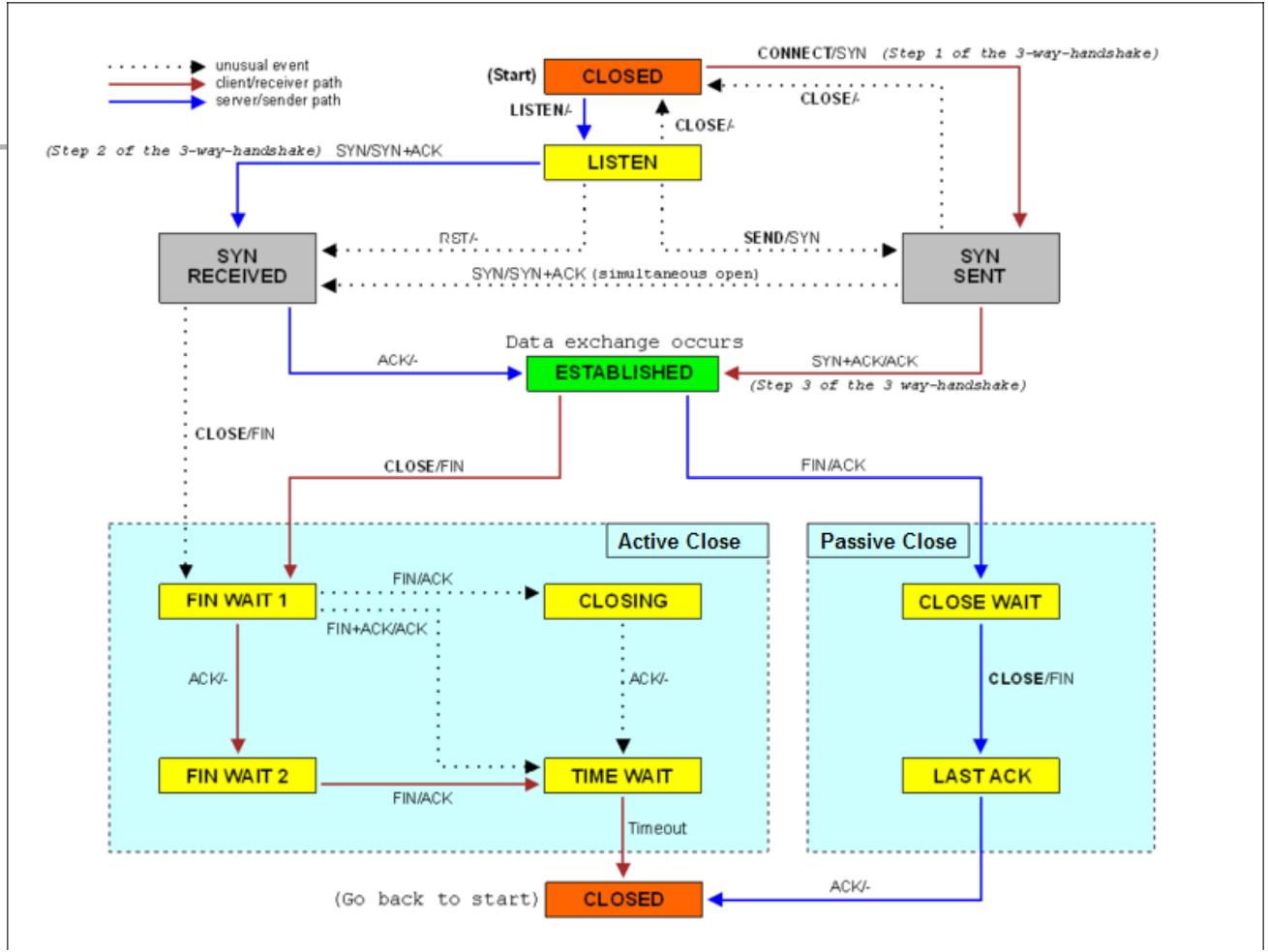
Throughout its life time it goes through a number of states

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

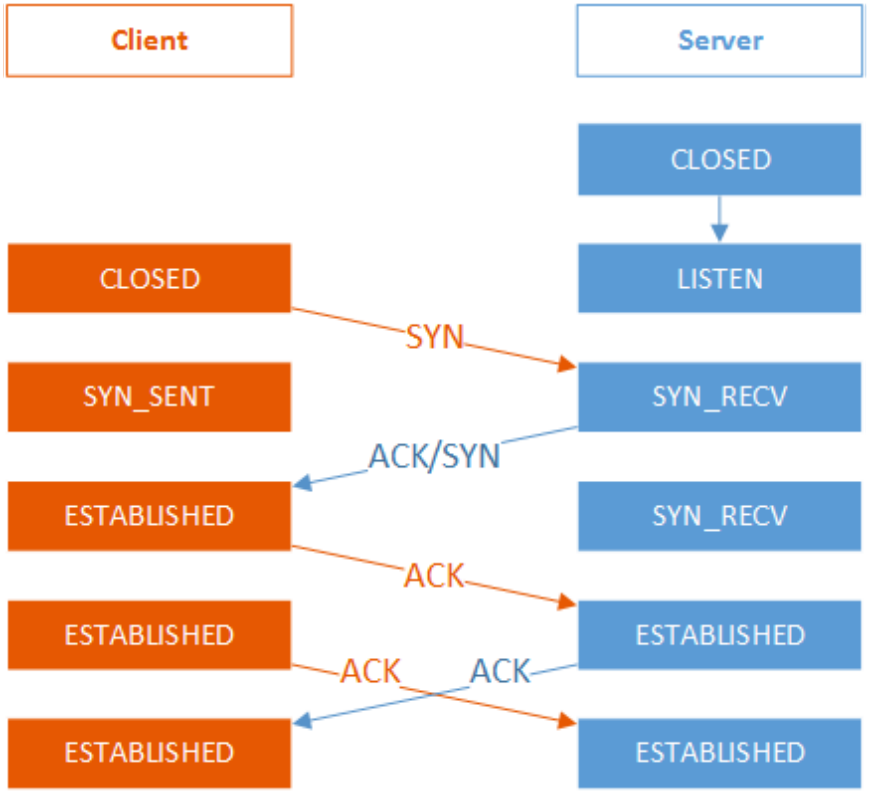
Diagram


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
LISTEN
RST/-
SEND/SYN
SYN RECEIVED
SYN/SYN+ACK (simultaneous open)
SYN SENT
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3 way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active Close
Passive Close
FIN WAIT 1
FIN/ACK
CLOSING
CLOSE WAIT
FIN+ACK/ACK
ACK/-
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
(Go back to start)
CLOSED
ACK/-

Opening a TCP connection

Let's review the sequence for opening a connection

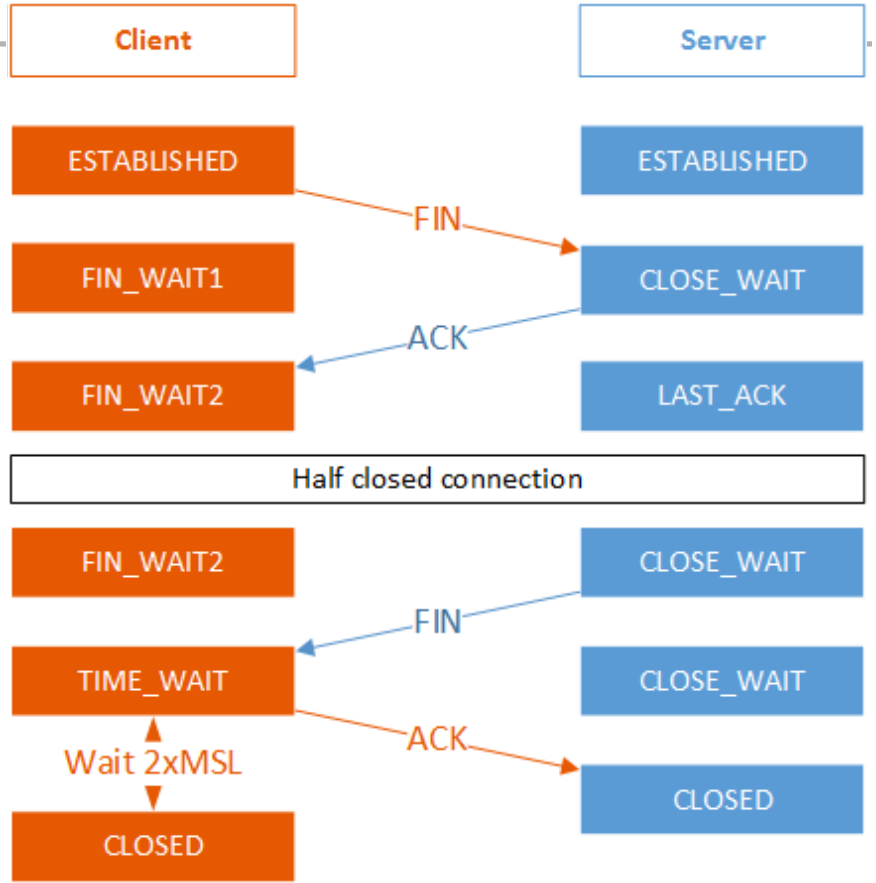
- Server side opens a port by changing to LISTEN state
- Client sends a SYN packet and changes state to SYN_SENT
- Server responds with SYN/ACK and changes state to SYN_RECV. For the client this is ESTABLISHED connection
- Client has to ACK and this completes the handshake for the server
- Packet exchange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



Closing a TCP connection

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

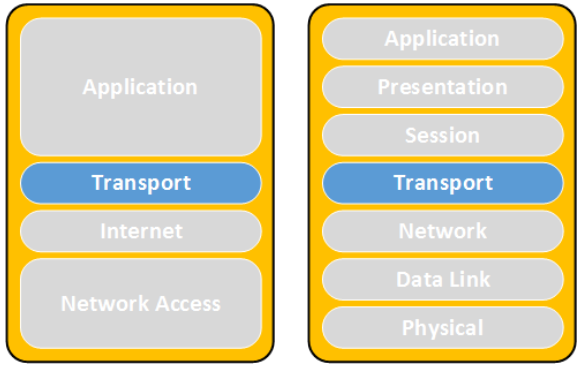
- Source
  - Single source
  - Distributed
- State change
  - Permanent
  - Recoverable

