Introduction

Presenter: Krassimir Tzvetanov
Date: October 12th, 2021

Course developed by: Krassimir Tzvetanov

Copyright

Introduction

- Who am I?
  - Currently Doctoral Student at Purdue University
  - Research Assistant at Purdue CyberTAP
  - Consulting specializing in DDoS investigation and network stack optimization
  - Expert witness
  - krassi@krassi.biz
    krassi@purdue.edu

- What is the goal of this tutorial?
  - Aims to explain how the most popular attacks work
  - How to conduct system level mitigation with the help of other providers
  - It will not teach you how to be a DDoS mitigation service provider

Overview

- What is DDoS?
- Terminology
- Factors supporting and accelerating DDoS

What is DoS/DDoS?

What is Denial of Service?

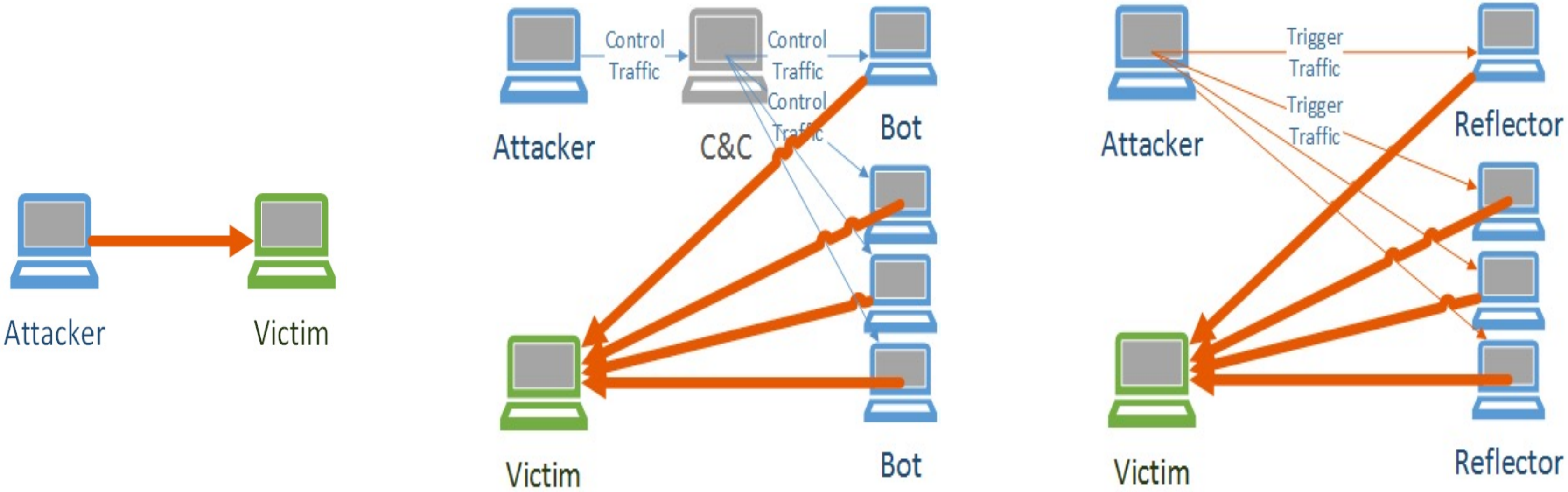
- Discussion
- Resource exhaustion… which leads to lack of availability
- Consider:
  - How is it different from The Guardian pointing to somebody's web site?
  - How is that different from company's primary Internet connection going down?

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

- What is the difference?
  - One system is sending the traffic vs many systems
  - Consider reflected attacks
- How does that change the attacks volume?
  - More systems – more capacity



Terminology

Terminology and Metrics

- Data rates (bandwidth)
  - bit/s
  - (SI x1000) kbit/s, Mbit/s, Gbit/s (bps, kbps, Mbps, Gbps)
  - (IEC80000-13 x1024) Kibit/s, Mibit/s, Gibit/s
    - Reference: https://en.wikipedia.org/wiki/Data-rate_units
- Storage
  - Bytes
  - (SI) kB, MB, GB
  - KiB, MiB, GiB
    - Reference: https://en.wikipedia.org/wiki/Units_of_information
- Clock rates
  - Hz – mostly GHz
- Latency
  - Usually, milliseconds
- Other
  - Packets per second
  - Requests per second

DDoS Volume Factors

Additional factors supporting and accelerating DDoS

- Overall bandwidth
- Reflectors
- IOT/Embedded home and SOHO devices
- Content management systems
- Booters/Stressors (lowers threshold)
- Accessible information

Home routers

- Embedded home and SOHO devices
  - Default username/password
  - Open DNS recursive resolvers
  - Software bugs (NetUSB)
  - Network diagnostic tools
  - Some do not allow the user to turn off DNS

- XBOX and Sony attacks over Christmas (2014)
  - Lizard Stresses, 2015
  - Mirai, 2017
- Is that intentional? – “follow the money”

Compromised CMSes

- Most targeted Content Management Systems:
  - WordPress
  - Joomla

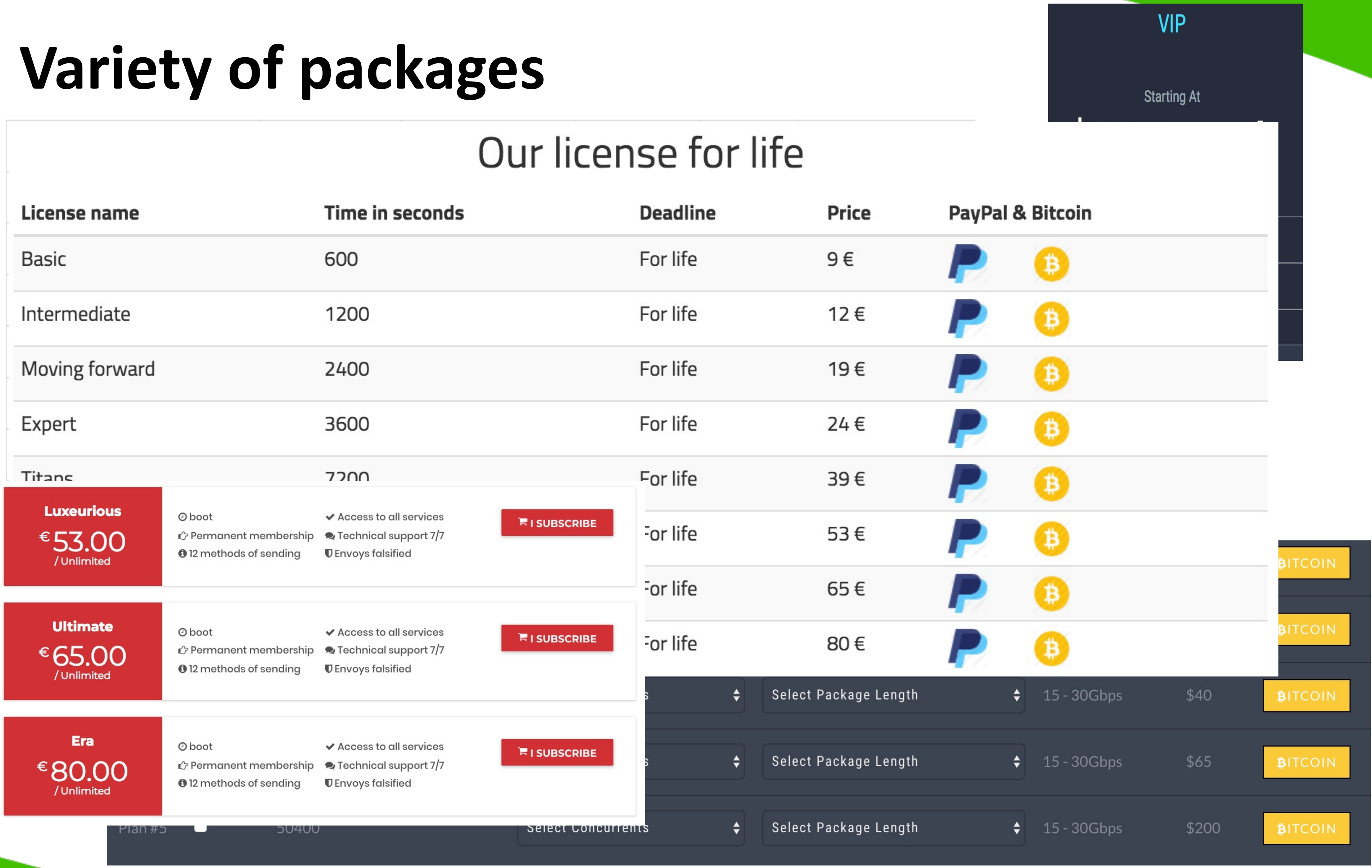
- Started in early 2013 - notably around the attacks against US financial institutions

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

Booters/Stressors

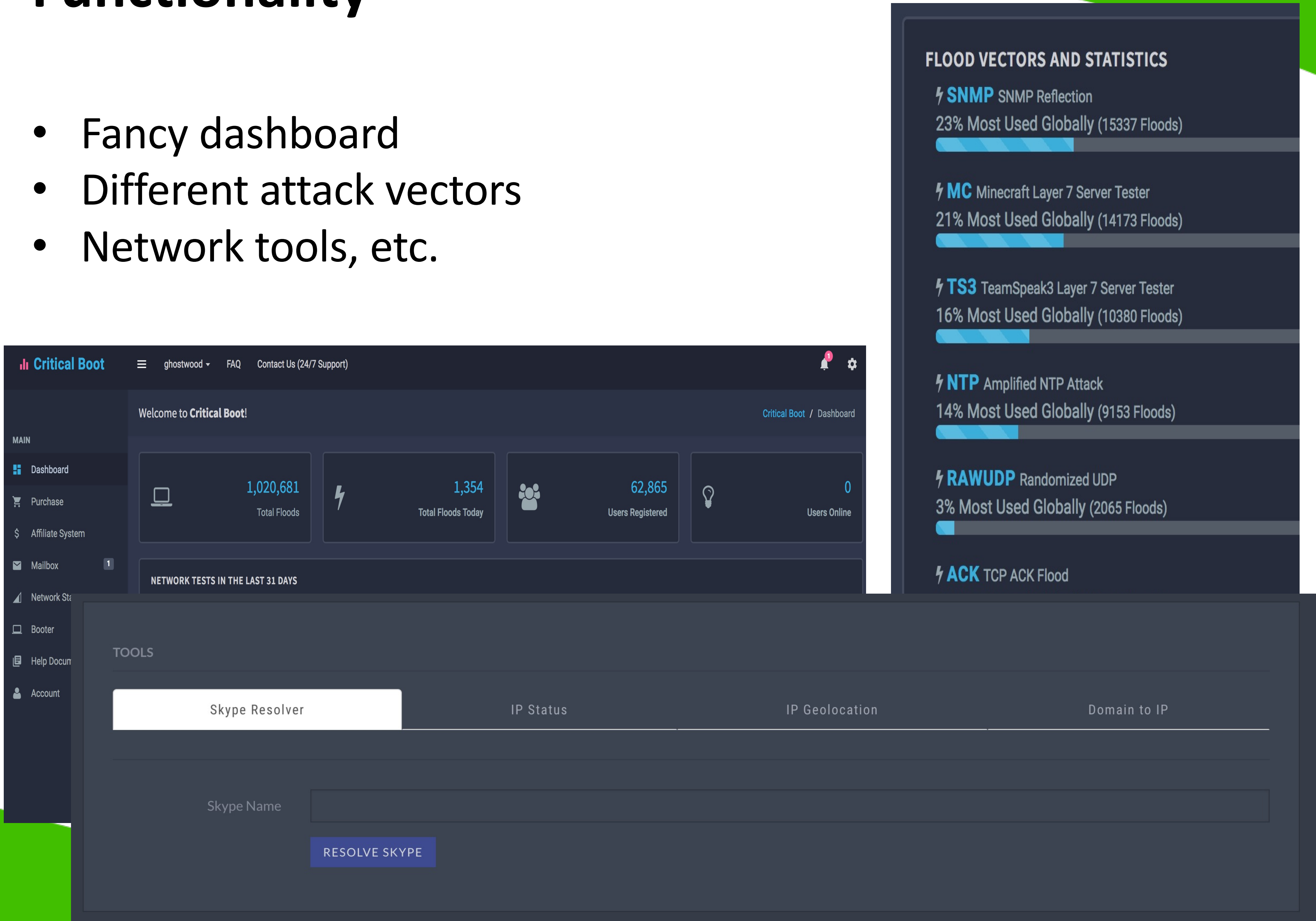
- Gained popularity over the past 6-7 years
- Inexpensive
- Popular among gamers
- Tools are sold for cheap on the black market (forums)
- Range 5-10 Gbps and up to 40GBps
- Usually short duration
- Mostly computer gaming industry related
    - Short, bursty attacks
    - Use rudimentary scripts
- Fairly inexpensive

Variety of packages



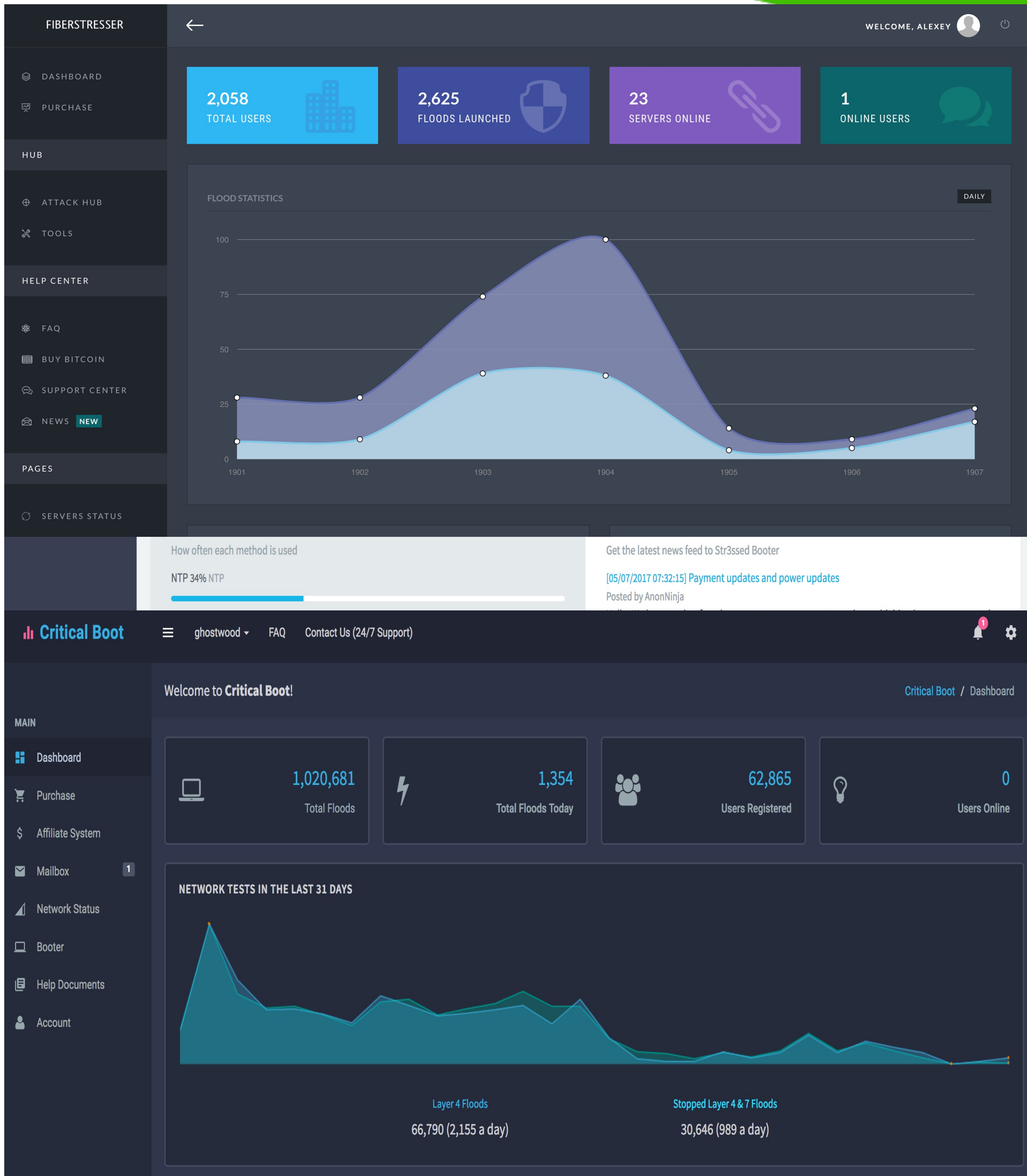
Functionality

- Fancy dashboard
- Different attack vectors
- Network tools, etc.

