Fundamentals of DDoS Mitigation

Krassimir Tzvetanov
krassi@fastly.com
NANOG 72

Atlanta, GA

Introduction and overview

Overview

- Discuss what DDoS is, general concepts, etc.
- High level mitigation models
- Discuss reflection and amplification
- Attacks you need to be familiar with?
  - SYN Flood
  - Sloworis
  - DNS, NTP reflection
  - DNS cachebusting

What is DoS/DDoS?

What is Denial of Service?

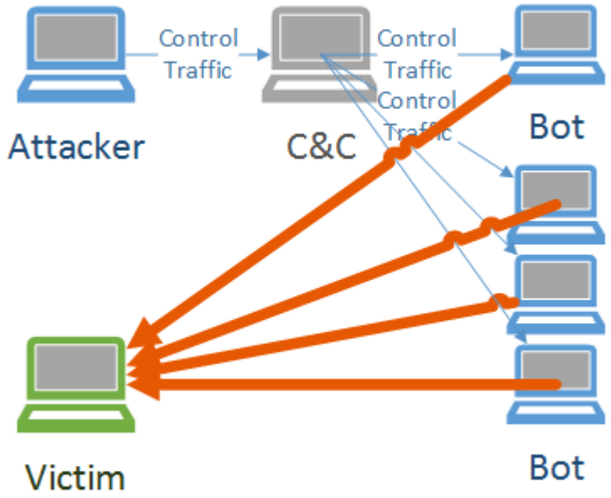
- Discussion
- Resource exhaustion… which leads to lack of availability
- Consider:
  - How is it different from The Guardian pointing to somebody's web site?
  - How is that different from company's primary Internet connection going down?

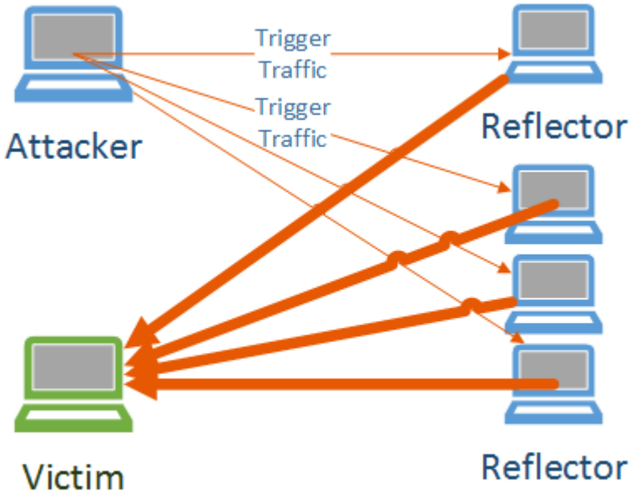
What is Denial of Service?

- From security point of view?
  - Decreased availability
- From operations point of view?
  - An outage
- From business point of view?
  - Financial losses

DoS vs. DDoS

- 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 adversary?

Composition

- Wide range of attackers
  - Gamers – on the rise!!! ☺
  - Professional DDoS operators and booters/stressors
  - Some of the attacks have been attributed to nation states
  - Hacktivists – though not recently

  …and more

Motivation

- Wide range of motivating factors as well
  - Financial gain
    - extortion (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** or financial fraud)
  - Disable firewalls
  - 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
  - Mirai