- Based on network layer

Important metrics and what to report

- Bandwidth (Kbps, Gbps)
- Latency
- PPS
- QPS
- Storage
- CPU
- Application specific – usually latency
- Protocol

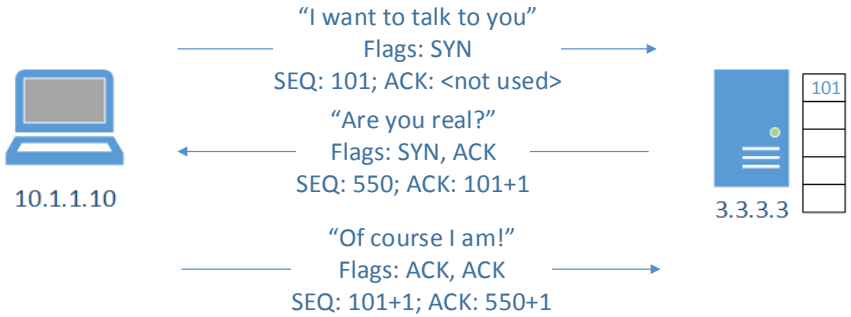
Attack type details


Application
Transport
Internet
Network Access
Application
Presentation
Session
Transport
Network
Data Link
Physical

SYN Flood

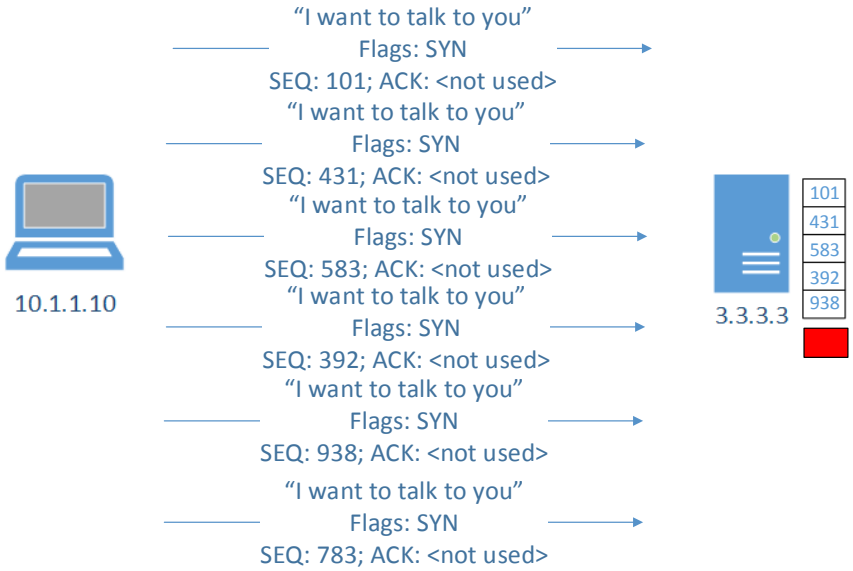
What is a SYN flood?

What is a 3-way handshake?



SYN flood

- Exploits the limited slots for pending connections
- Overloads them



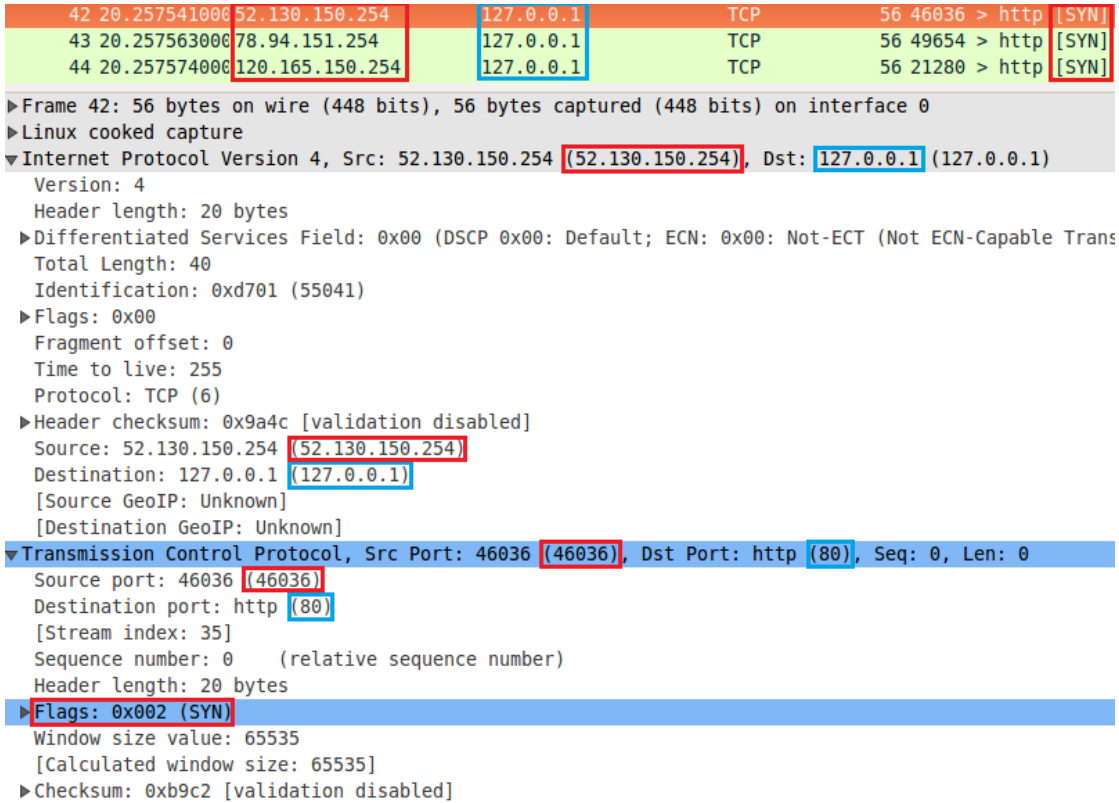
SYN flood through the eyes of netstat

netstat –anp

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	1339/rpcbind
tcp	0	0	0.0.0.0:33586	0.0.0.0:*	LISTEN	1395/rpc.statd
tcp	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN	1962/dnsmasq
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1586/cupsd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	2703/sendmail: acce
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49718**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49717**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49722**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49720**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49719**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49721**	**SYN_RECV**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:49716**	**SYN_RECV**	**-**

SYN on the wire



Attacker
Random IP address/ port

Target
127.0.0.1:80

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:

Timestamp % 32 + MSS + 24-bit hash

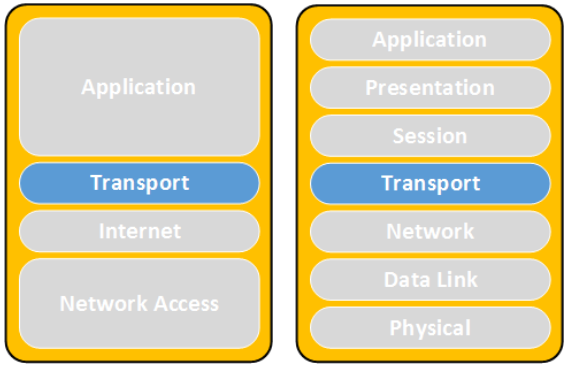
Components of 24-bit hash:
server IP address
server port number
client IP address
client port
timestamp >> 6 (64 sec resolution)

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 observed via netstat

Socket exhaustion would look like this:

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	1339/rpcbind
tcp	0	0	0.0.0.0:33586	0.0.0.0:*	LISTEN	1395/rpc.statd
tcp	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN	1962/dnsmasq
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1586/cupsd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	2703/sendmail: acce
tcp	0	0	0.0.0.0:1241	0.0.0.0:*	LISTEN	1851/nessusd: waiti
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60365**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60240**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60861**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60483**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60265**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60618**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60407**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60423**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60211**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60467**	**TIME_WAIT**	**-**
tcp	**0**	**0**	**127.0.0.1:25**	**127.0.0.1:60213**	**TIME_WAIT**	**-**

How to enable socket reuse (IoT issue)

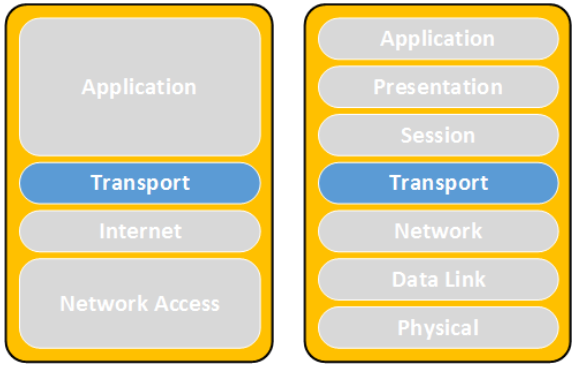
- How to determine

  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


Application
Transport
Internet
Network Access
Application
Presentation
Session
Transport
Network
Data Link
Physical

Slowloris

Connection handling architectures

Process based connection handling?

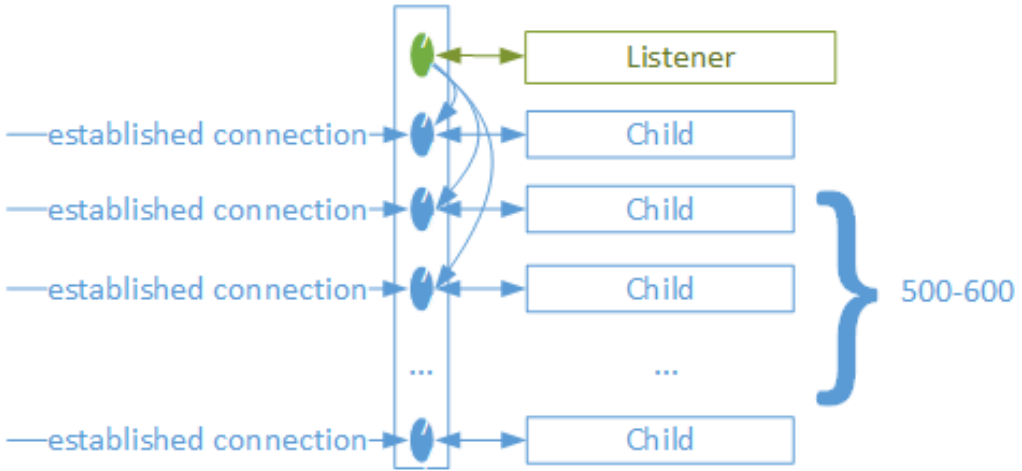
Think “Apache”

Event based connection handling?