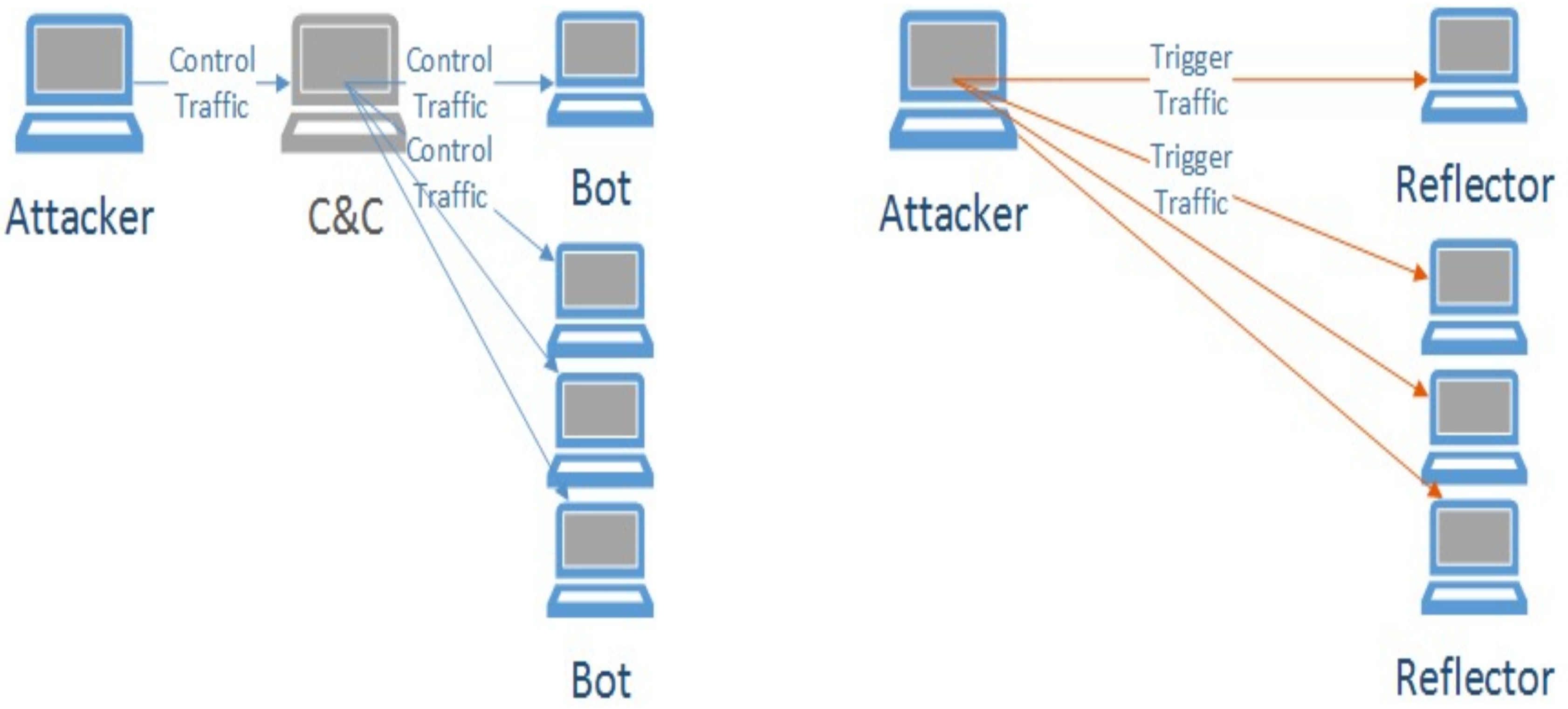


Code reuse

- Individual attack scripts reused widely
- Limited set of kits (control panel)
- Also some operators set multiple fronts



Low cost thanks to reflection



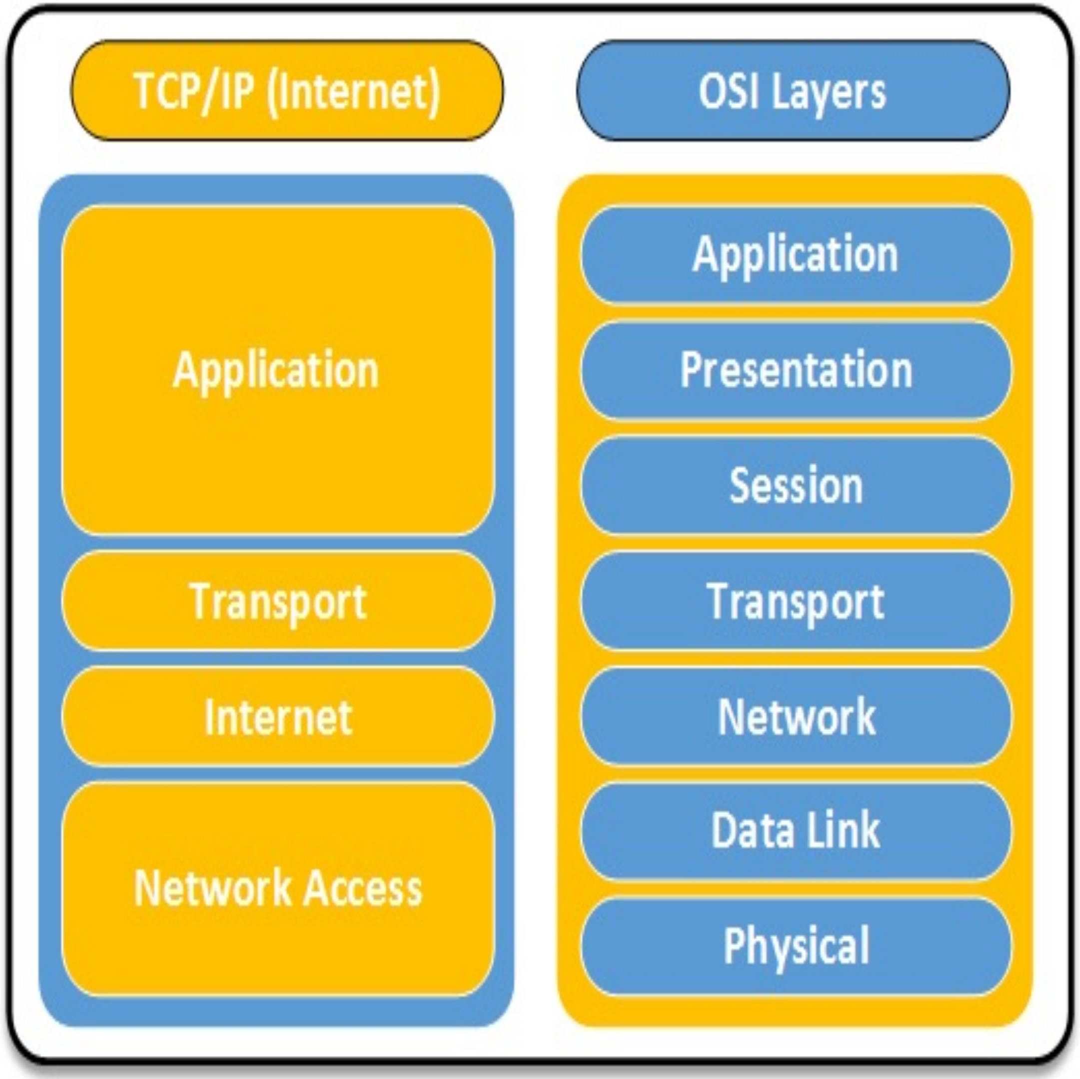
Questions

Attack Surface

Overview

- Attack Surface
- Correlation between layer and type of attack

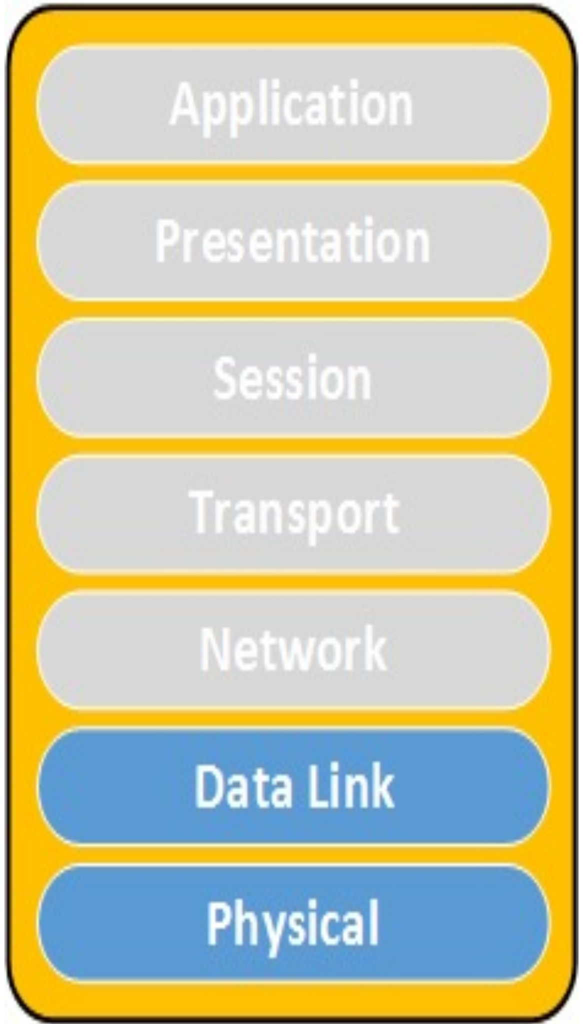
Network Layers – OSI vs Internet Model



Physical and Data-link Layers

- Cut cables
- Jamming
- Power surge
- EMP

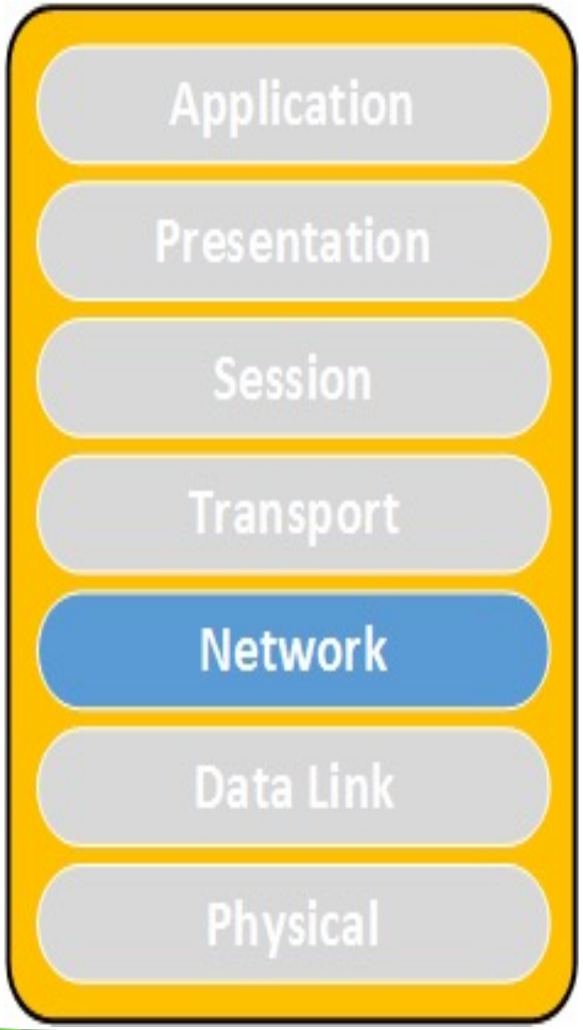
- MAC Spoofing
- MAC flood
- Wi-Fi Deauthentication



Network Layer

- Floods (ICMP)

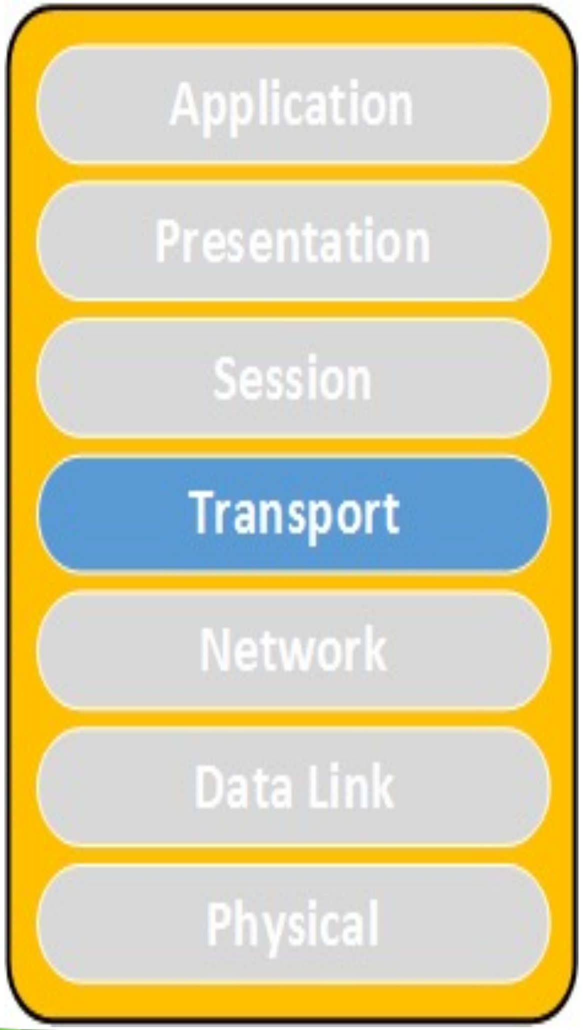
- Teardrop
  (overlapping IP segments)



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)

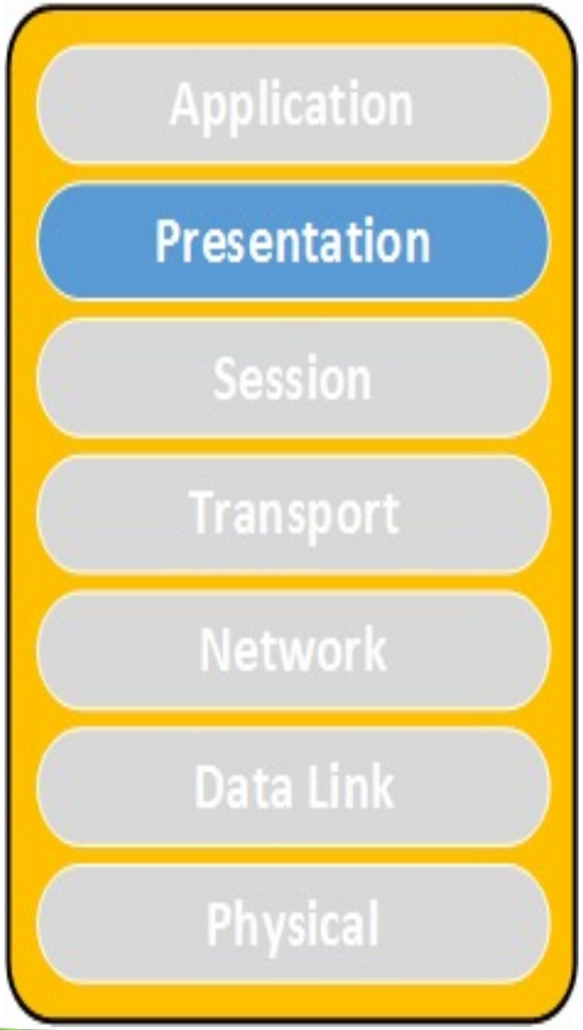


Session Layer

- Slowloris
- HTTP POST attack
- Sending data to a port with no NL/CR characters in it

Presentation Layer

- Expensive queries (repeated many times)
- XML Attacks (Billion laughs attack)
  <!DOCTYPE lolz
  [
  <!ENTITY lol1 "&lol2;">
  <!ENTITY lol2 "&lol1;">
  ]>
  <lolz>&lol1;</lolz>



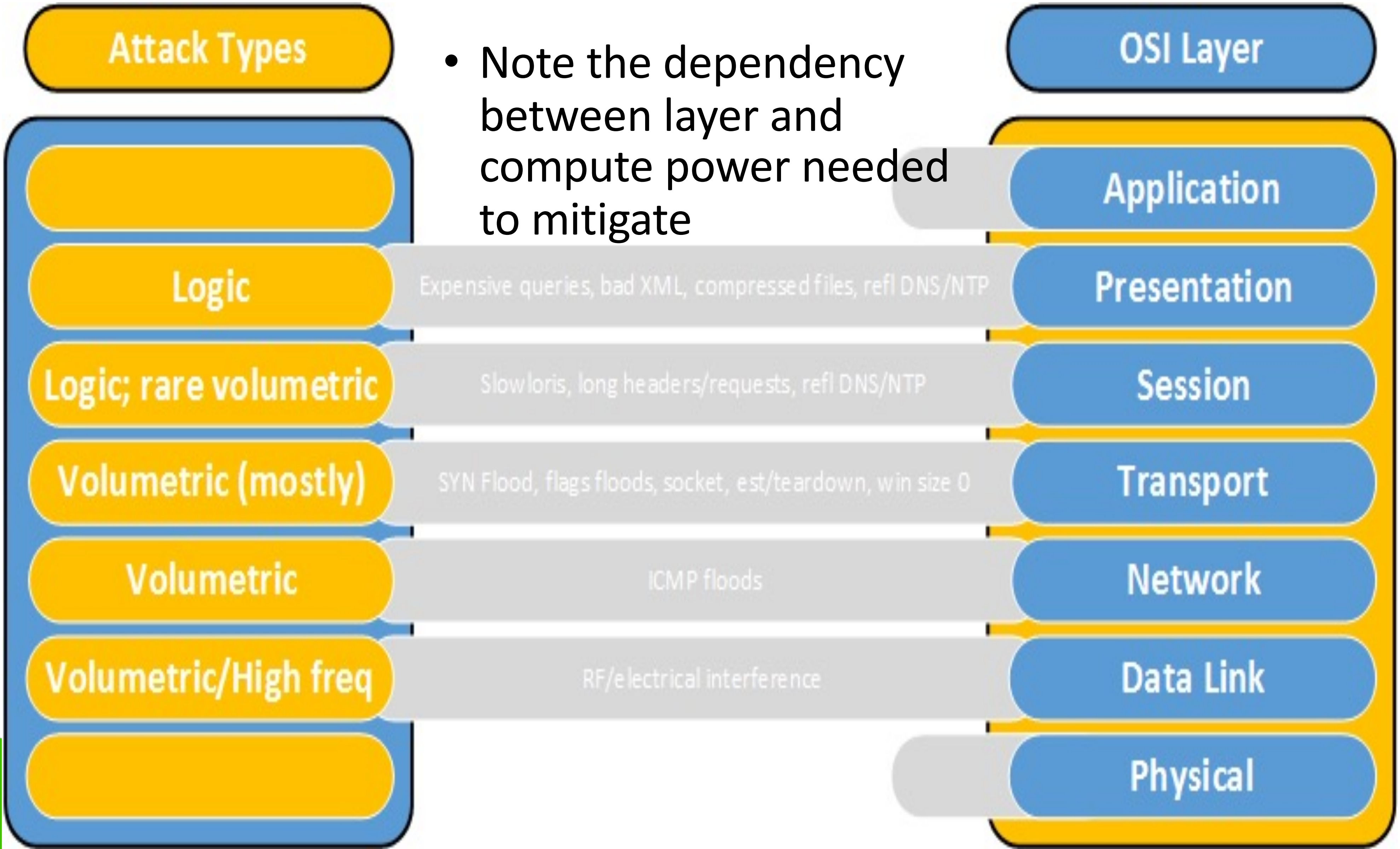
Application Layer

- Depends on the application
- Black fax

- Often confused with Internet Model Application Layer attacks.