Additional factors

Additional factors

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

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/
  - Mirai

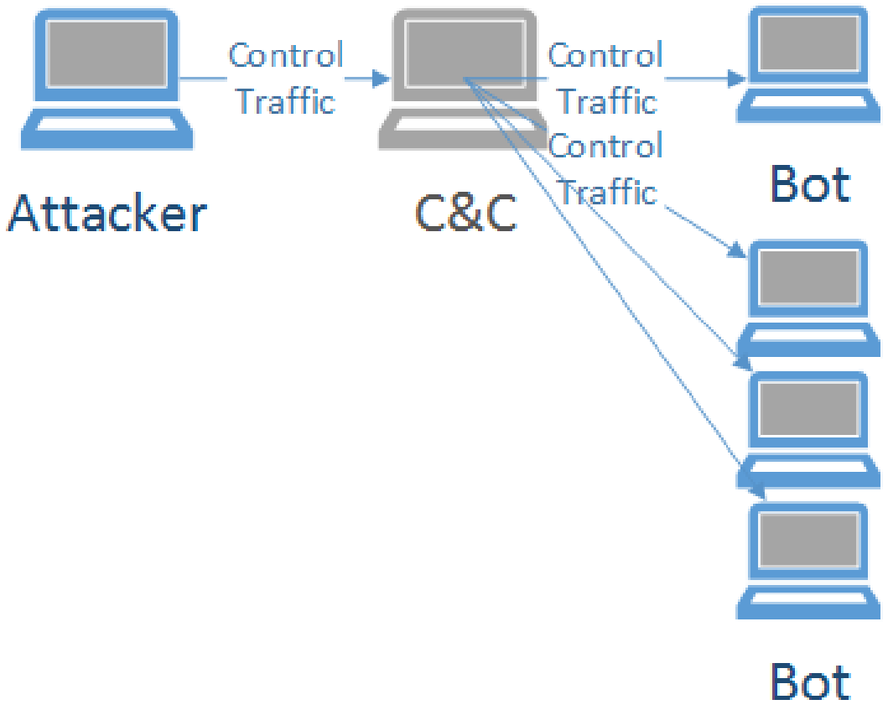
Compromised CMSes

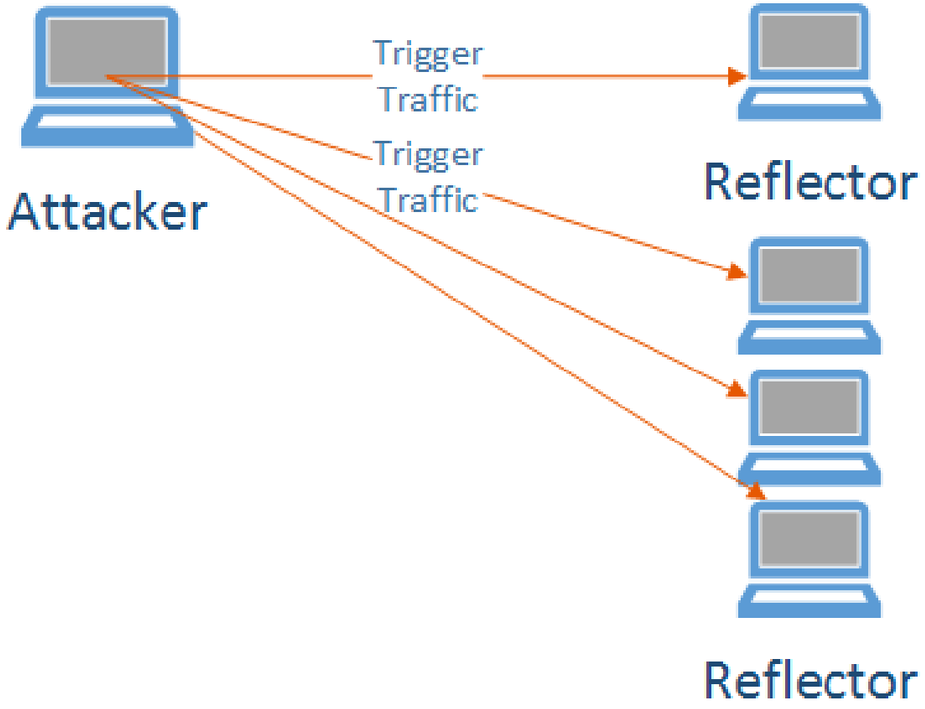
- Most targeted Content Management Systems:
  - WordPress
  - Joomla
- Started in early 2013
- Started with a particular group of people abusing it
- Now it is an easy way to build a botnet and other groups abuse it as well

Booters/Stressors

- Inexpensive
- Tools are sold for cheap on the black market (forums)
- Range 5-10 Gbps and up to 40GBps
- Usually short duration
- Popular among gamers