Think “nginx”

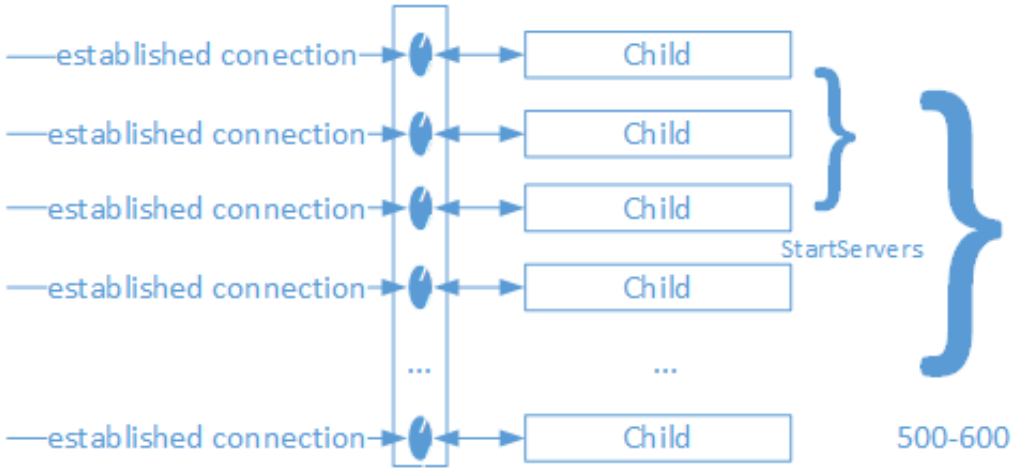
Process oriented explained

- 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



Apache 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

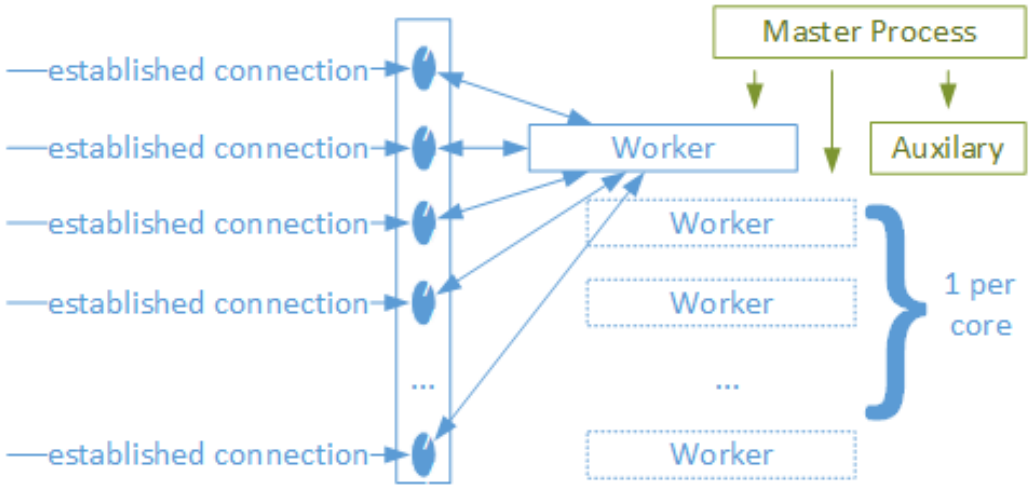


Nginx (simplified)

- 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

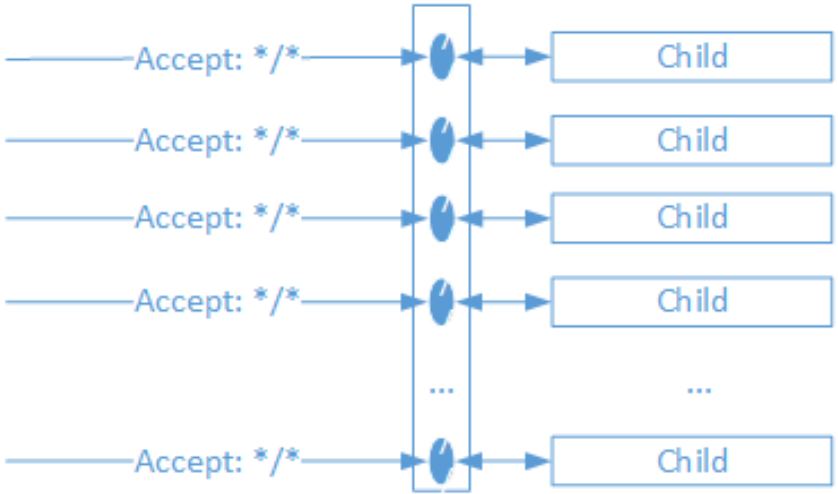
Slowloris request

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

Slowloris illustrated

- 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 the an unwilling intermediary is used to deliver the attack traffic

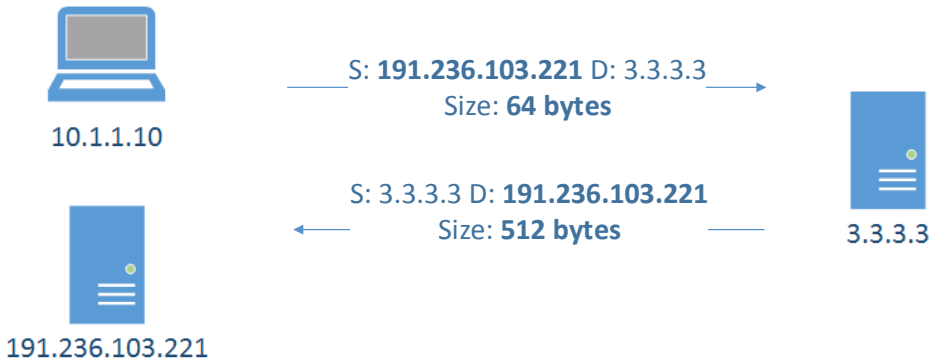
- The attacker would normally send a packet with a forged source IP address to the intermediary. The forget address is going to be the one of the target. The intermediary will deliver a response which will go to the target instead of the attacker

Note to audience: think what protocols we can use for that?

What is reflection(ed) attack

Attacks where the 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



Reflector types

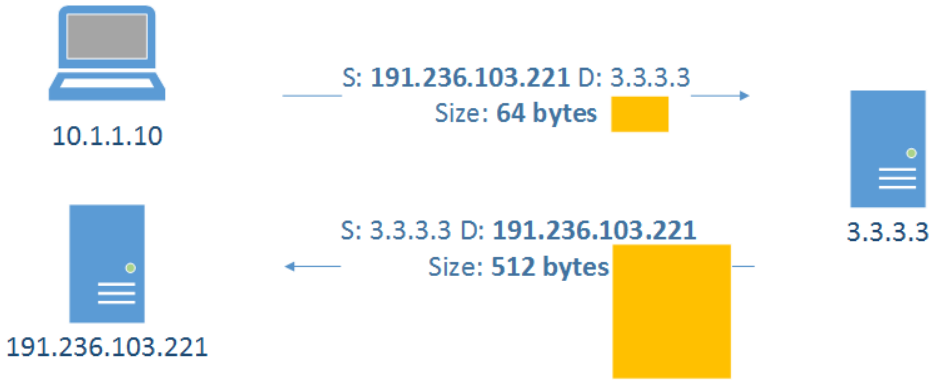
The ones that are of interest are:

- 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:

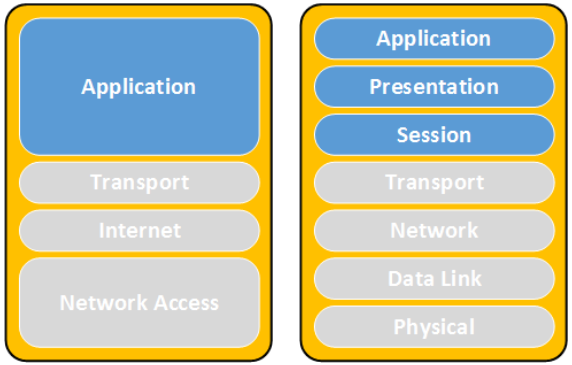
https://www.us-cert.gov/ncas/alerts/TA14-017A

Amplification quotients

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

- Source: US-CERT: https://www.us-cert.gov/ncas/alerts/TA14-017A

Questions?