Attack summary by layer

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



Questions

Network Technology

Overview

- Sockets
- TCP state machine
- Three-way handshake
- Use of some basic tools
- DNS Resolution

Sockets

Sockets

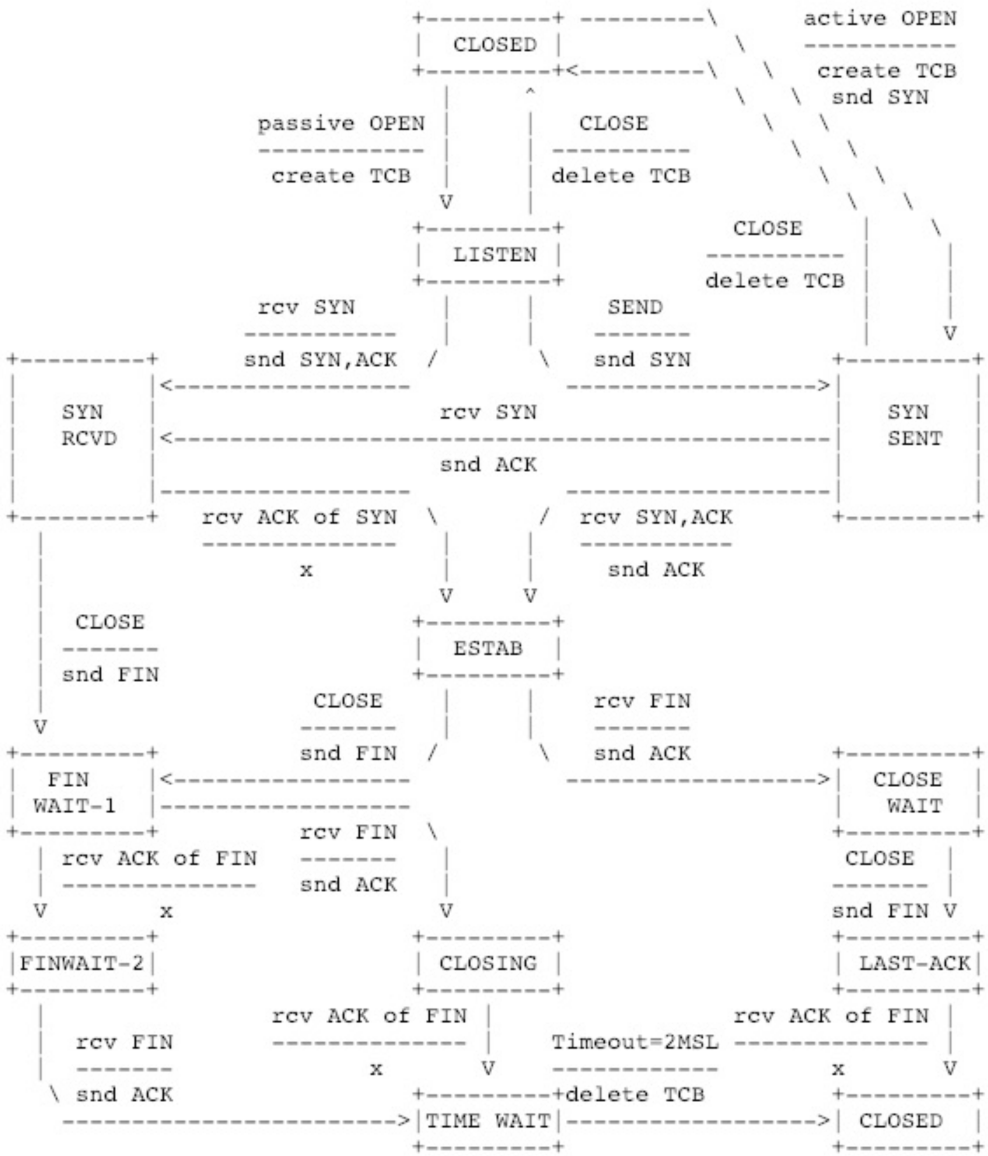
- Socket is an abstraction allowing an application to bind to a transport layer address (aka network port)
- It is described by a finite-state machine
- Throughout its lifetime it goes through a number of states

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

Socket State Diagram

- As described in RFC 791:

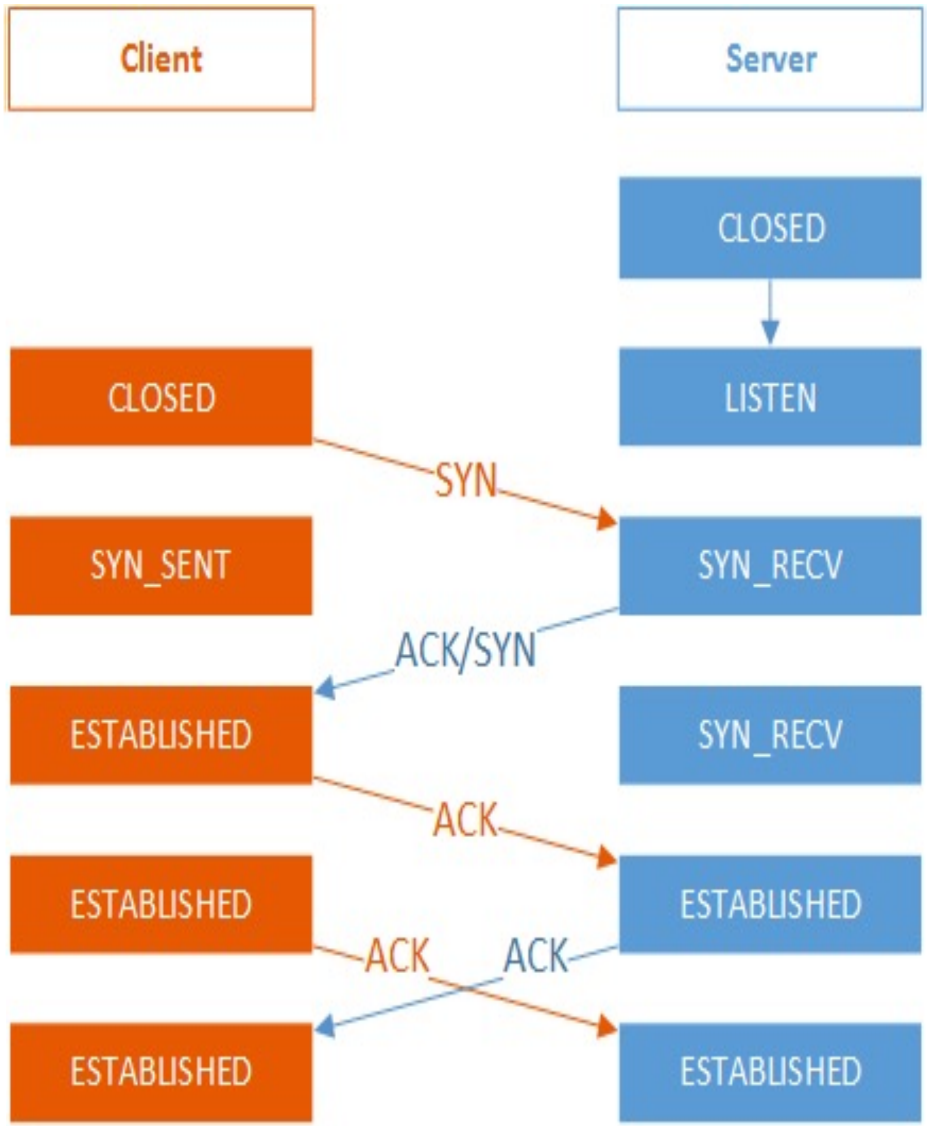


Source: RFC 791

Opening a TCP connection

Let's review the sequence for opening a connection

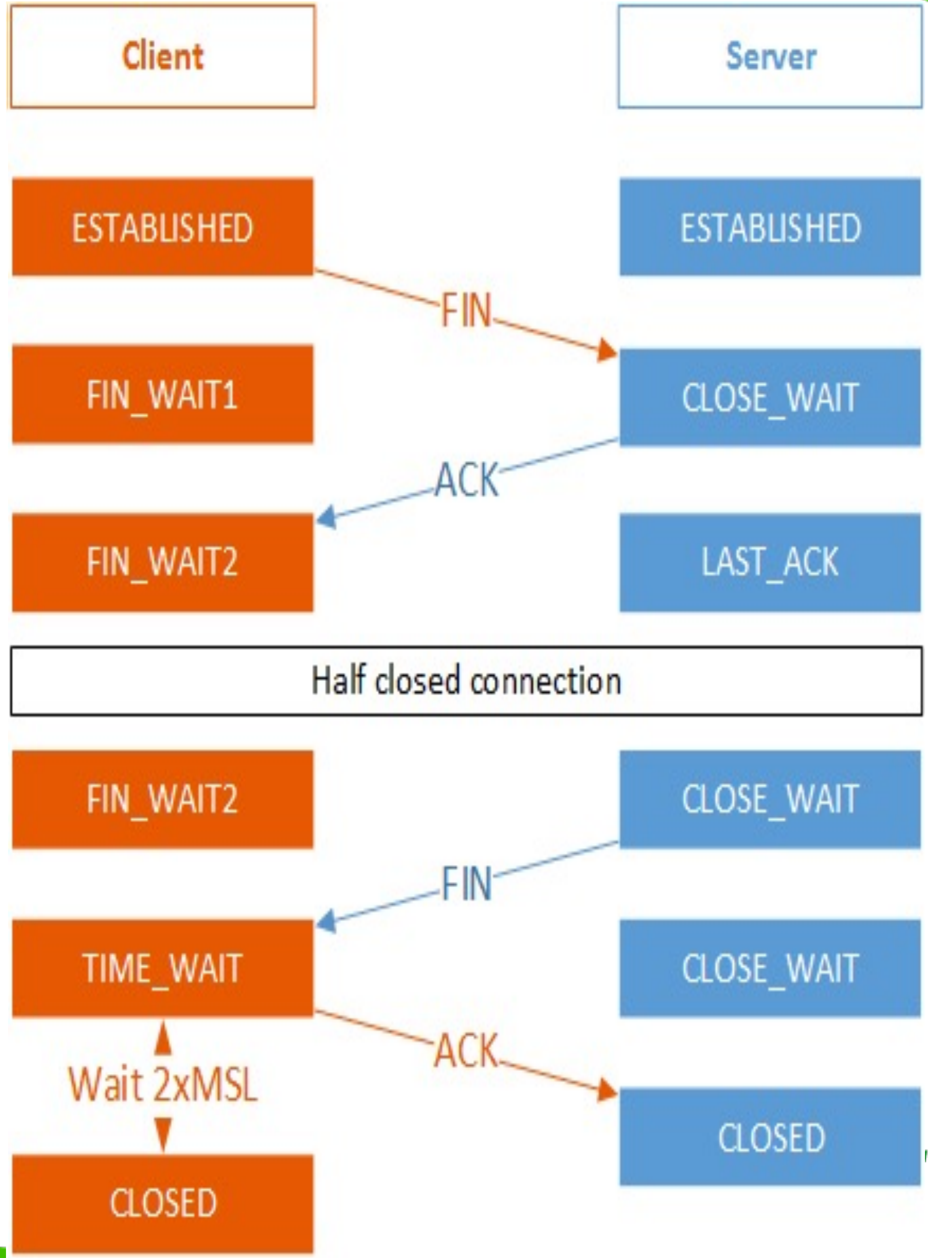
- Server side opens a port by changing to LISTEN state
- Client sends a SYN packet and changes state to SYN_SENT
- Server responds with SYN/ACK and changes state to SYN_RECV. For the client this is ESTABLISHED connection
- Client has to ACK and this completes the handshake for the server
- Packet exchange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

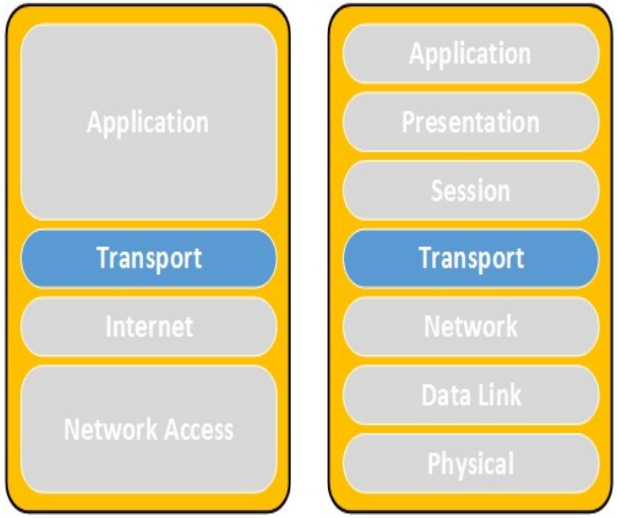
Questions

Attacks

Overview

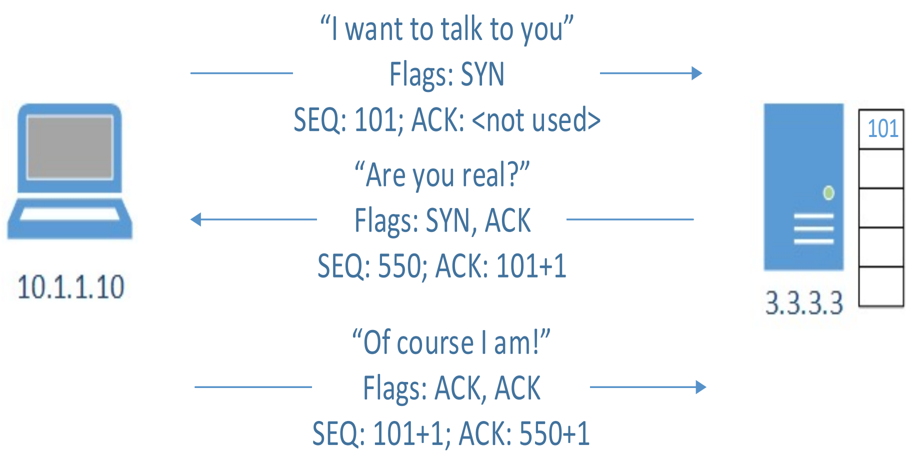
- SYN Flood
- SYN Cookies
- Socket Exhaustion (socket reuse)
- Sloworis

SYN Flood


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

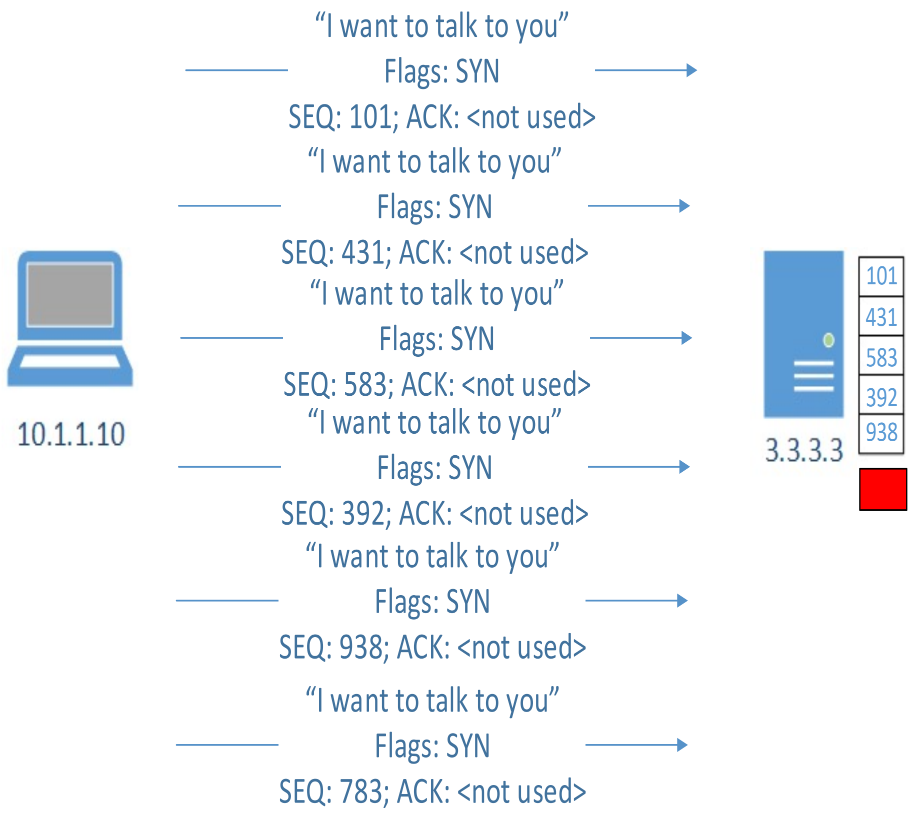
What is a SYN flood?