DNS Resolution

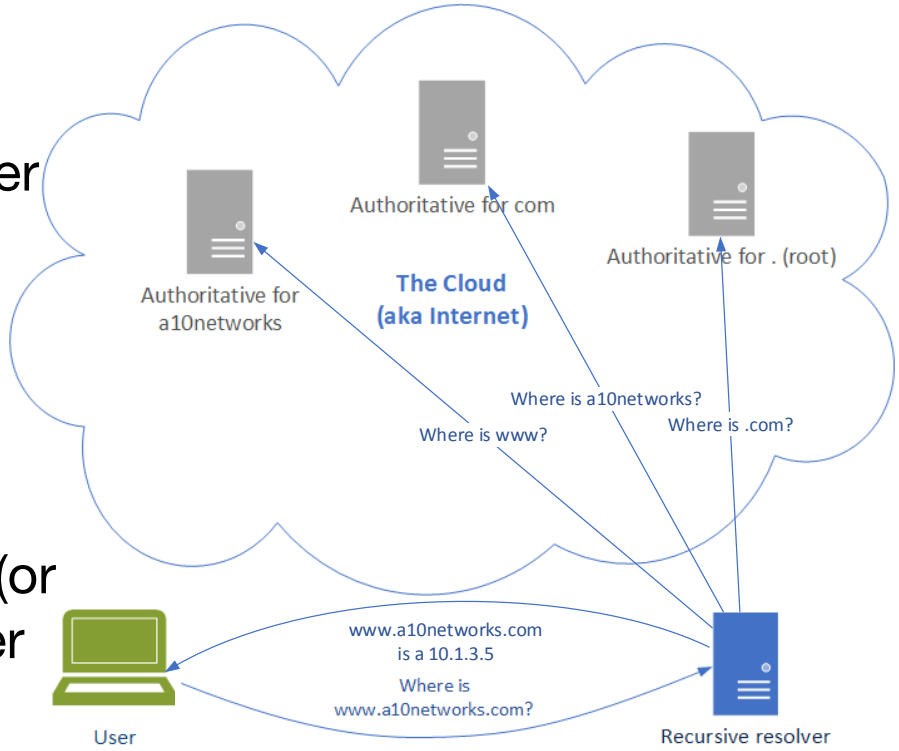
DNS server types

- 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

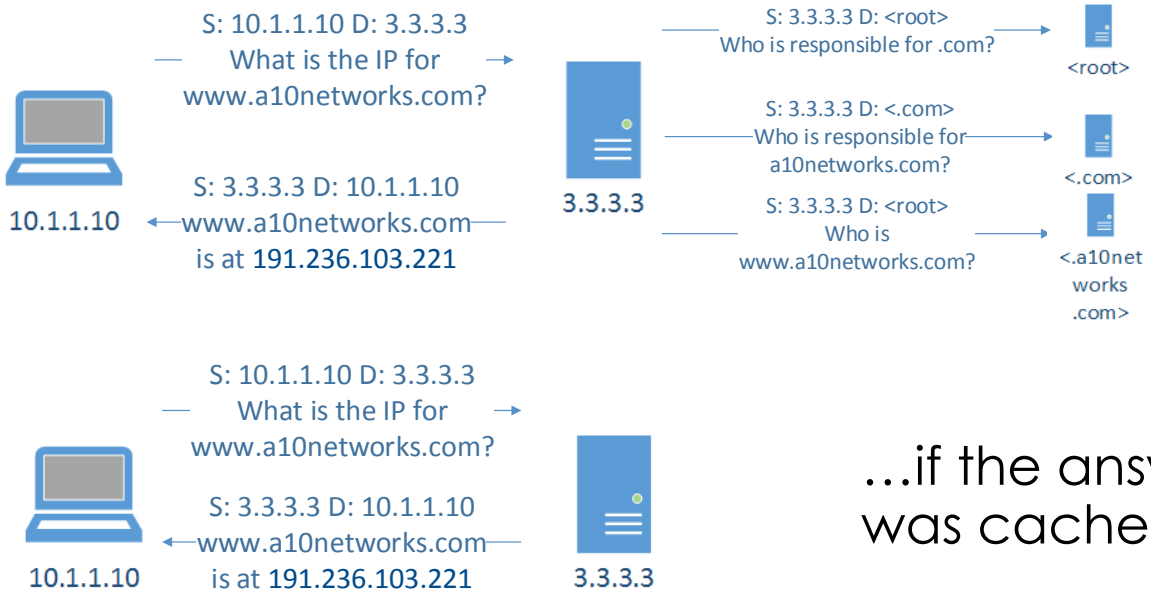
DNS resolution

- 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

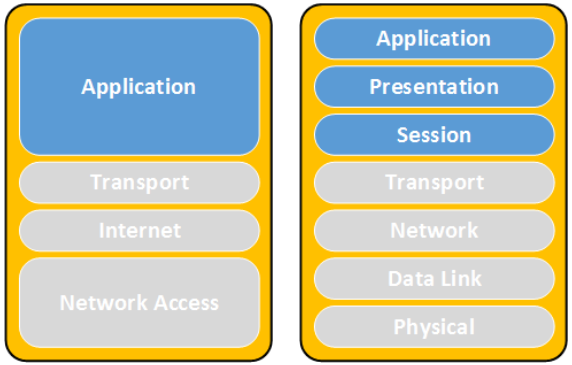


DNS resolution at the packet level

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



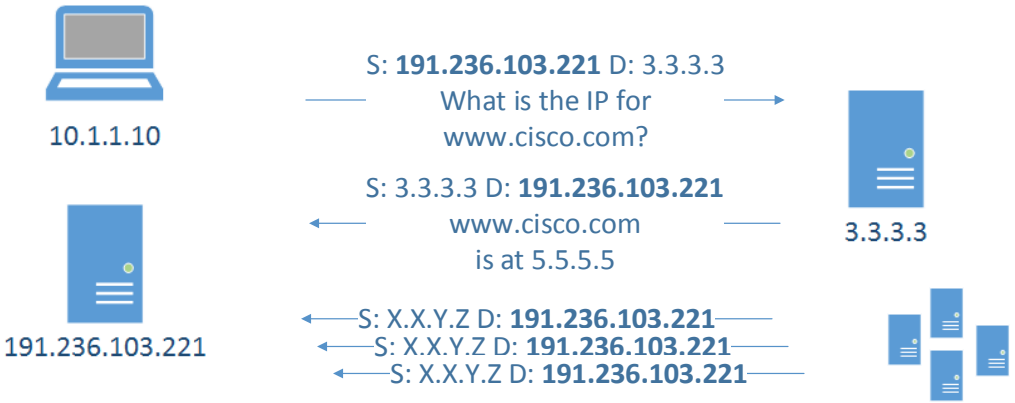
…if the answer was cached



DNS Reflection

What is DNS reflection attack?

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



…the reflected traffic goes to the target server

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

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

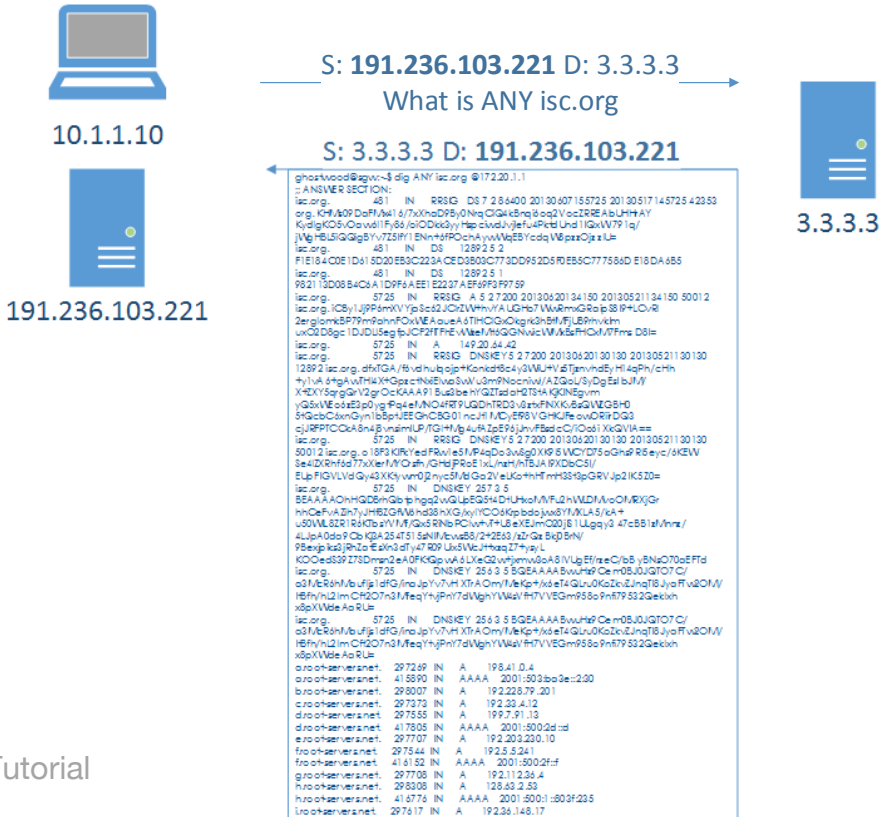
▪ Flip over for answer

Consider this (cont'd)