- 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

```
42 20.257541000 52.130.150.254   127.0.0.1   TCP   56 46036 > http [SYN]
43 20.257563000 78.94.151.254    127.0.0.1   TCP   56 49654 > http [SYN]
44 20.257574000 120.165.150.254  127.0.0.1   TCP   56 21280 > http [SYN]

▶Frame 42: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
▶Linux cooked capture
▼Internet Protocol Version 4, Src: 52.130.150.254 (52.130.150.254), Dst: 127.0.0.1 (127.0.0.1)
   Version: 4
   Header length: 20 bytes
  ▶Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Trans
   Total Length: 40
   Identification: 0xd701 (55041)
  ▶Flags: 0x00
   Fragment offset: 0
   Time to live: 255
   Protocol: TCP (6)
  ▶Header checksum: 0x9a4c [validation disabled]
   Source: 52.130.150.254 (52.130.150.254)
   Destination: 127.0.0.1 (127.0.0.1)
   [Source GeoIP: Unknown]
   [Destination GeoIP: Unknown]
▼Transmission Control Protocol, Src Port: 46036 (46036), Dst Port: http (80), Seq: 0, Len: 0
   Source port: 46036 (46036)
   Destination port: http (80)
   [Stream index: 35]
   Sequence number: 0    (relative sequence number)
   Header length: 20 bytes
  ▶Flags: 0x002 (SYN)
   Window size value: 65535
   [Calculated window size: 65535]
  ▶Checksum: 0xb9c2 [validation disabled]
```

- Attacker
  - Random IP address/port
- Target
  - 127.0.0.1:80
- Pay attention to the SYN flag!

SYN flood mitigation

- Technology
  - SYN Cookies
  - Whitelists + syn cookies
- In the past
  - TCP Proxy (TCP Intercept – active mode)
  - TCP Resets (TCP Intercept – passive)

What is a SYN cookie?

- Preserves information in ISN (initial sequence number)

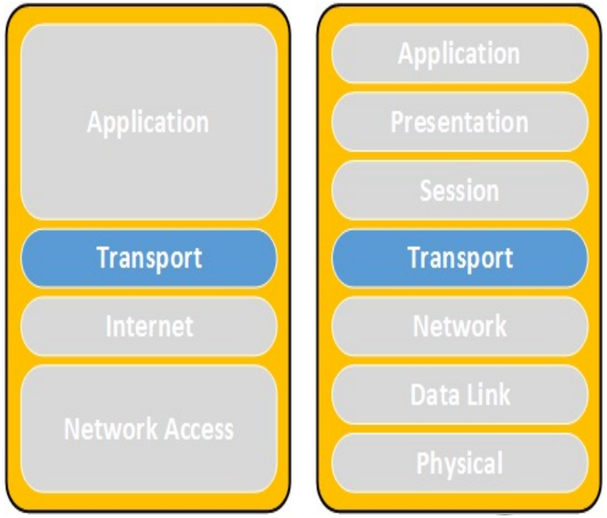
- SYN Cookie:

  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)

Questions

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?

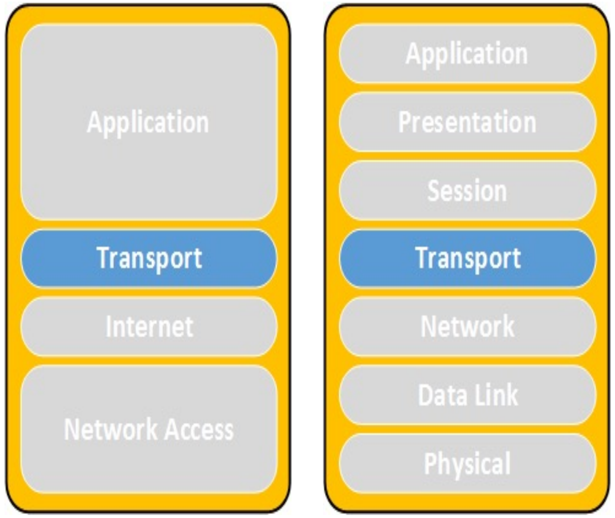
Socket Exhaustion through the eyes of netstat