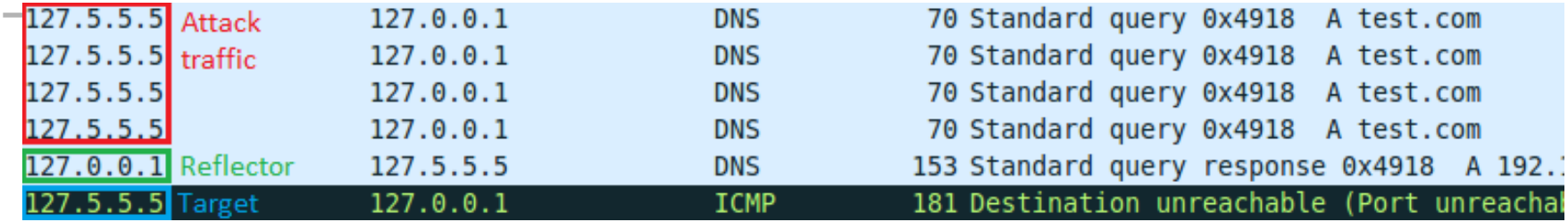
```
ghostwood@sgw:~$ dig ANY isc.org @172.20.1.1
;; ANSWER SECTION:
isc.org.        481     IN      RRSIG   DS 7 2 86400 20130607155725 20130517145725 42353 org. KHMs09DaFMx416/7xXhaD9By0NrqCiQ4kBnqi6oq2VocZRREAbUHHrAY
KydlgKO5vOaw6l1Fy86/oiODkk3yyHspciwdJvjlefu4PktdUnd1IQxW 791q/jWgHBL5iQQigBYv7Z5IfY1ENn+6fPOchAywWqEBYcdqW8pzzOjz zlU=
isc.org.        481     IN      DS      12892 5 2 F1E184C0E1D615D20EB3C223ACED3B03C773DD952D5F0EB5C777586D E18DA6B5
isc.org.        481     IN      DS      12892 5 1 982113D08B4C6A1D9F6AEE1E2237AEF69F3F9759
isc.org.        5725    IN      RRSIG   A 5 2 7200 20130620134150 20130521134150 50012 isc.org. iCBy1Jj9P6mXVYjaSc62JClrZW+hvYAUGHo7WwRmxGRaipS8I9+LCvRl
2erglomkBP79m9ahnFOxWEAaueA6TlHClGxOkgrk3hBtMFjUB9rhvkIm uxO2D8gc1DJDLl5egfpJCF2flTFhEvWzeMt6QGNwicWMxBsFHCxM7Fms 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/f6vdhuIqojp+Konkdt8c4y3WiU+Vs5TjznvhdEyH14qPh/cHh
+y1vA6+gAwTHl4X+GpzctNxiElwaSwVu3m9Nocniwl/AZQoL/SyDgEsI bJM/X+ZXY5qrgQrV2grOcKAAA91Bus3behYQZTsdaH2TStAKjKINEgvm
yQ5xWEo6zE3p0ygtPq4eMNO4fRT9UQDhTRD3v3ztxFlNXKvBsQWZGBH0 5tQcbC6xnGyn1bBptJEEGhCBG01ncJt1MCyEf98VGHKJFeowORiirDQ3
cjJRFPTCCkA8n4j8vnsimIUP/TGl+Mg4ufAZpE96jJnvFBsdcC/iOo6i XkQVIA==
isc.org.        5725    IN      RRSIG   DNSKEY 5 2 7200 20130620130130 20130521130130 50012 isc.org. o18F3KIFkYedFRw1e5MP4qDo3wSg0XK9l5WCYD75aGhs9Rl5eyc/
6KEW Se4lZXRhf6d77xXlerMYCrsfh/GHdjPRoE1xL/nzH/hTBJAI9XDbC5I/ EUpFlGVLVdQy43XKtywm0j2nyc5MdGa2VeLKo+hHTmH3St3pGRVJp2IK 5Z0=
isc.org.        5725    IN      DNSKEY  257 3 5 BEAAAAOhHQDBrhQbtphgq2wQUpEQ5t4DtUHxoMVFu2hWLDMvoOMRXjGr hhCeFvAZih7yJHf8ZGfW6hd38hXG/
xylYCO6Krpbdojwx8YMXLA5/kA+ u50WIL8ZR1R6KTbsYVMf/Qx5RiNbPClw+vT+U8eXEJmO20jIS1ULgqy3 47cBB1zMnnz/4LJpA0da9CbKj3A254T515sNIMcwsB8/2+2E63/
zZrQz Bkj0BrN/9Bexjpiks3jRhZatEsXn3dTy47R09Uix5WcJt+xzqZ7+ysyL KOOedS39Z7SDmsn2eA0FKtQpwA6LXeG2w+jxmw3oA8lVUgEf/rzeC/bB yBNsO70aEFTd
isc.org.        5725    IN      DNSKEY  256 3 5 BQEAAAABwuHz9Cem0BJ0JQTO7C/a3McR6hMaufljs1dfG/inaJpYv7vH XTrAOm/MeKp+/
x6eT4QLru0KoZkvZJnqTl8JyaFTw2OM/ltBfh/hL2lm Cft2O7n3MfeqYtvjPnY7dWghYW4sVfH7VVEGm958o9nfi79532Qeklxh x8pXWdeAaRU=


a.root-servers.net.     297269  IN      A       198.41.0.4
a.root-servers.net.     415890  IN      AAAA    2001:503:ba3e::2:30
b.root-servers.net.     298007  IN      A       192.228.79.201
c.root-servers.net.     297373  IN      A       192.33.4.12
d.root-servers.net.     297555  IN      A       199.7.91.13
d.root-servers.net.     417805  IN      AAAA    2001:500:2d::d
e.root-servers.net.     297707  IN      A       192.203.230.10
f.root-servers.net.     297544  IN      A       192.5.5.241
f.root-servers.net.     416152  IN      AAAA    2001:500:2f::f
```

Reflection and Amplification



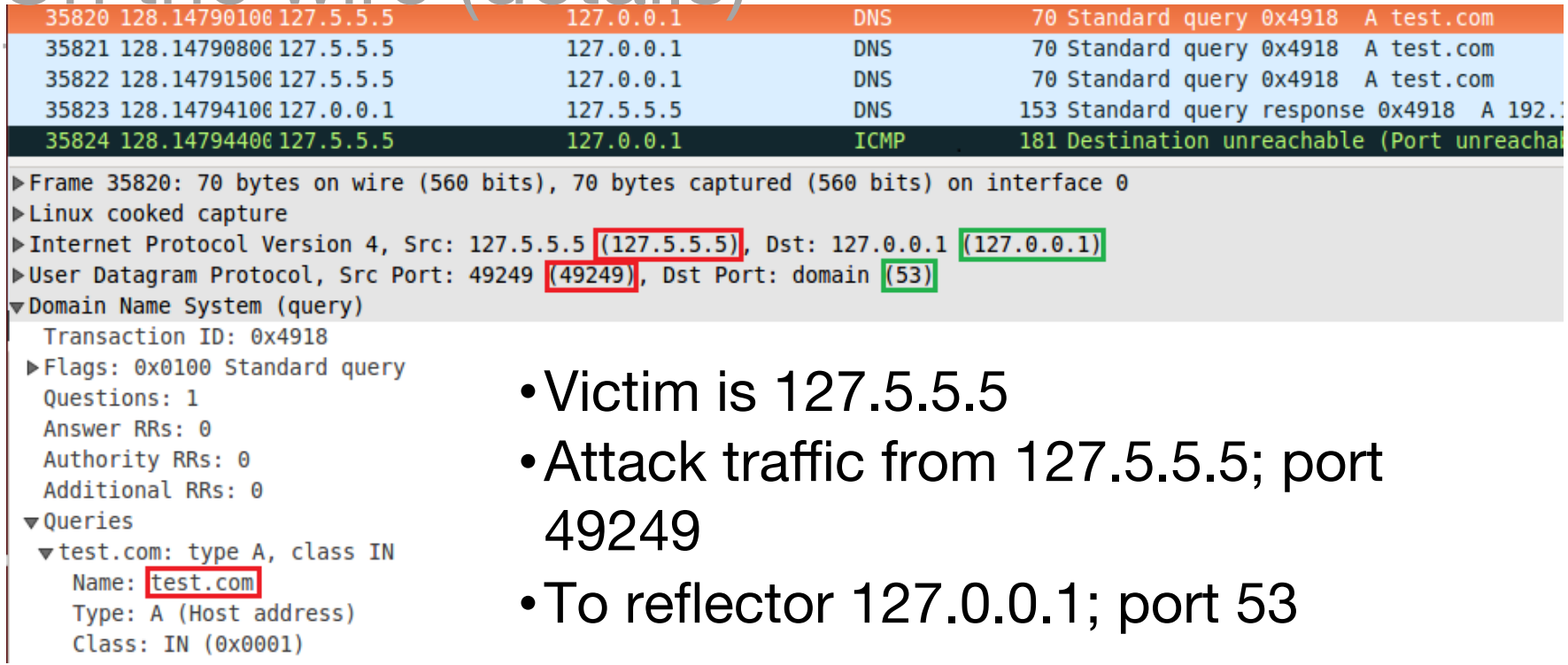
On the wire



- 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

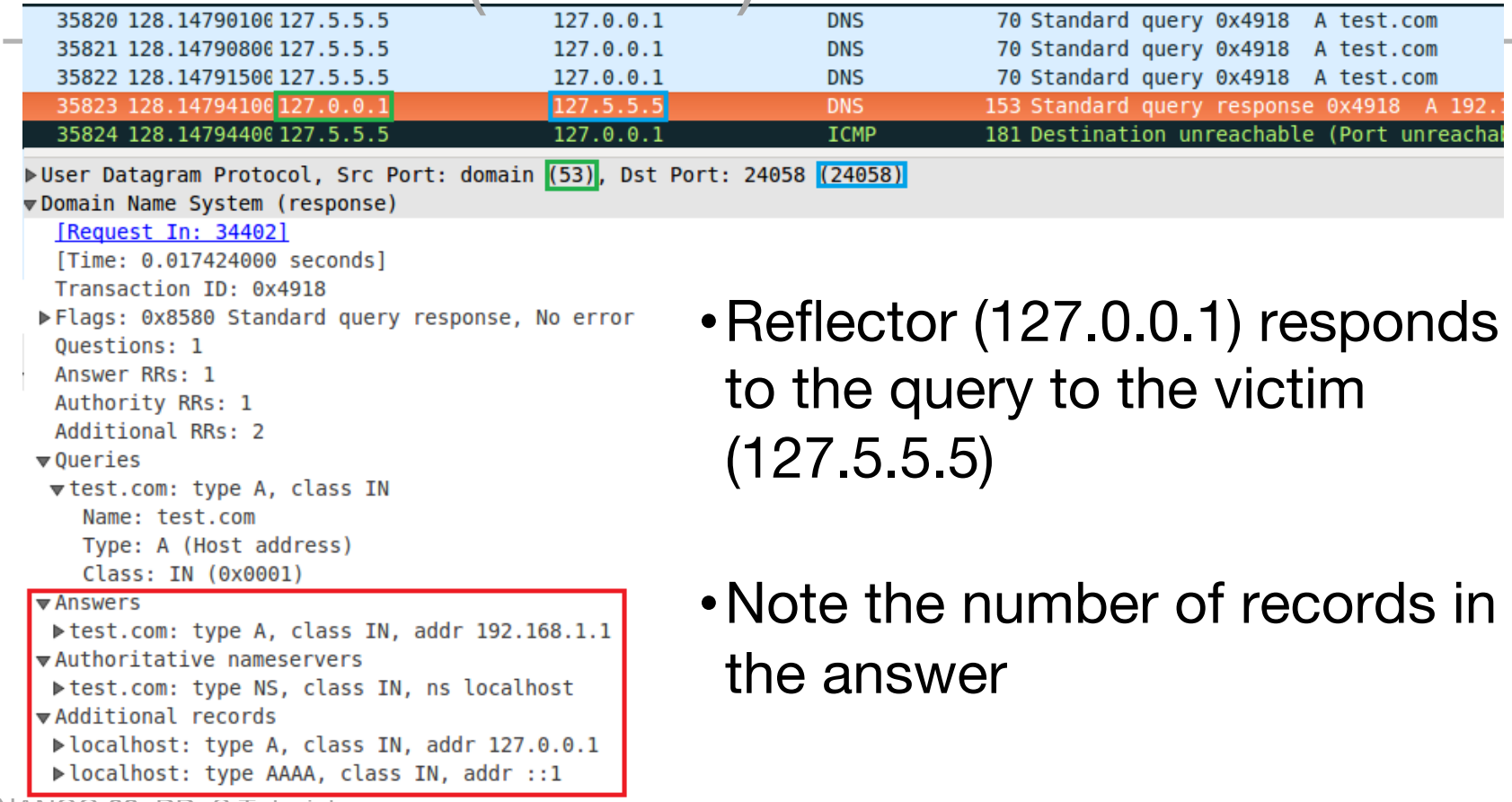
- BACK SCATTER
  Notice the victim is responding with port unreachable because there is nothing running on that UDP port. This is called back-scatter