- 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   -
```

Questions

Slowloris


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

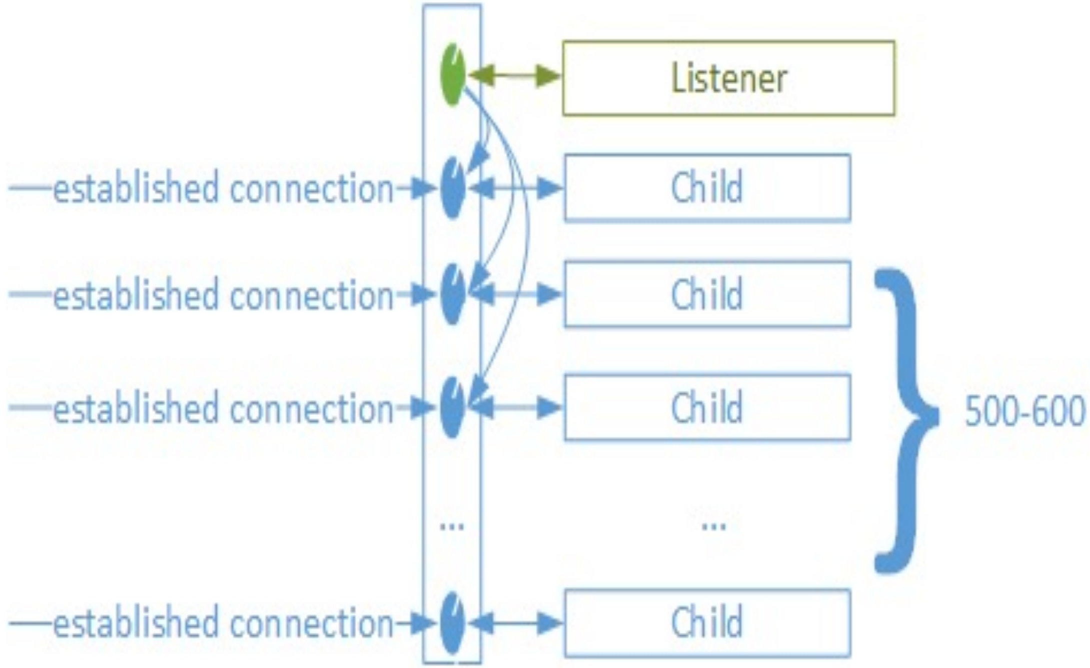
Connection handling architectures

- Process based connection handling?
  - Think “Apache”

- Event based connection handling?
  - Think “nginx”

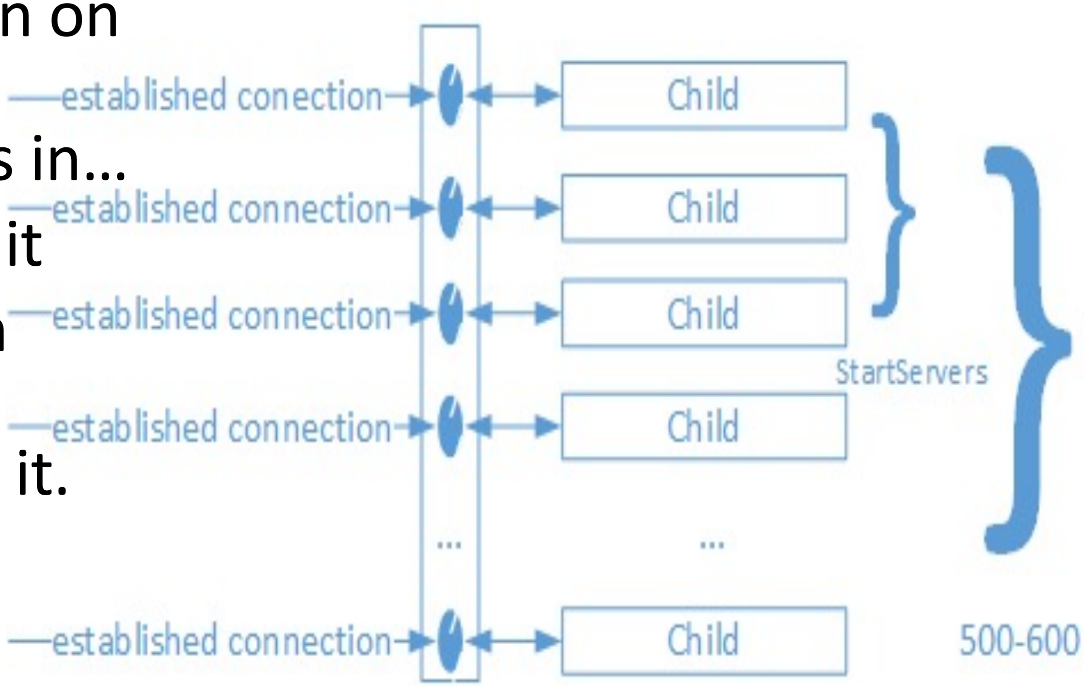
Process oriented explained

- 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



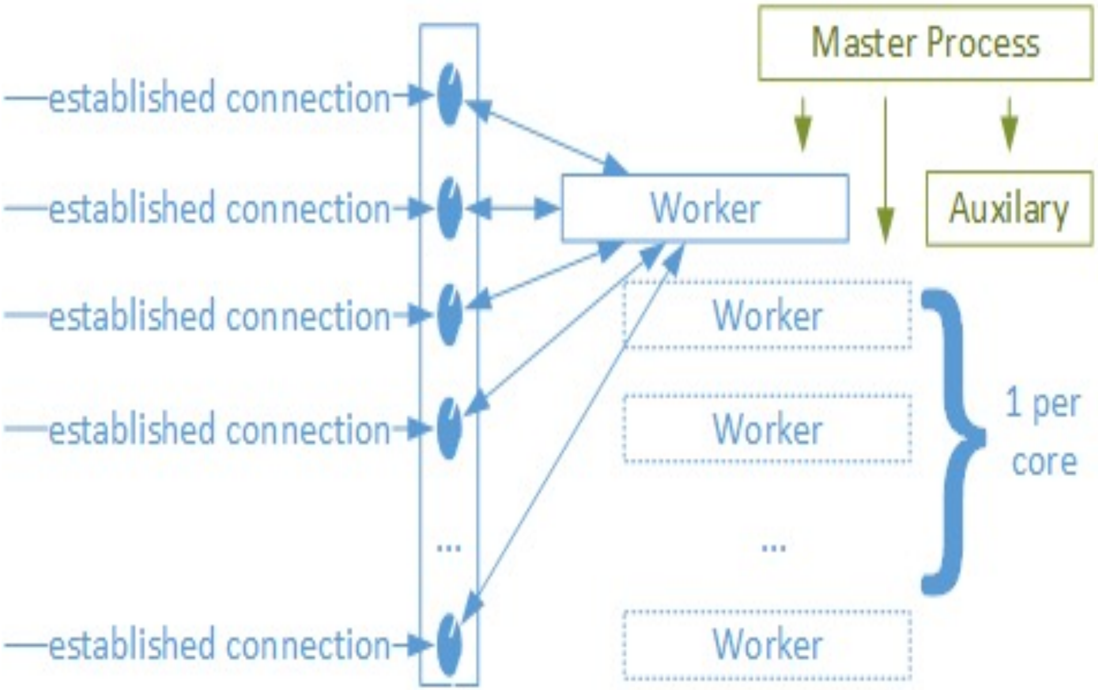
Nginx (simplified)

- 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;

…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…

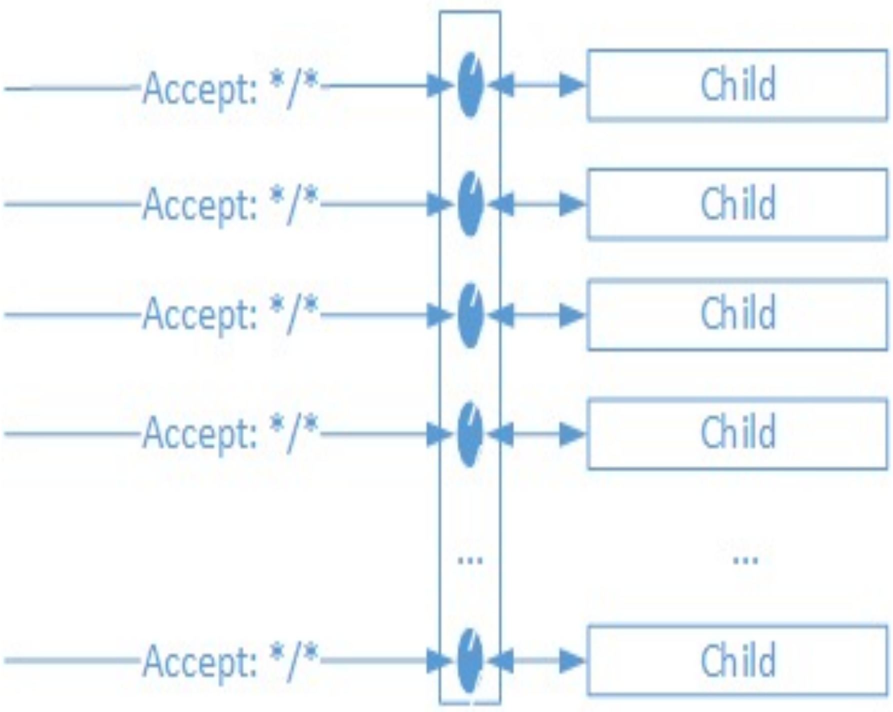
  …and so on.

  …and waits…

  …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

Overview

- Reflection
- Amplification
- DNS Reflection Attacks
- Backstatter
- Cache Busting Attacks

Two different terms

- Reflection
  using an intermediary to deliver the attack traffic

- Amplification
  ability to deliver larger response than the query traffic

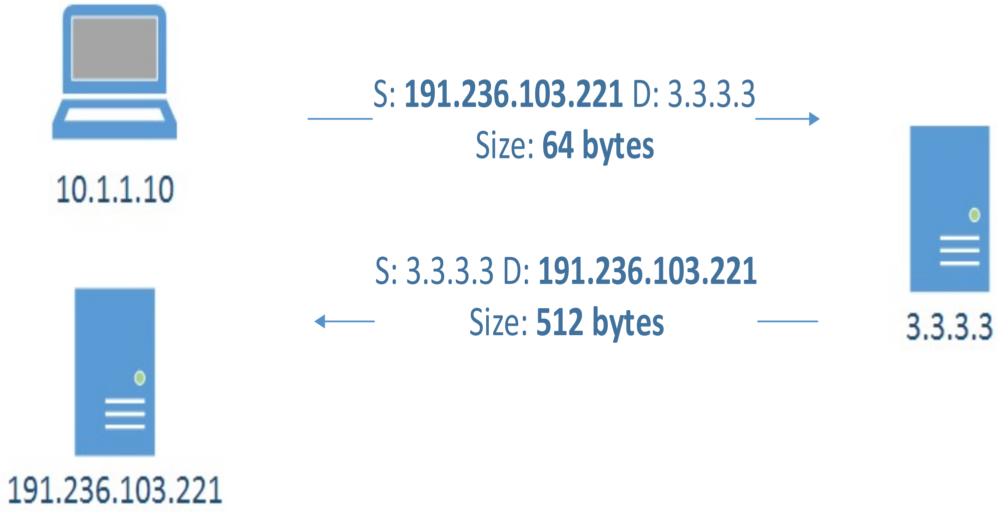
Reflection

Reflection 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 an intermediary. The forged source address is going to be the one of the target. The intermediary will respond and this packet will go to the target instead of the attacker

What is reflection(ed) attack?

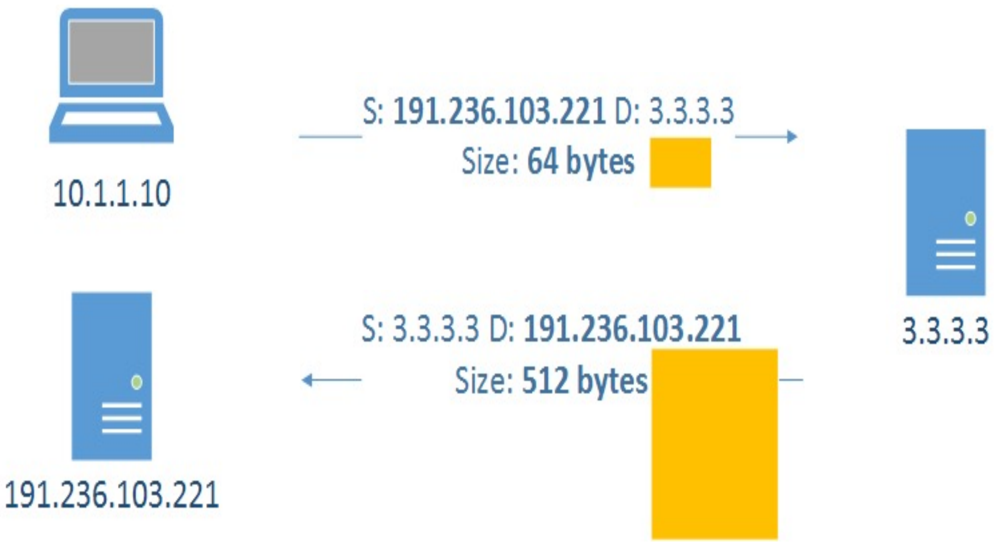
- Attacks where the an unwilling intermediary is used to deliver the attack traffic
- Attacker sends a packet with a spoofed source IP set to the victim’s
- Reflectors respond to the victim



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

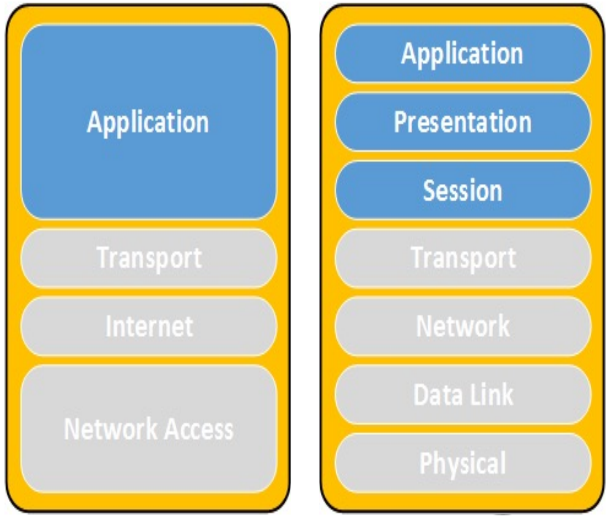
Amplification quotients

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



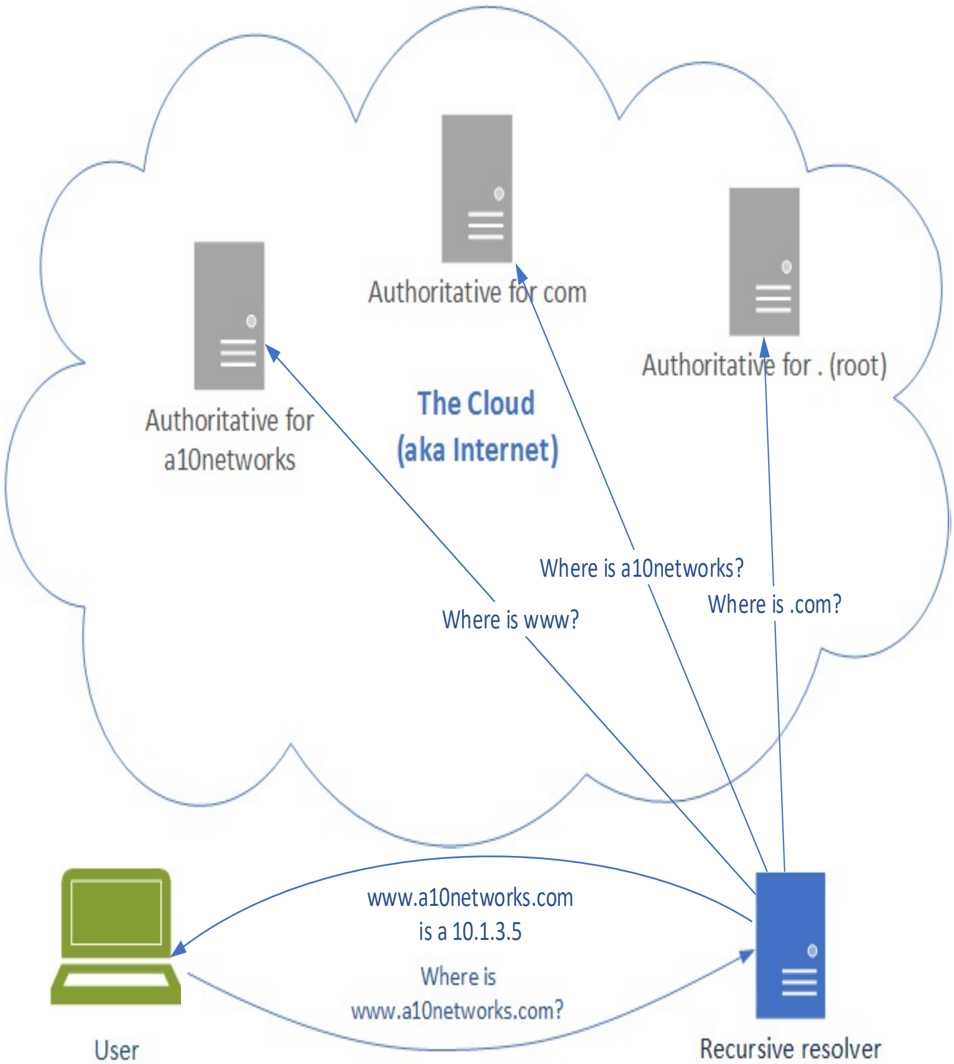
DNS server types

- 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

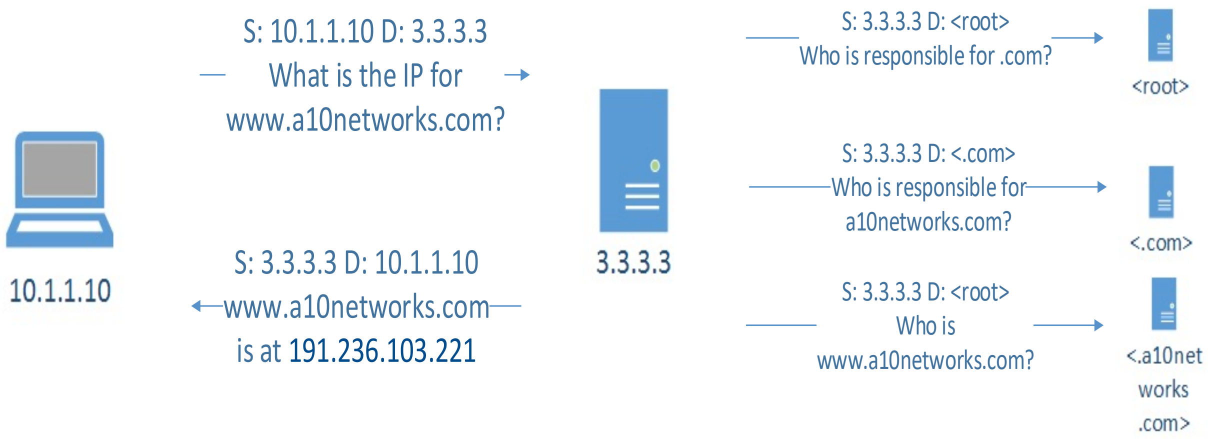
DNS resolution

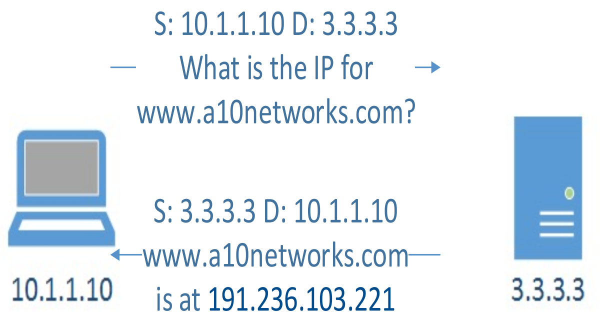
- How does DNS work?
- User talks to recursive resolver
- The recursive goes on the Internet and talks to the authoritative servers
- When an answer is obtained (or not) it reports back to the user



DNS resolution at the packet level

- The process of mapping:
  www.a10networks. com => 191.236.103.221

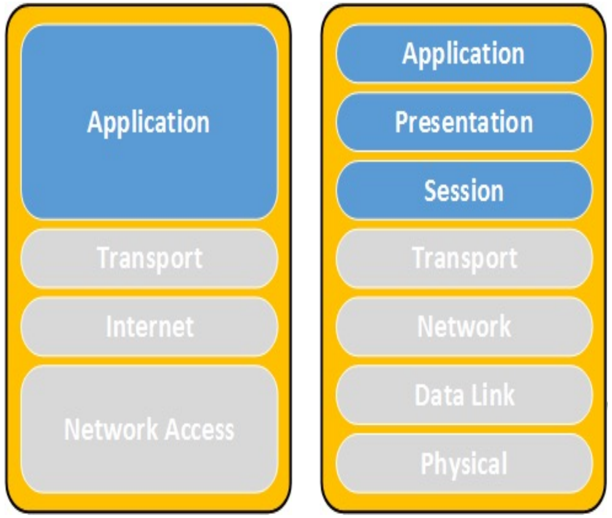




…if the answer
was cached

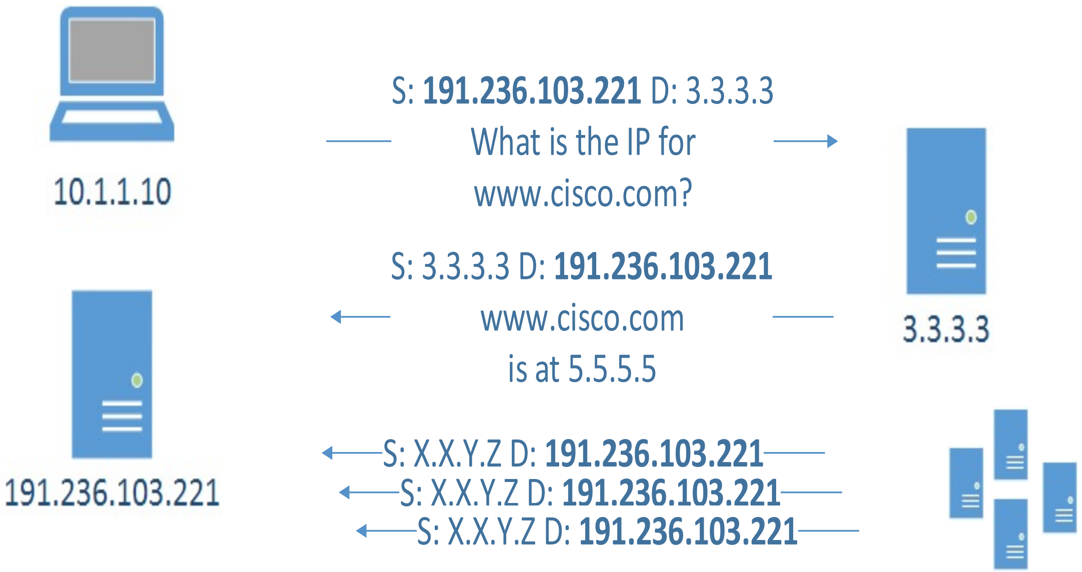
Questions

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:

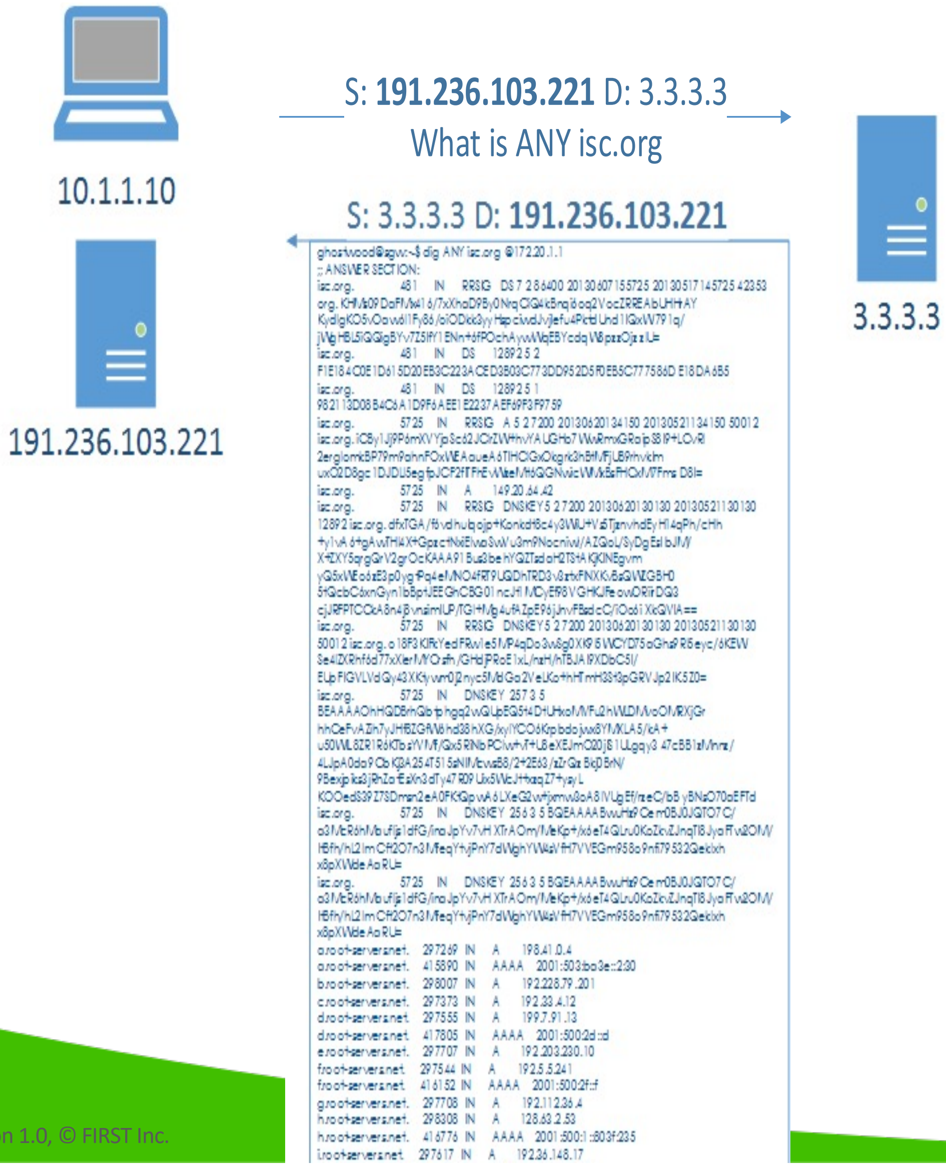
  ~$ dig ANY isc.org @172.20.1.1 # My home lab

  Flip over for thes 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/oiODkk3yyHspciwdJvjIefu4PktdUnd1IQxW 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
2erglomkBP79m9ahnFOxWEAaueA6TIHClGxOkgrk3hBtMFjUB9rhvkIm uxO2D8gc1DJDLl5egfpJCF2fITFhEvWzeMt6QGNwicWMxBsFHCxM7Fms 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/ EUpFIGVLVdQy43XKtywm0j2nyc5MdGa2VeLKo+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+/x6eT4QLru0KoZkvZJnqTI8JyaFTw2OM/ItBfh/hL2lm
Cft2O7n3MfeqYtvjPnY7dWghYW4sVfH7VVEGm958o9nfi79532QekIxh 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
```

Reflection and Amplification



On the wire

127.5.5.5	Attack	127.0.0.1	DNS	70	Standard query 0x4918 A test.com
127.5.5.5	traffic	127.0.0.1	DNS	70	Standard query 0x4918 A test.com
127.5.5.5		127.0.0.1	DNS	70	Standard query 0x4918 A test.com
127.5.5.5		127.0.0.1	DNS	70	Standard query 0x4918 A test.com
127.0.0.1	Reflector	127.5.5.5	DNS	153	Standard query response 0x4918 A 192.1
127.5.5.5	Target	127.0.0.1	ICMP	181	Destination unreachable (Port unreachab

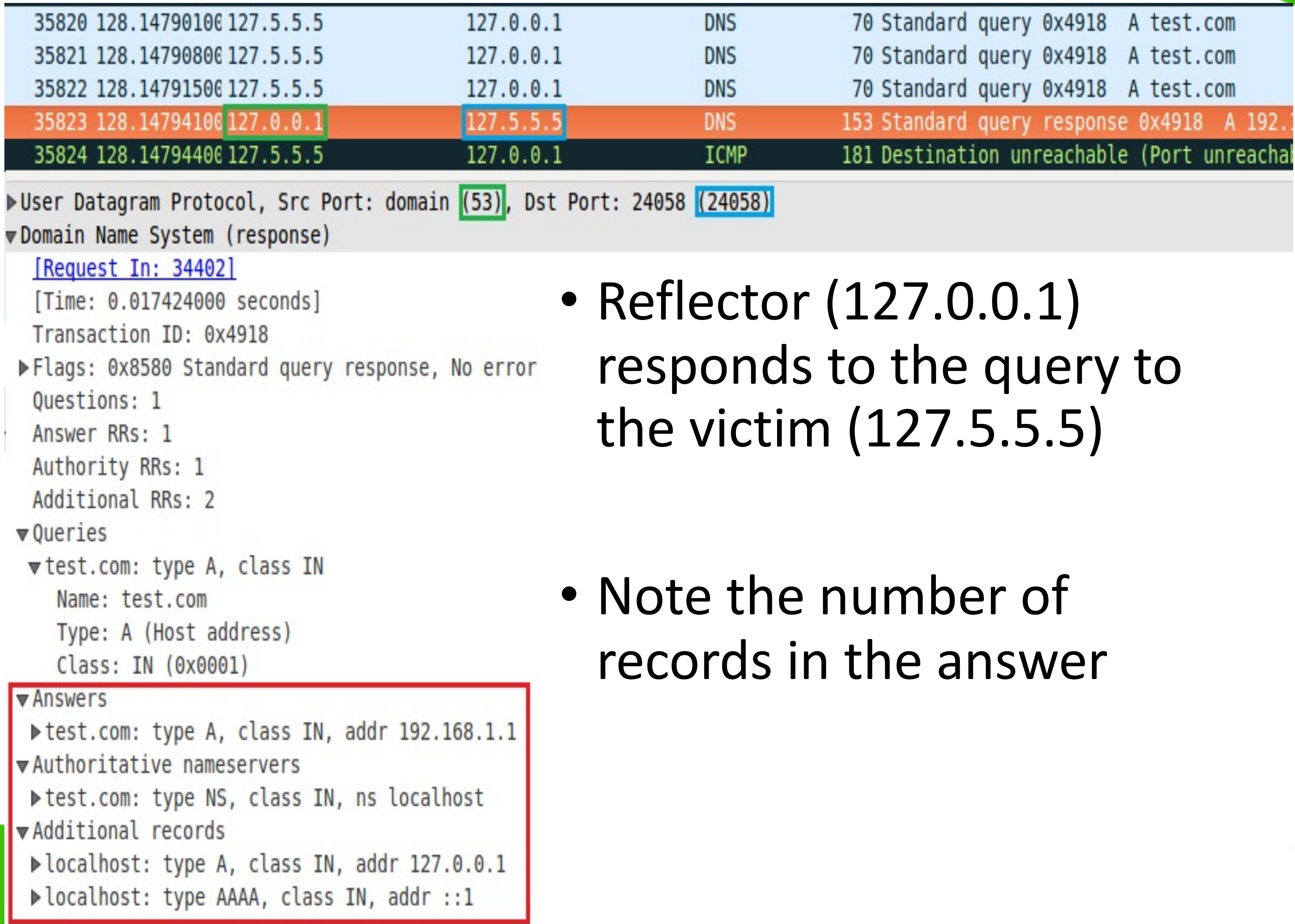
- Victim is 127.5.5.5
- Attacker spoofs traffic as if it comes from 127.5.5.5
- Reflector (127.0.0.1) responds to the query to the victim
- BACK SCATTER
  Notice the victim is responding with port unreachable because there is nothing running on that UDP port.
  This is called back-scatter

On the wire (details)

```
35820 128.14790100 127.5.5.5    127.0.0.1    DNS    70 Standard query 0x4918  A test.com
35821 128.14790800 127.5.5.5    127.0.0.1    DNS    70 Standard query 0x4918  A test.com
35822 128.14791500 127.5.5.5    127.0.0.1    DNS    70 Standard query 0x4918  A test.com
35823 128.14794100 127.0.0.1    127.5.5.5    DNS    153 Standard query response 0x4918  A 192.1
35824 128.14794400 127.5.5.5    127.0.0.1    ICMP   181 Destination unreachable (Port unreachab

▶Frame 35820: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
▶Linux cooked capture
▶Internet Protocol Version 4, Src: 127.5.5.5 (127.5.5.5), Dst: 127.0.0.1 (127.0.0.1)
▶User Datagram Protocol, Src Port: 49249 (49249), Dst Port: domain (53)
▼Domain Name System (query)
   Transaction ID: 0x4918
  ▶Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
  ▼Queries
   ▼test.com: type A, class IN
      Name: test.com
      Type: A (Host address)
      Class: IN (0x0001)
```

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

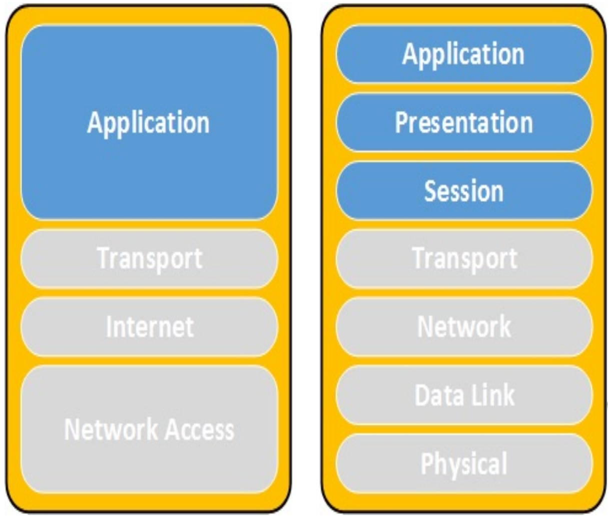
On the wire (details)



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

- Note the number of records in the answer

Non-reflective DNS attacks



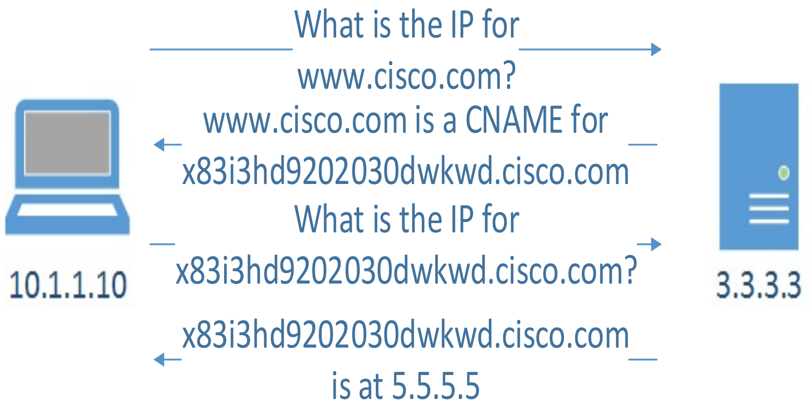
DNS attacks mitigation (victim)

- Validate packet and query structure
- Whitelisting
- Challenges*
- High performance equipment
  - Variety of techniques
  - Vendor dependent

- Drop known reflector traffic:
  http://openresolverproject.org/

DNS attacks mitigation (victim - DNS challenge)

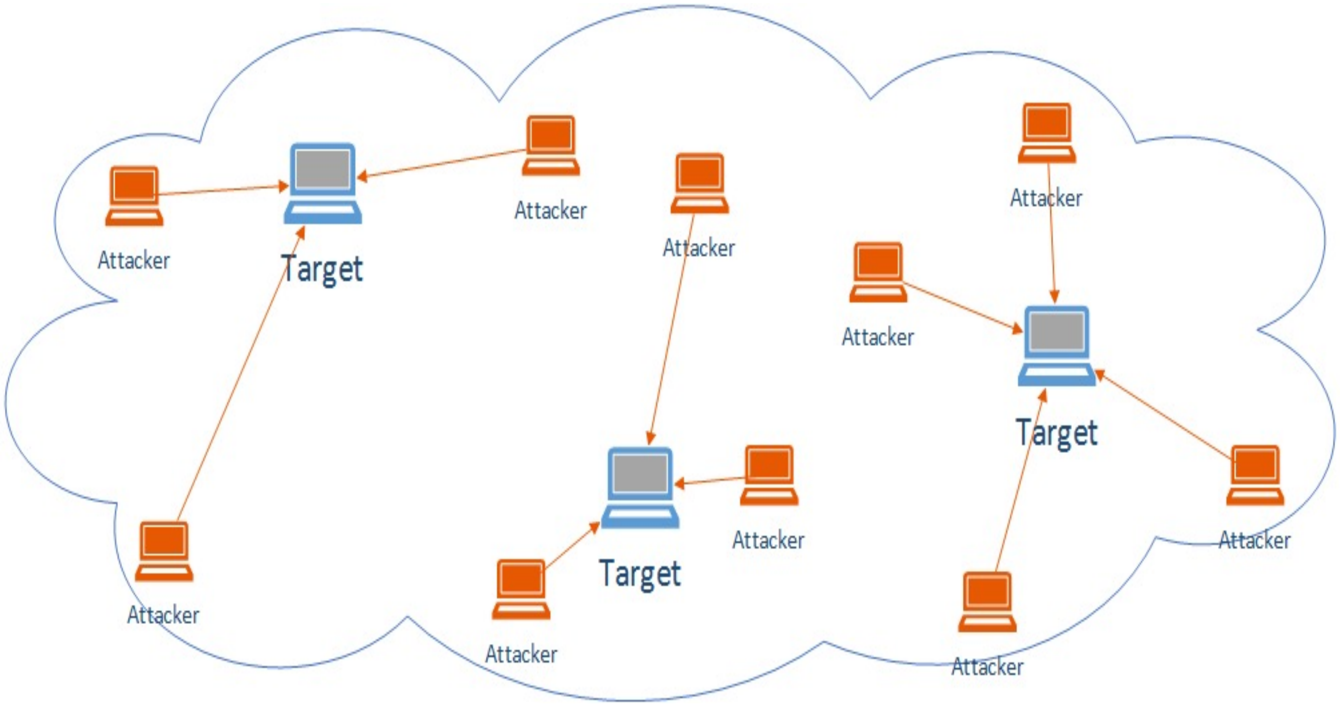
- What is a DNS challenge?



- Challenges with DNS challenge?
  - Two times the amount of traffic
  - Two times the packet rate
  - Computational resources

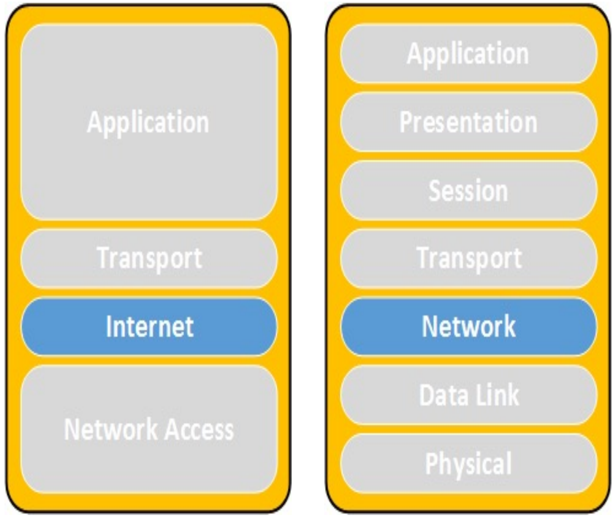
Large scale mitigation and load distribution: Anycast

- Unicast operation: one point of presence, all traffic goes there
- Anycast: multiple points of presence advertise the same address space
- Network ensures user is routed to the “closest” instance



Questions

Backscatter

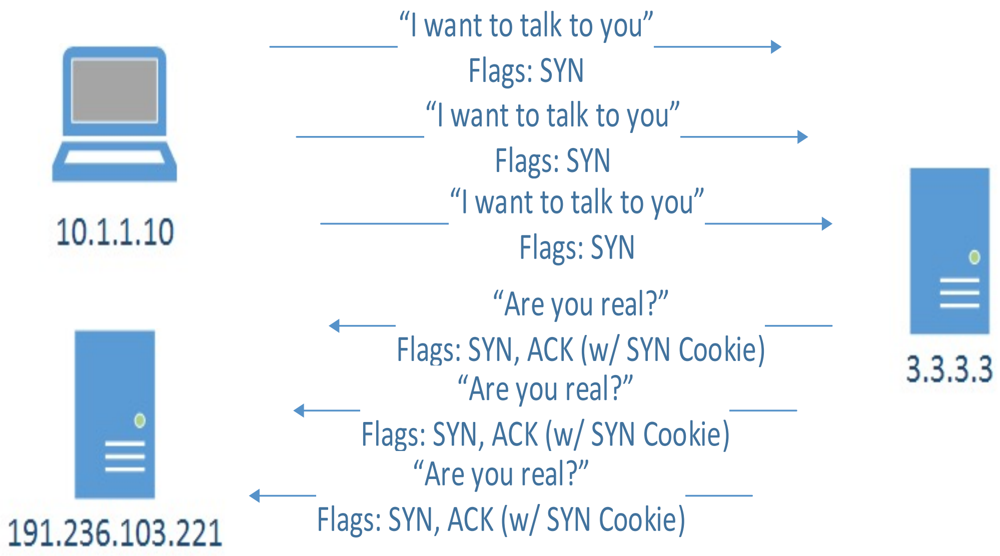


Backscatter

- Traffic that is a byproduct of the attack
- Why is that interesting?
  - It is important to distinguish between the actual attack traffic and unintended traffic sent by the victim
  - Imagine a SYN flood against a "victim" protected by a major scrubbing provider spoofed from IP address
    - What is the traffic to X going to look like?

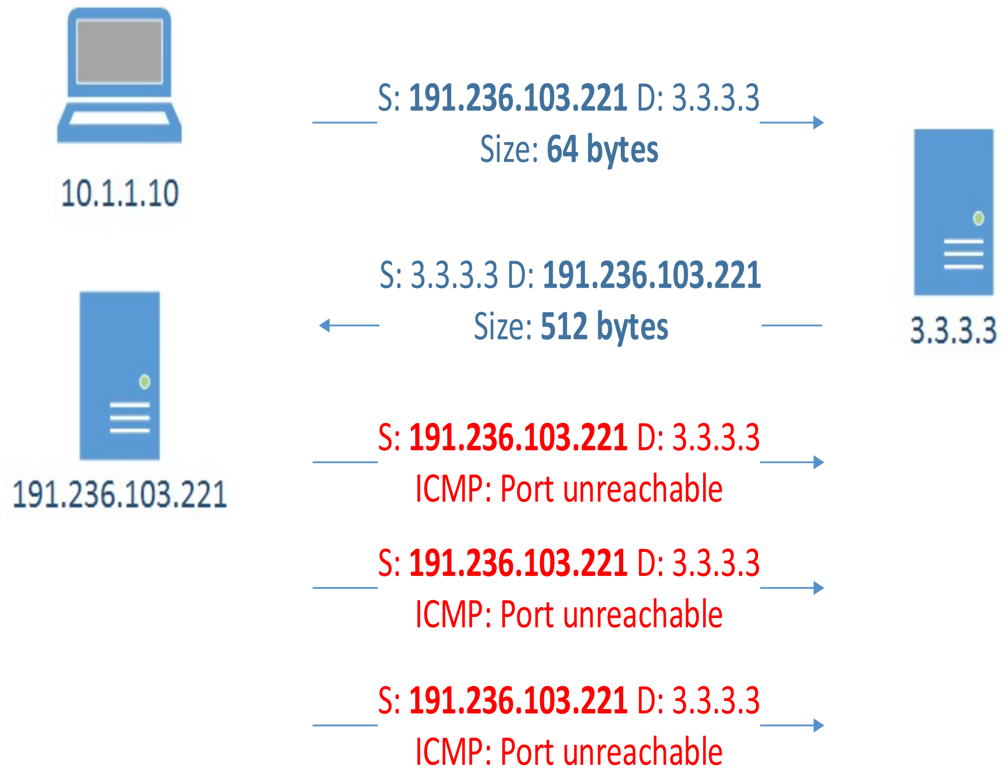
SYN Flood Backscatter?