On the wire (details)

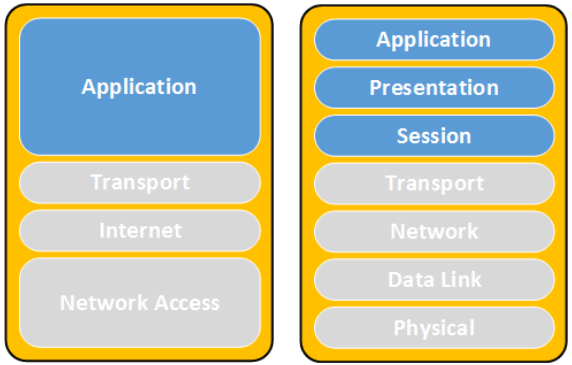


- Victim is 127.5.5.5
- Attack traffic from 127.5.5.5; port 49249
- To reflector 127.0.0.1; port 53

On the wire (details)



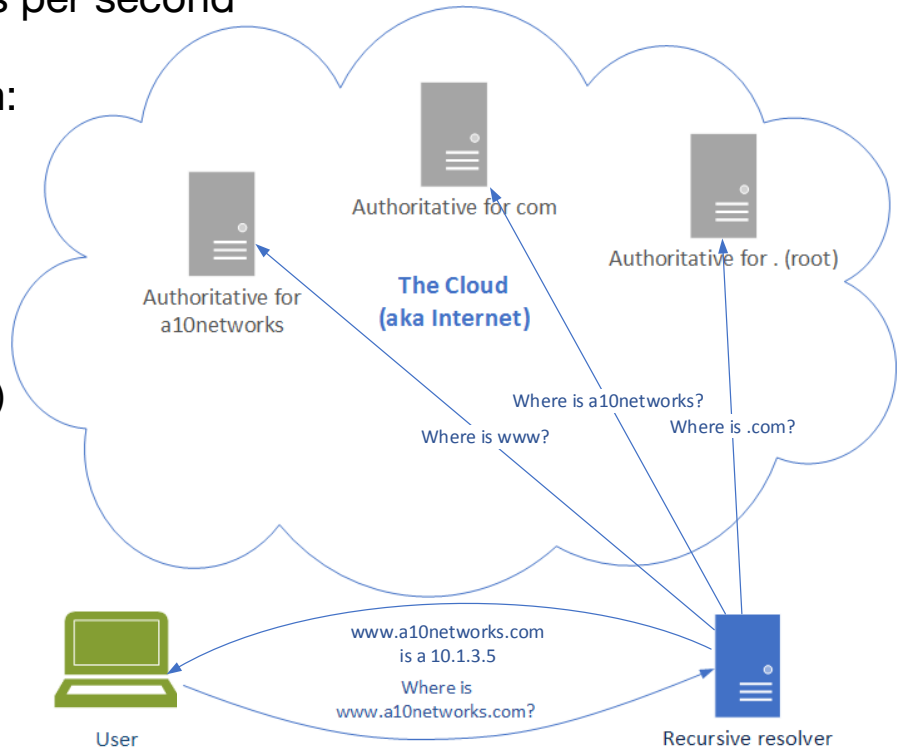
- Reflector (127.0.0.1) responds to the query to the victim (127.5.5.5)
- Note the number of records in the answer



Cache busting

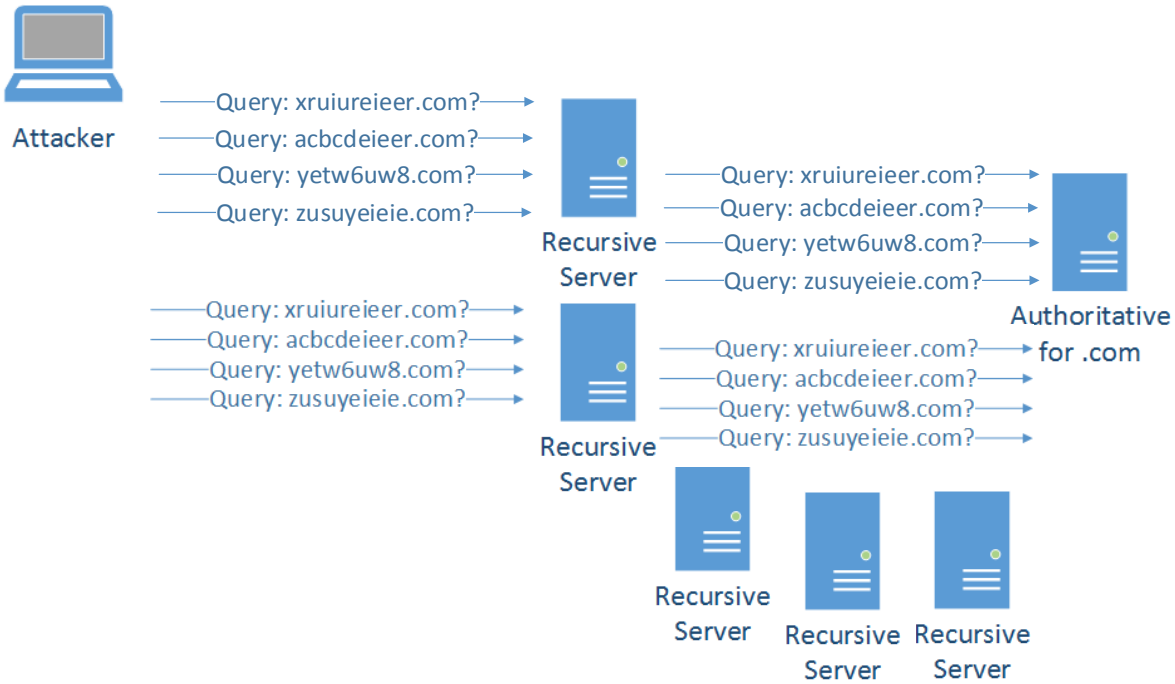
DNS resolution (rehash)

- 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?

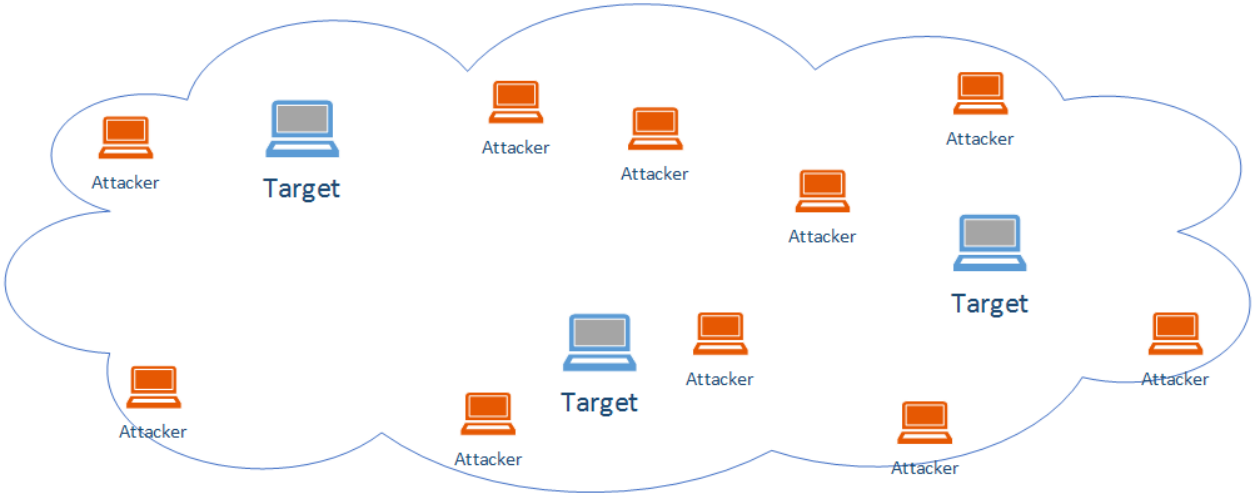
Attacker sends a query to recursive/reflector
Recursive forwards the query
And so on…
Imagine one more recursive resolver
Rinse and repeat…

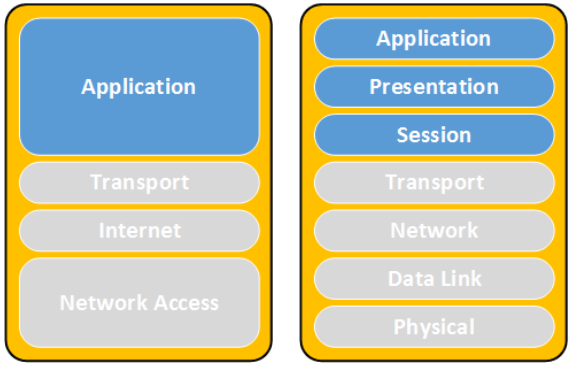


Questions?