- Cookie flood ☺



Are you a reflector? (Backscatter)

- In some cases return traffic/backscatter

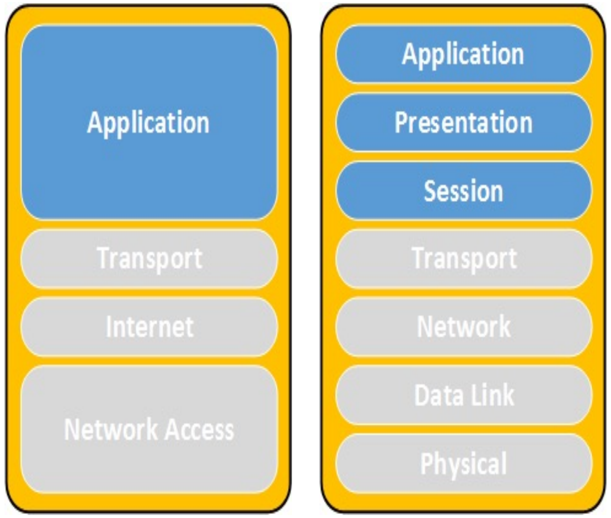


Back scatter on the wire

```
20021 1.756892000  127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
20022 1.756900000  127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
20023 1.756907000  127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
20024 1.756915000  127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
20025 1.756942000  127.0.0.1     127.5.5.5     DNS     153 Standard query response 0x4cb1  A 192.
20026 1.756945000  127.5.5.5     127.0.0.1     ICMP    181 Destination unreachable (Port unreacha

▼Internet Protocol Version 4, Src: 127.5.5.5 (127.5.5.5), Dst: 127.0.0.1 (127.0.0.1)
   Version: 4
   Header length: 20 bytes
  ▶Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
   Total Length: 165
   Identification: 0x4ea9 (20137)
  ▶Flags: 0x00
   Fragment offset: 0
   Time to live: 64
   Protocol: ICMP (1)
  ▶Header checksum: 0x27e4 [validation disabled]
   Source: 127.5.5.5 (127.5.5.5)
   Destination: 127.0.0.1 (127.0.0.1)
   [Source GeoIP: Unknown]
   [Destination GeoIP: Unknown]
▼Internet Control Message Protocol
   Type: 3 (Destination unreachable)
   Code: 3 (Port unreachable)
   Checksum: 0x47d2 [correct]
```

- The victim (127.5.5.5) sends and ICMP port unreachable to the reflector (127.0.0.1)

Cache busting (back to DNS)



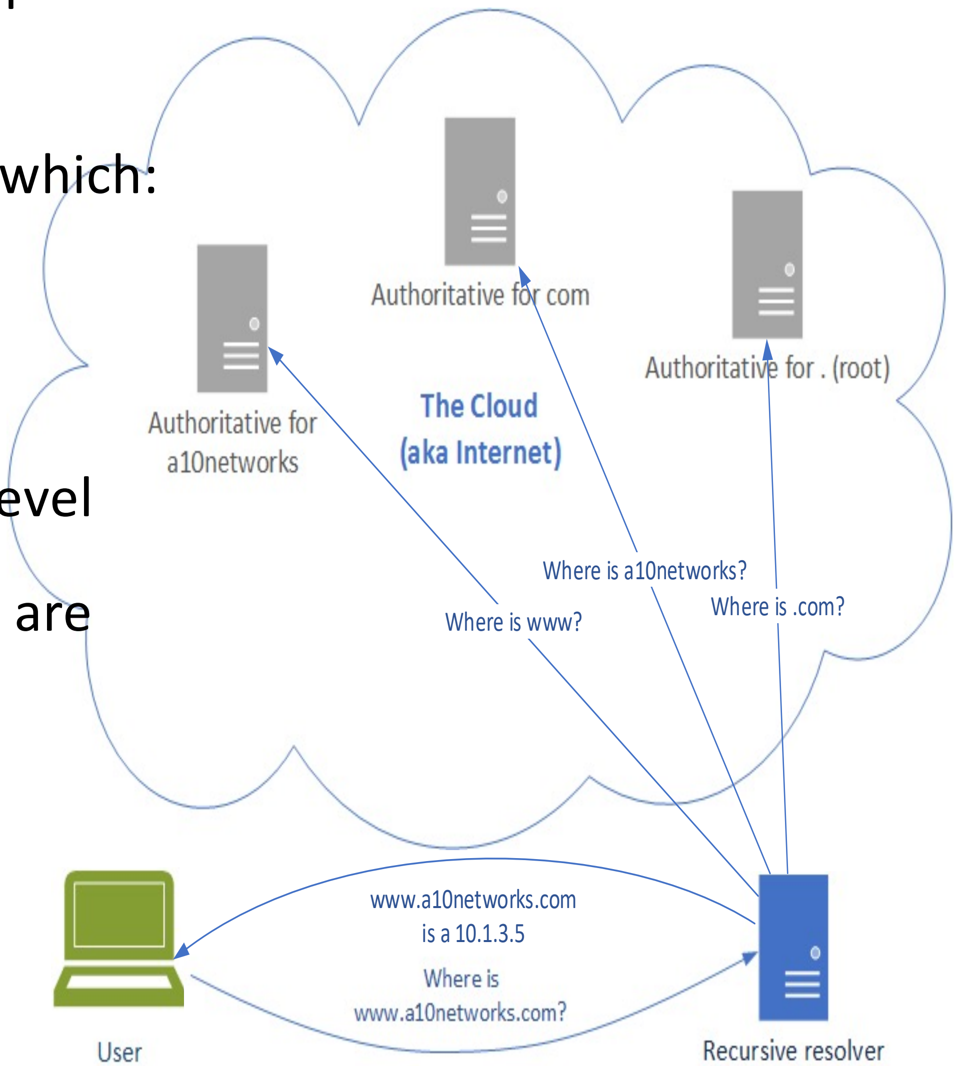
DNS resolution (repeat)

- Let's focus on the number of requests per second

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

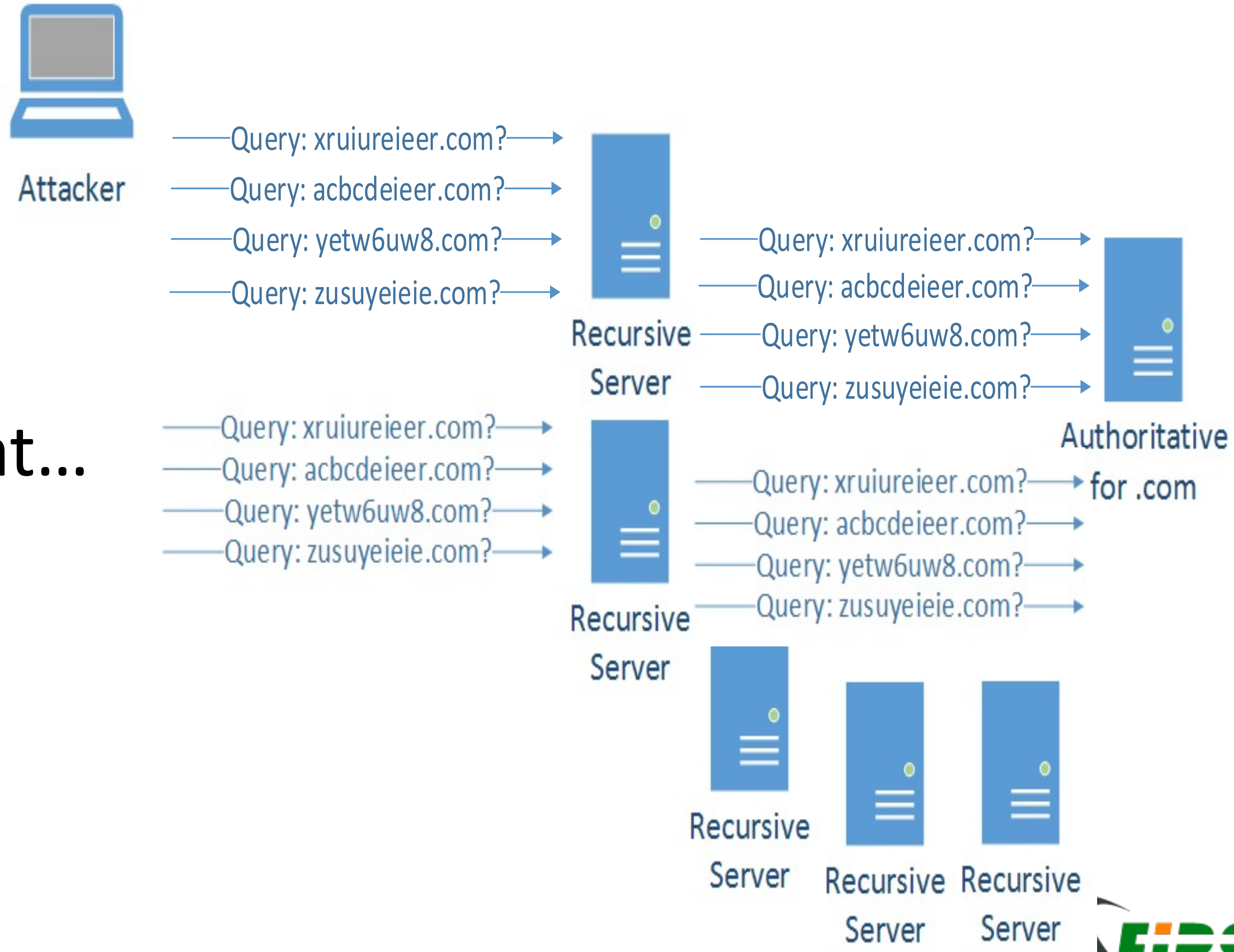
- The recursive talks to different level of authoritative servers, which:
  - Do not cache answers (they are auths)
  - Relatively lower number of queries

- Consider caching and authoritative capacity



What cache busting?

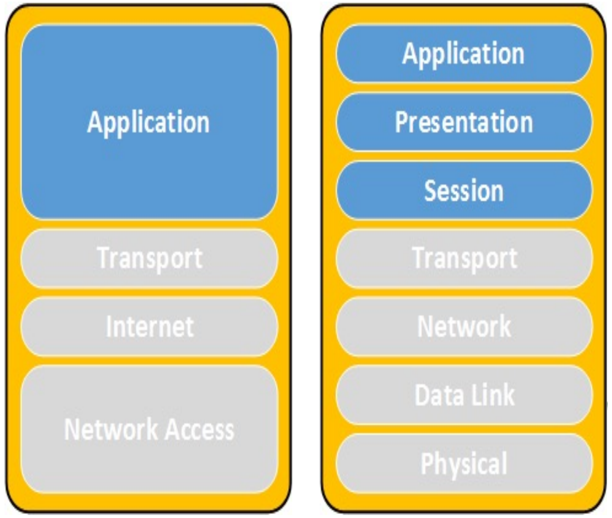
- Attacker sends a query to recursive/reflector
- Recursive forwards the query
- And so on…
- Imagine one more recursive resolver
- Rinse and repeat…



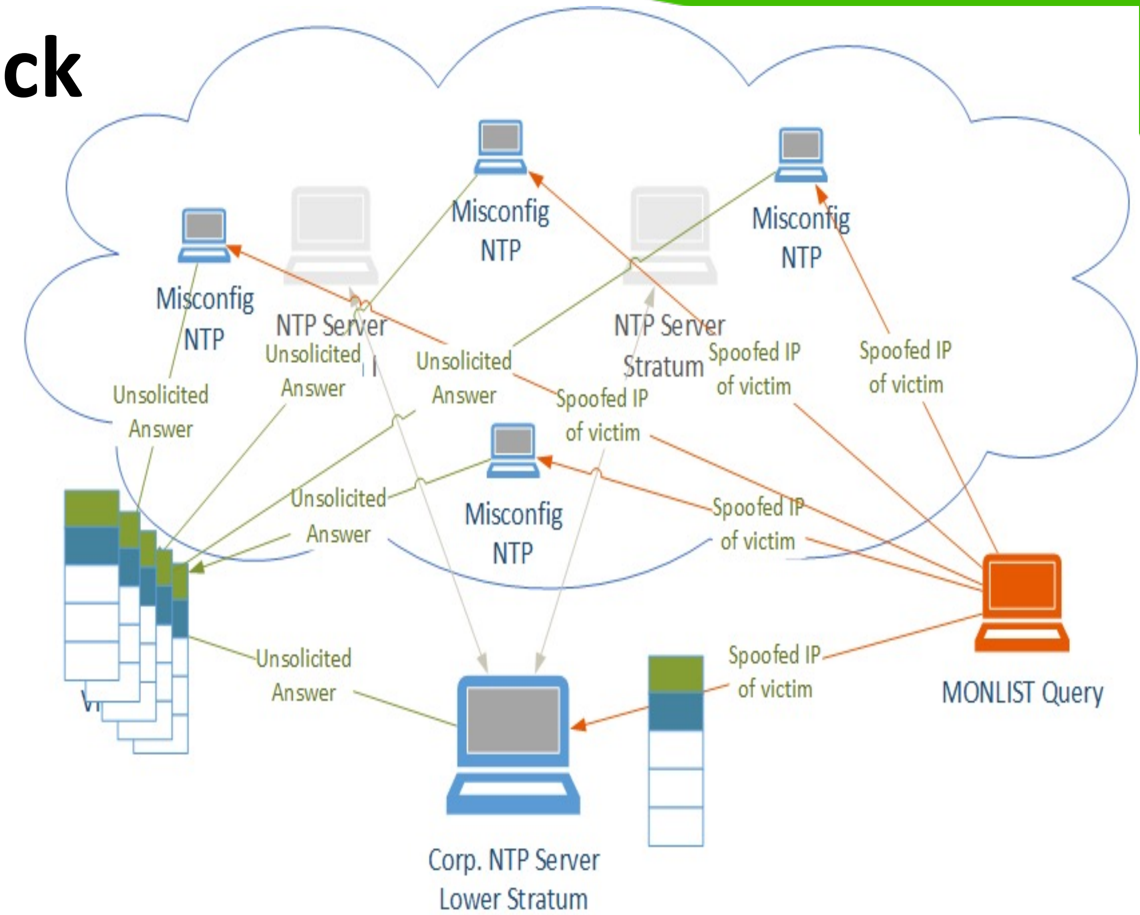
Questions

75%

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/

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

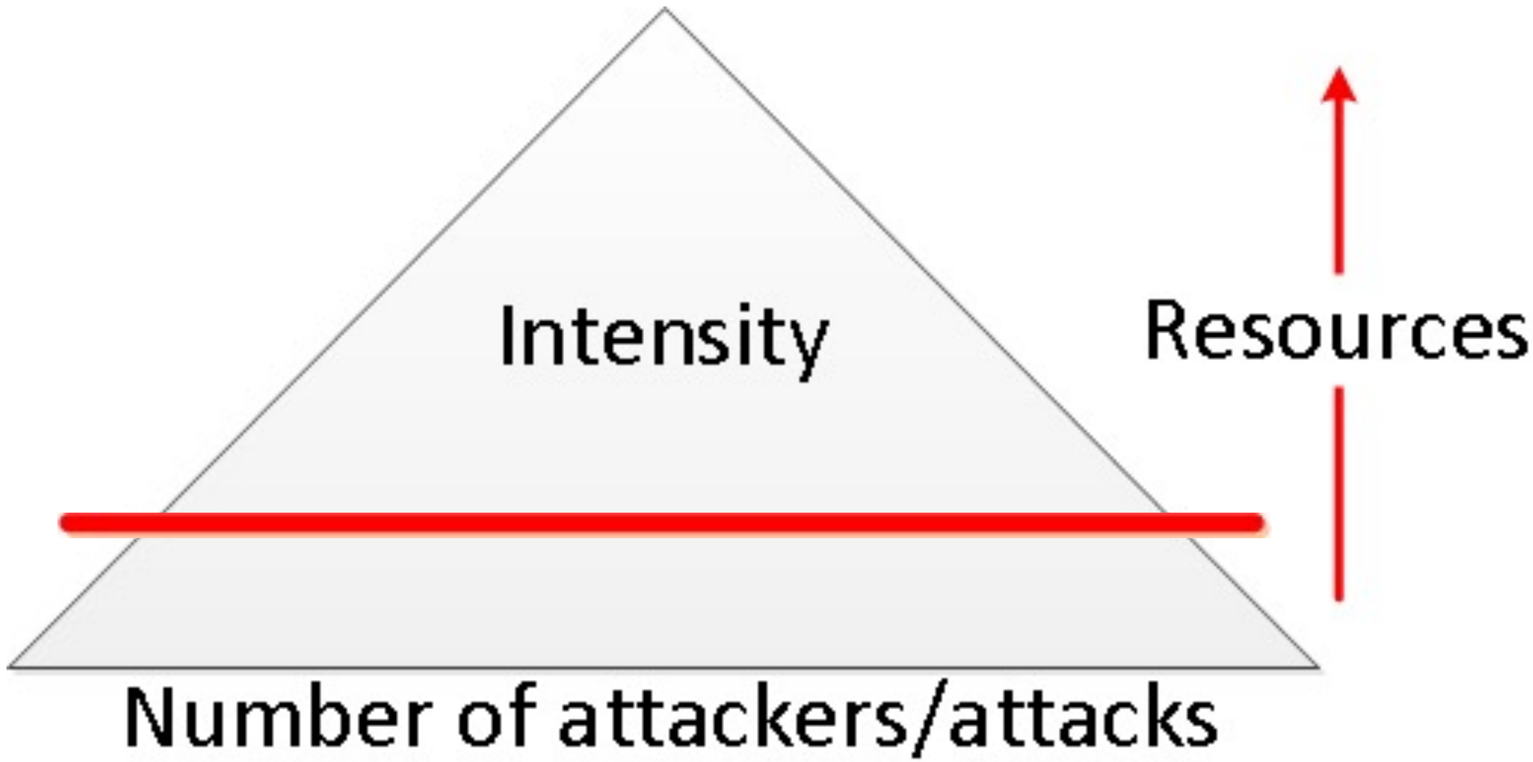
Questions

Mitigation Strategies

Overview

- Risk pyramid
- Value of being online/Outage costs
- Mitigation strategies

Risk Pyramid



The cost of a minute?