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



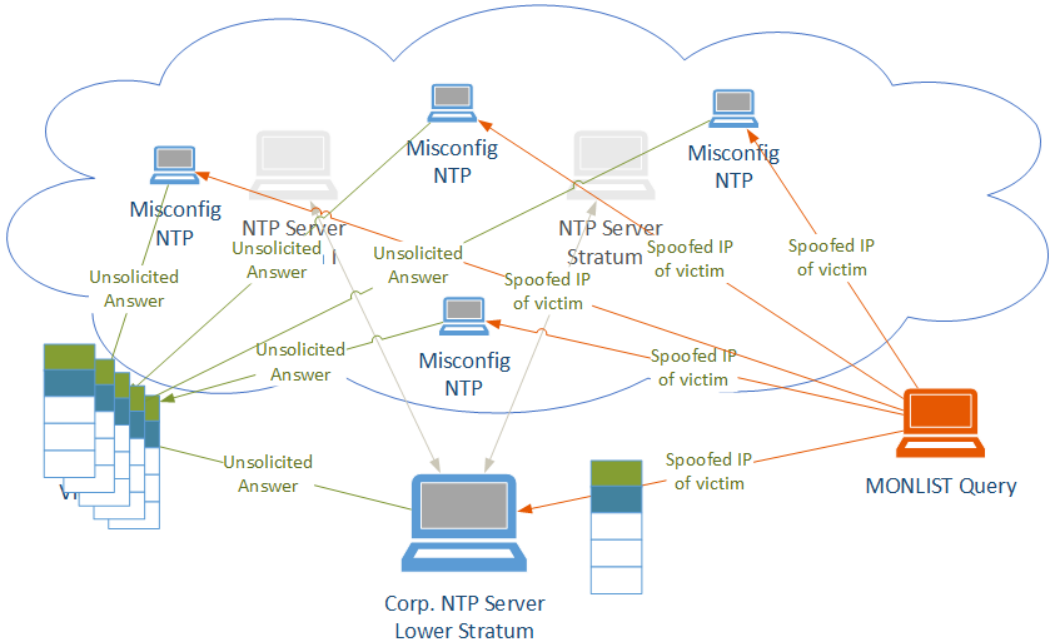


Network Time Protocol (NTP)

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)

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/

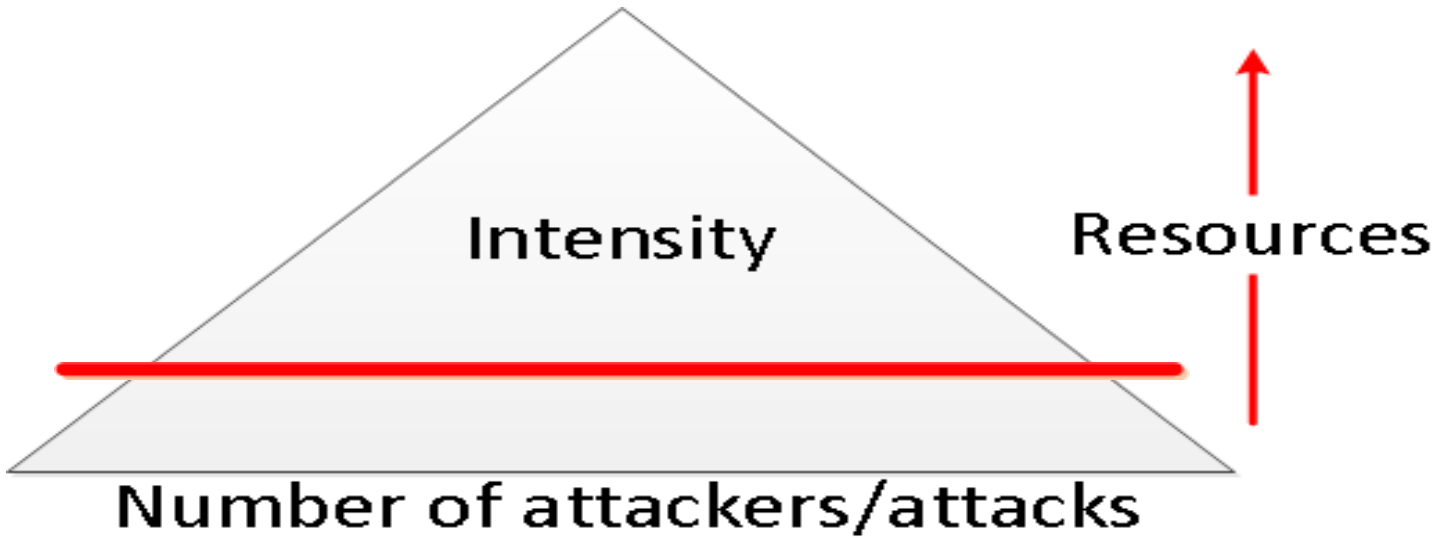
Questions?

Thank you!

krassi@fastly.com

Mitigation

Risk Pyramid



The cost of a minute?

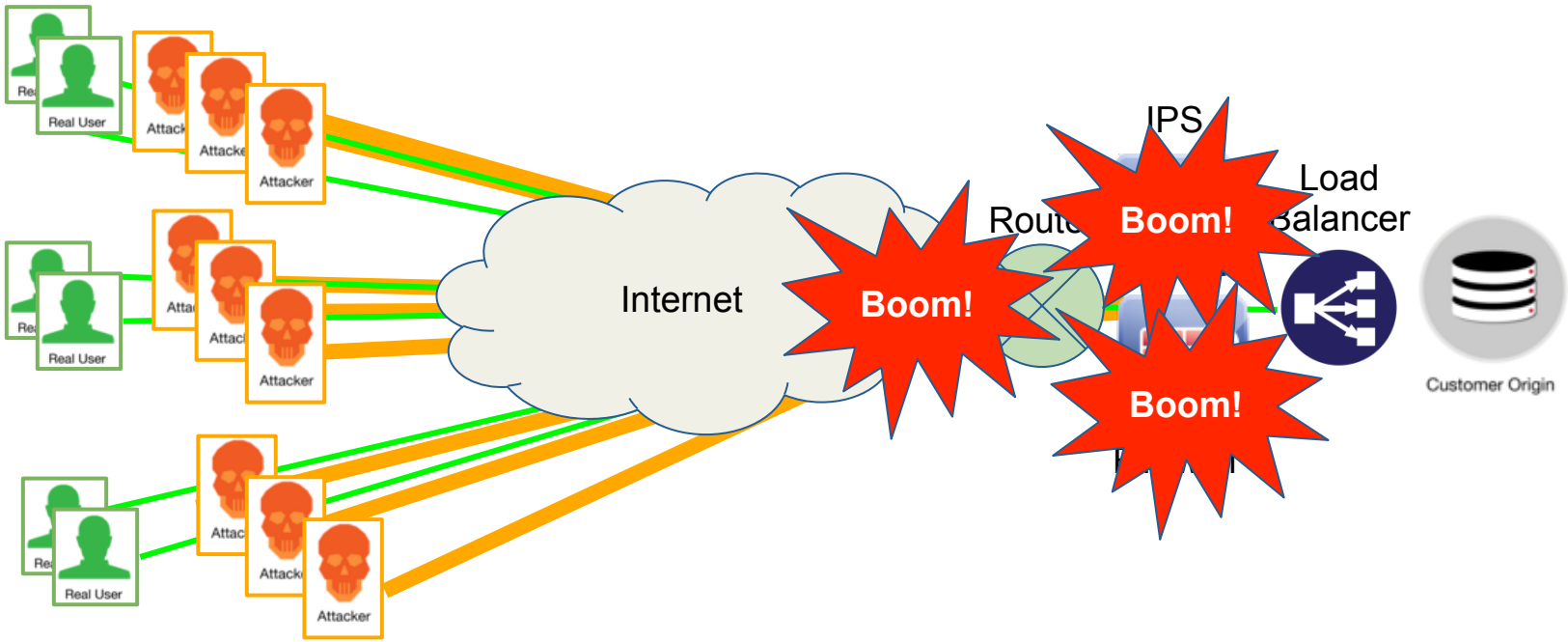
- How much does a minute of outage cost to your business?
- Are there other costs associated with it? Reputation?
- Are you in a risk category?
- How much is executive management willing to spend to stay up?

- 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

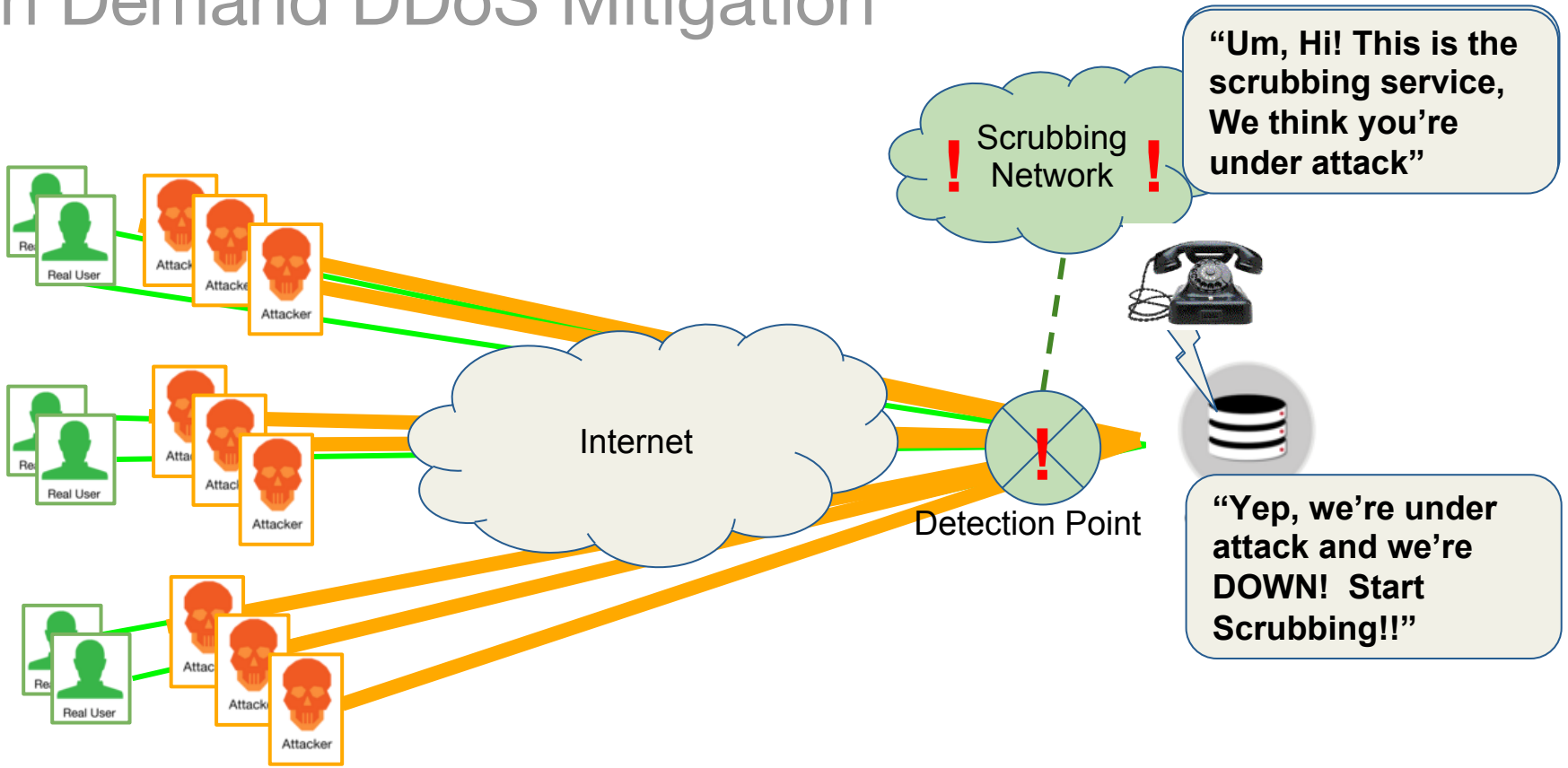
On Premise DDoS Mitigation



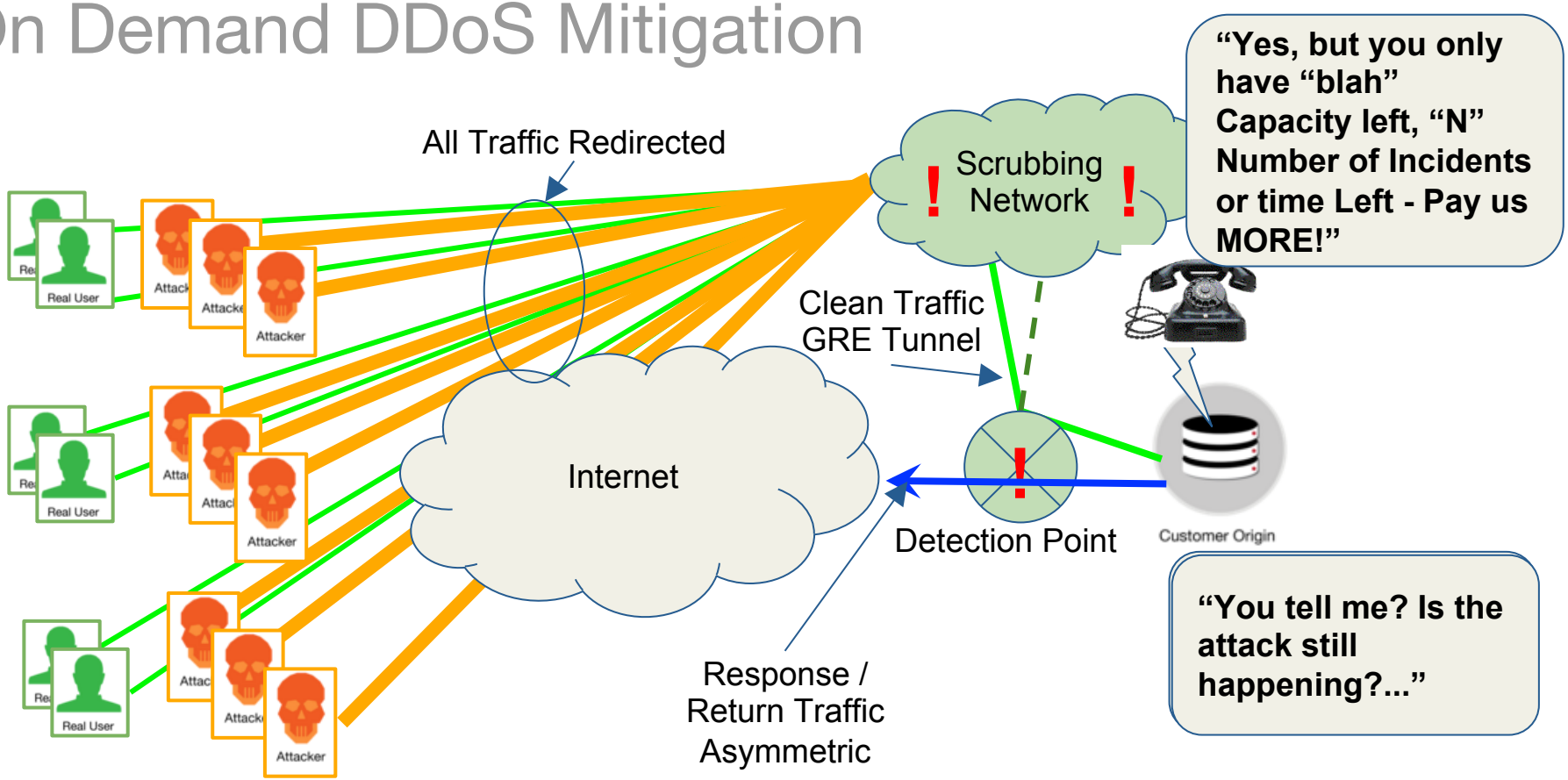
Outsource / scrubbing center

- Limited protocol support (usually HTTP/S)
- Added latency
- May loose visibility to source IP of the client
- Pay per MB of clean traffic (usually)
- Fast setup/Lower overhead
- More expensive per MB

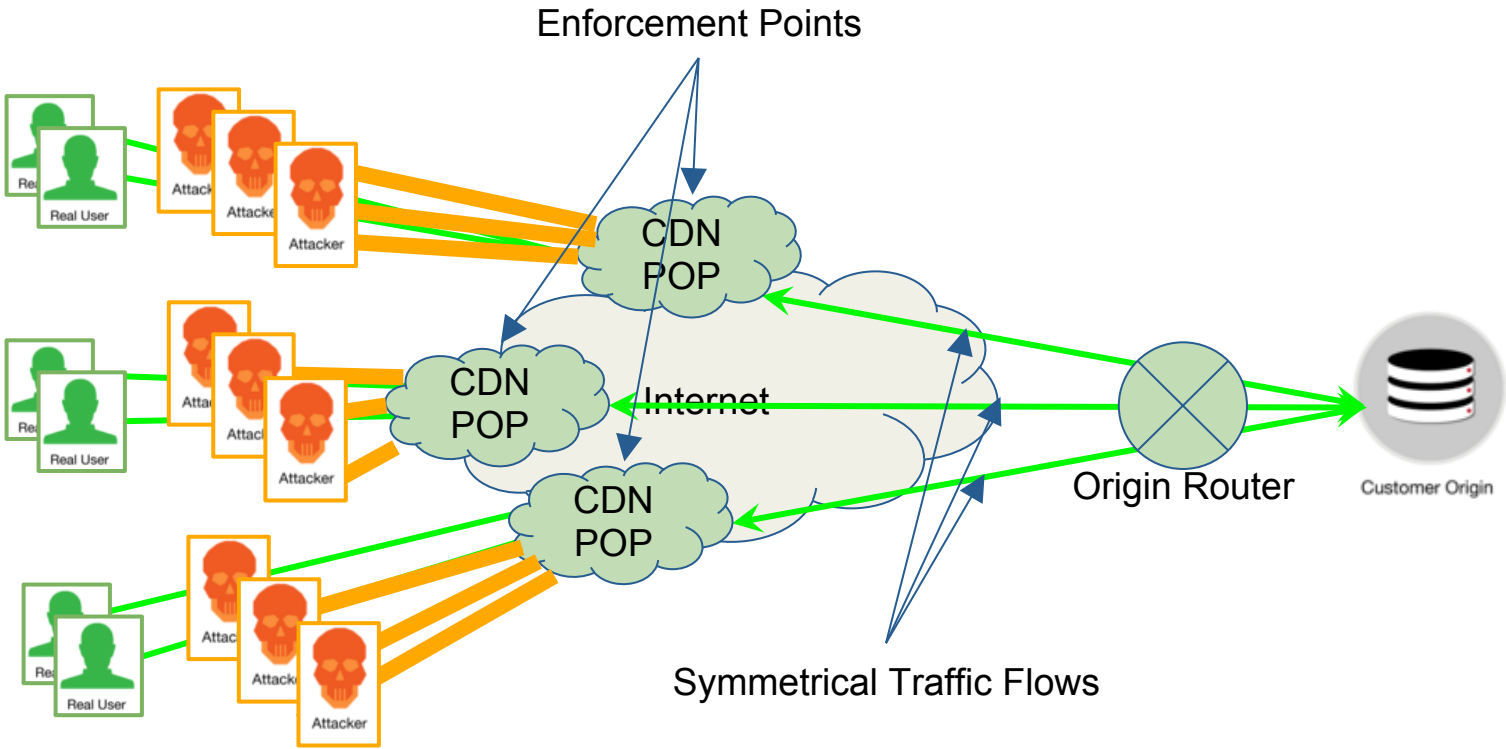
On Demand DDoS Mitigation



On Demand DDoS Mitigation



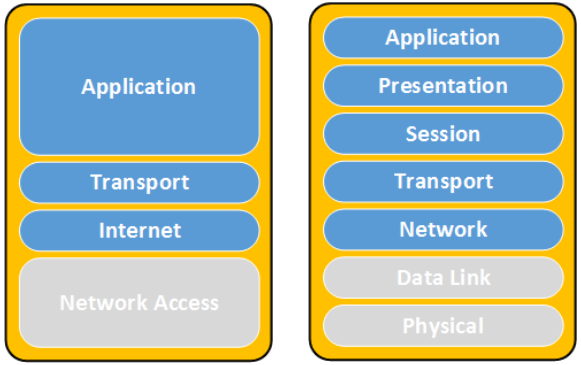
Always On DDoS Mitigation



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

- **Lots of money**

Defend the Internet

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

- **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”
- http://tools.ietf.org/html/bcp140
- Aka RFC 5358

Resources

- DNS
  http://openresolverproject.org/
- NTP
  http://openntpproject.org/
- 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