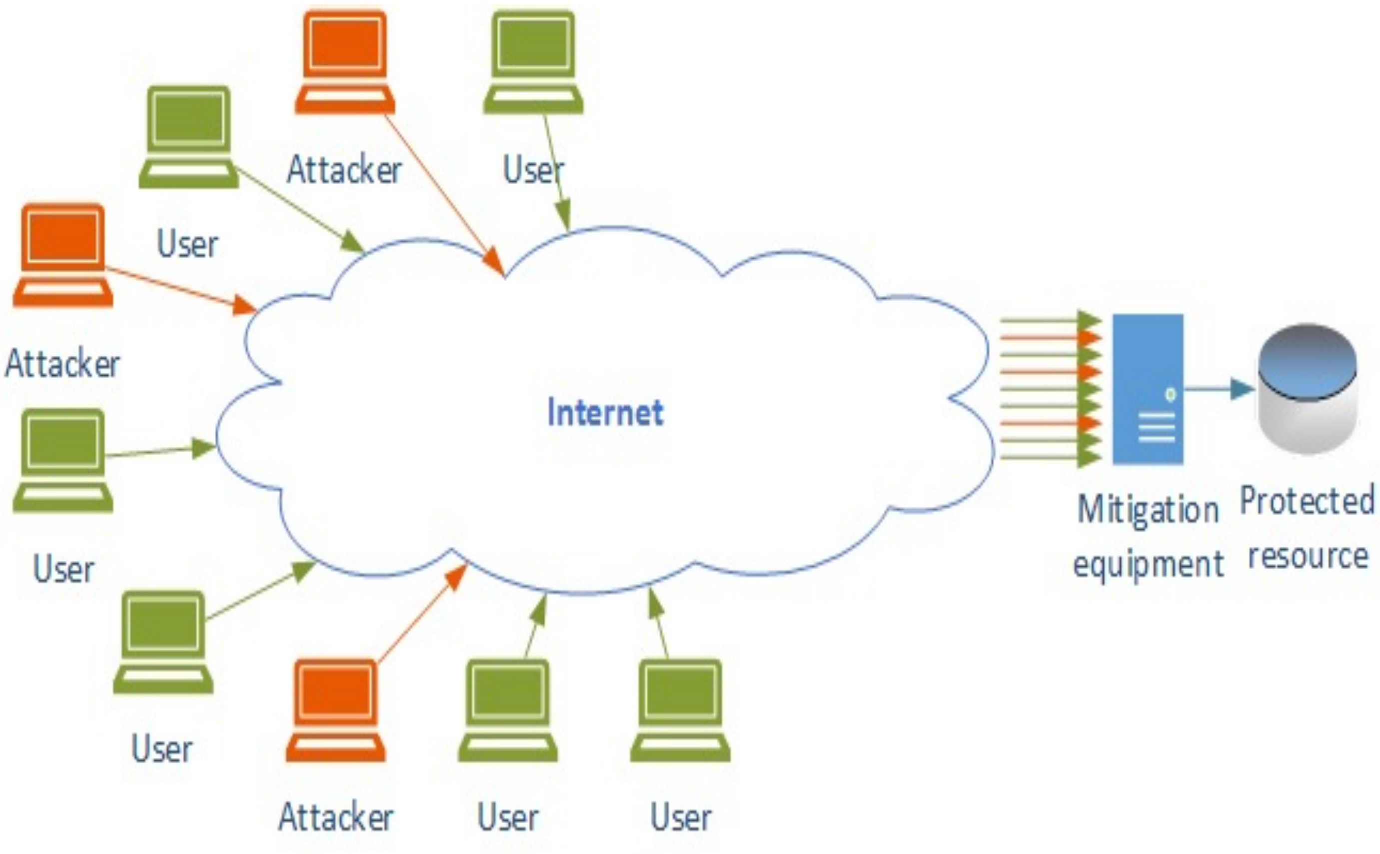
- How much does a minute of outage cost to your business?
- Are there other costs associated with it? Reputation?
- Are you in a risk category?
- How much is executive management willing to spend to stay up?
- Are there reasons you need to mitigate on-site vs offsite? Latency?

Mitigation

Different approaches:

- Do it yourself (DIY)
- Outsource/service
  - On demand
  - Always on
- Hybrid

Do it Yourself (On Premise)



DIY: Considerationss

- Network capacity: bandwidth
- Hardware capacity: packet rates, inspecting headers and content?
- One time cost (refresh every 3-4 years)
- Depending on attacks size can be in $100,000s

DIY: Benefits

- Very low latency
- Can be application specific (non-http, gaming industry)
- Better control of the mitigation
- If inspecting TLS traffic keeps the keys in the company

DIY: Drawbacks

Network capacity:

- Fluctuates
- How much do you over provision? Double, triple, ten times?

- Need to procure
  - bandwidth - monthly recurring - expensive, adds up
  - compute and network hardware
  - qualified personnel – hard to find; expensive; hard to retain

DIY: Bottom line

- traffic – 10GBps = $2,000/mo (NA)
- colocation space - $400/mo
- power – depends on equipment and location
- equipment – min $20,000 per 10GBps port
- personnel – go figure… ☺

…and you need them in many locations, with multiple per location.

DIY: Conclusions

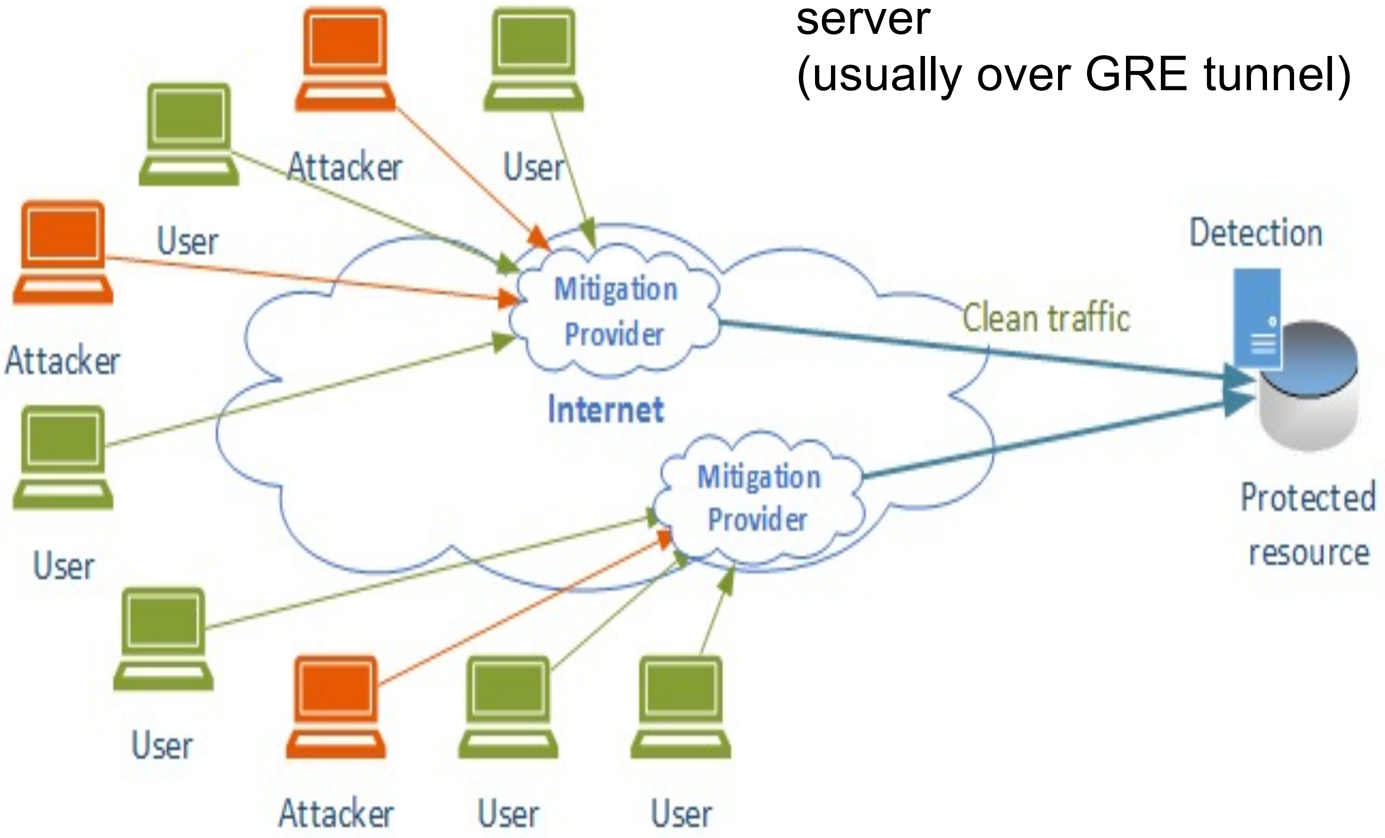
- At present DDoS attacks are at a very large scale but DIY is not easy to scale for small and medium networks
- Leverages economy of scale – requires a large infrastructure
- Infrastructure is very expensive to build and maintain
- Requires significant amount of know-how
- Unless hosting a very large site it's better left to the professionals

External service

- DDoS mitigation service providers and CDNs
- Pricing:
  - based on size of attack
  - based on clean traffic
- Operating model:
  - on demand
  - always on

On Demand DDoS

- Target: detect and signal the mitigation provider
- Mitigation provider: Inject BGP routes
- Traffic is redirected to the mitigation provider
- Clean traffic is delivered to the origin server
  (usually over GRE tunnel)



On Demand Mitigation - benefits

- Scales up very easily
- Since most applications are HTTP/S based, it is compatible with them
- Easier to deploy
- May leave the target vulnerable to bypass

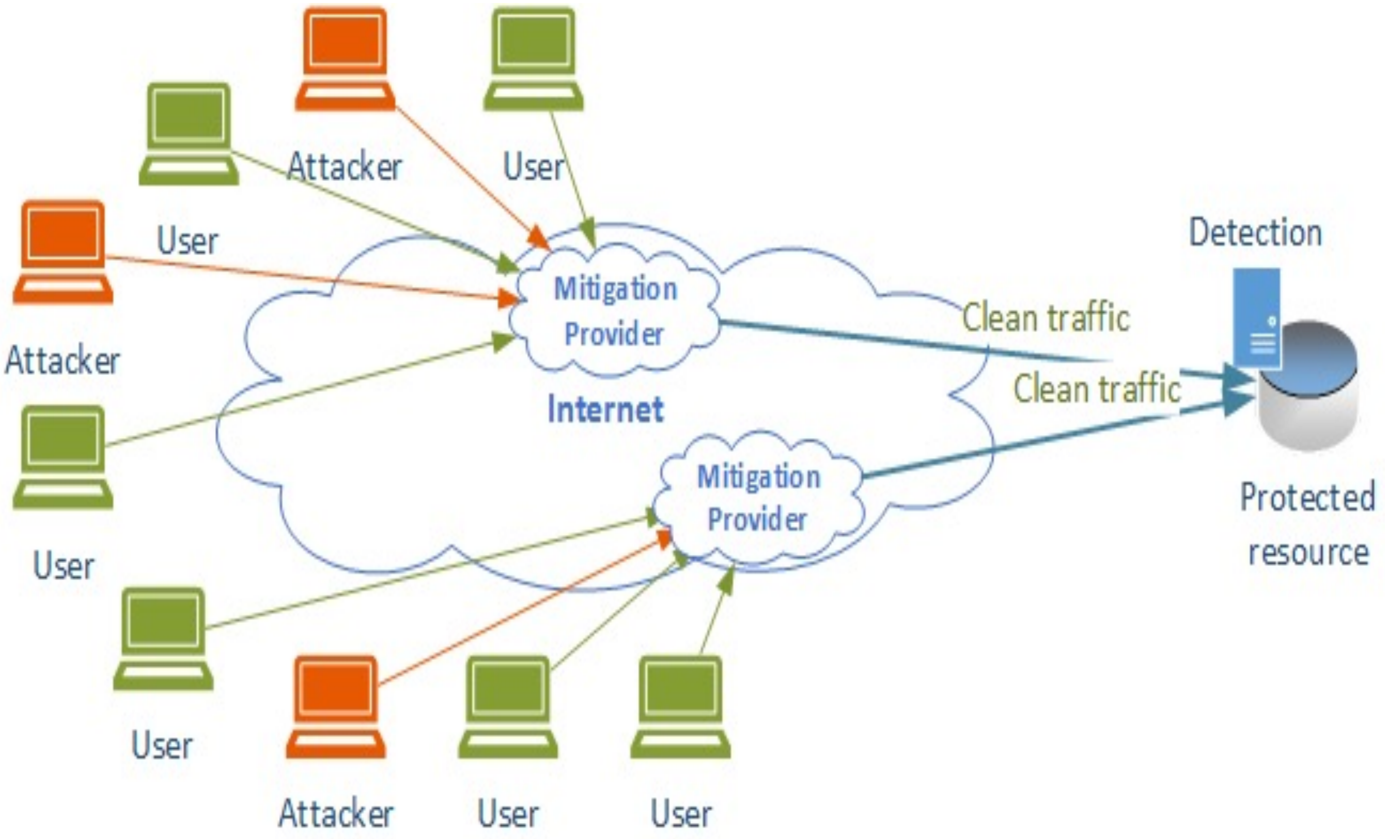
On Demand Mitigation - drawbacks

- Takes time between the site being attacked until it switches to the service provider
- Potential outages
- Difficult to establish TLS
- May have increased latency
- Target may still be exposed
- Detection is not Application Aware
- GRE Tunnels create complexity

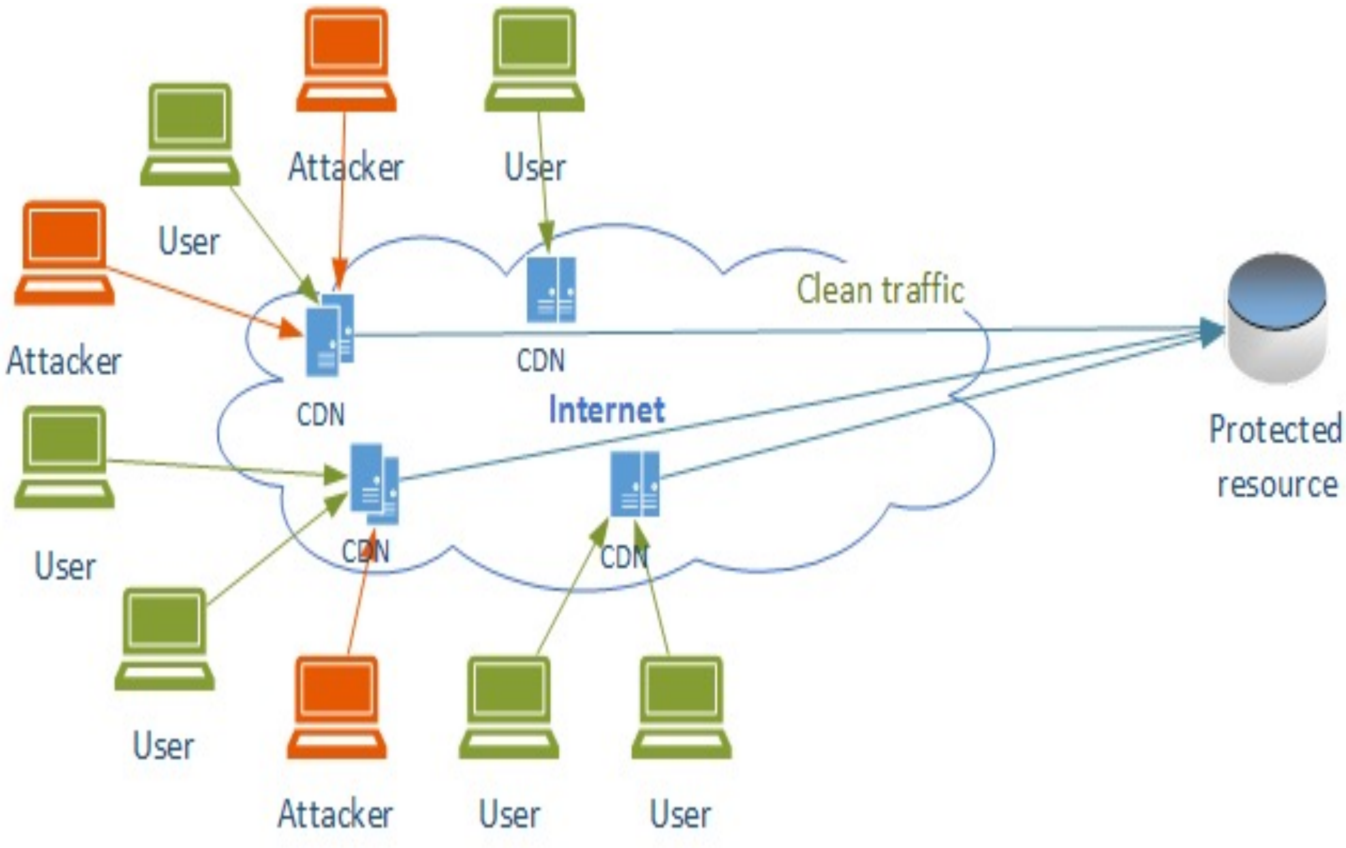
Always On Mitigation

- Permanently serve the customer space
  - Advertise IP address space
  - Use shared delivery infrastructure (CDN)
- Traffic is always flowing through the mitigation systems
- Usually combined with services like CDN, which further increases website performance (even during peace time)

Always On DDoS Mitigation (advertise IP space)



Always On DDoS Mitigation (CDN)



Always On Mitigation - benefits

- Scales up very well during volumetric attacks
- Mitigation can be virtually instantaneous
  - No moving parts during the attack
- Can protect most applications
- Once it’s on there are no moving parts
- Very hard to bypass
- (proxy/caching) If deployed properly, it may improve website performance
- Cost depends on the website traffic (not the attack)

Always On Mitigation - drawbacks

- Can increase latency
- Challenges around TLS
- Stale caches
- May be much more expensive

Hybrid

- Combination of DIY and service providers
- Helps customers manage their risk profile in a more flexible way

Benefits:

- Provides protection against large scale events without the added service cost
- Allows for escalating response postures and risk/finance management
- Overall most of the benefits of On Demand

Drawbacks:

- Increased complexity
- Requires skilled personnel
- May have interoperability issues

DDoS mitigation service providers

- It is an ongoing expense
- Depending on the business model it can be big or small
- Hides the complexities of managing the problem
- May introduce latencies, but also may accelerate content if used properly

DDoS mitigation svc providers – bottom line

- Depends on the exact setup

  - in CDN cases usually depends on the size of normal traffic and not the size of the attack

  - varied: $50/month – thousands…

DDoS mitigation service providers

- Pros
- Hides the complexities of managing the problem
- May accelerate content delivery
- May be much cheaper, especially as attack sizes grow but are not common
- Cost: much, much lower than DIY

- Cons
- May not be applicable to all applications - gaming
- May increase latency
- May end up expensive
- Third party sees the users (and maybe the content) - privacy, security
- Issues with stale cache

Questions

FIRST™

Improving Security Together