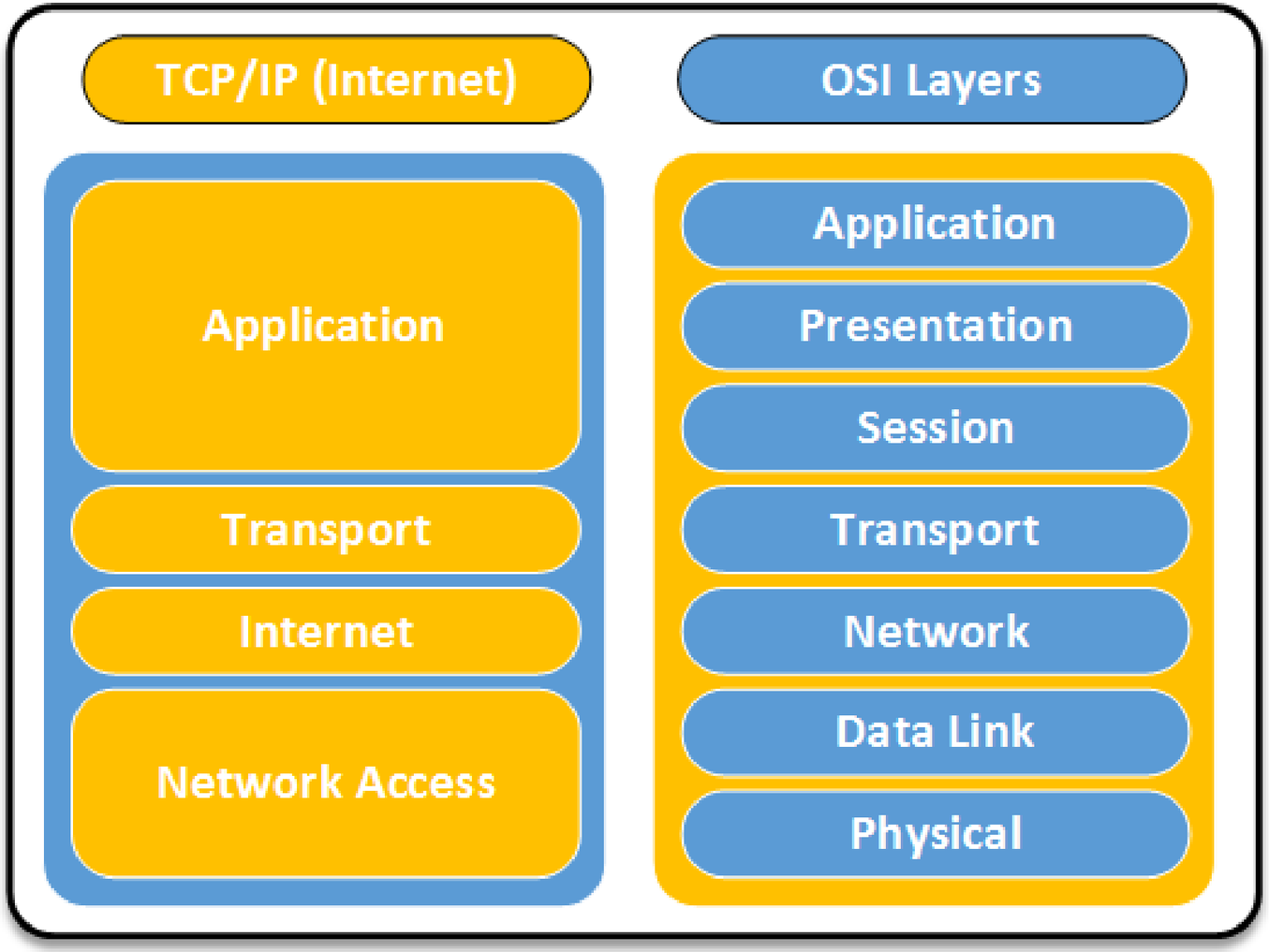
Low cost thanks to reflection





Attack surface

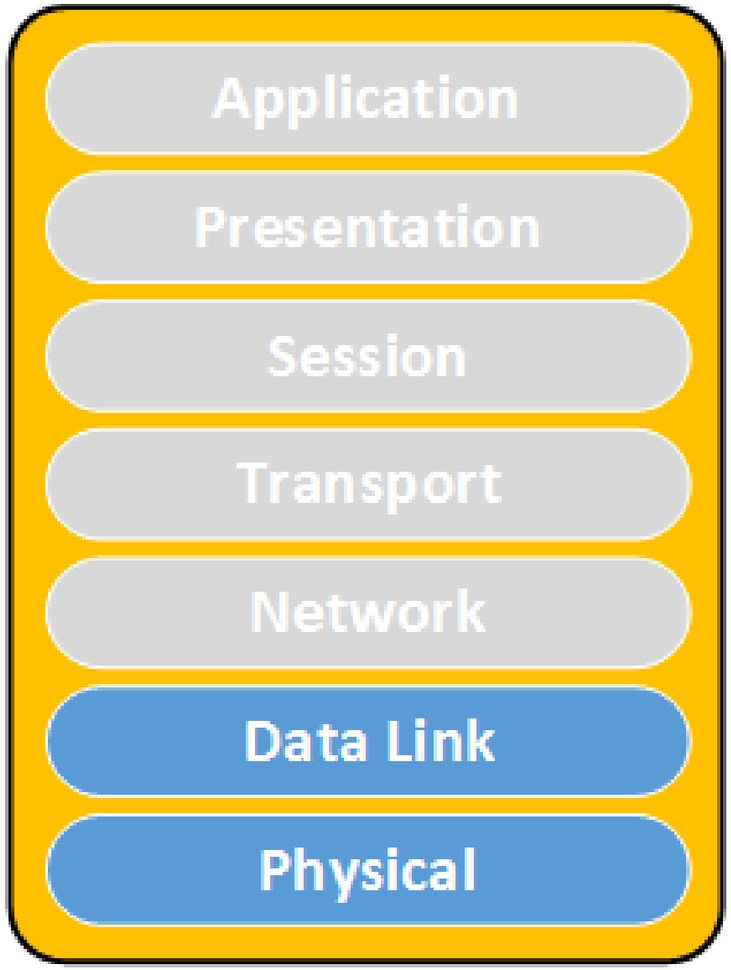
Network Layers – OSI vs Internet Model



Physical and Data-link Layers

- Cut cables
- Jamming
- Power surge
- EMP

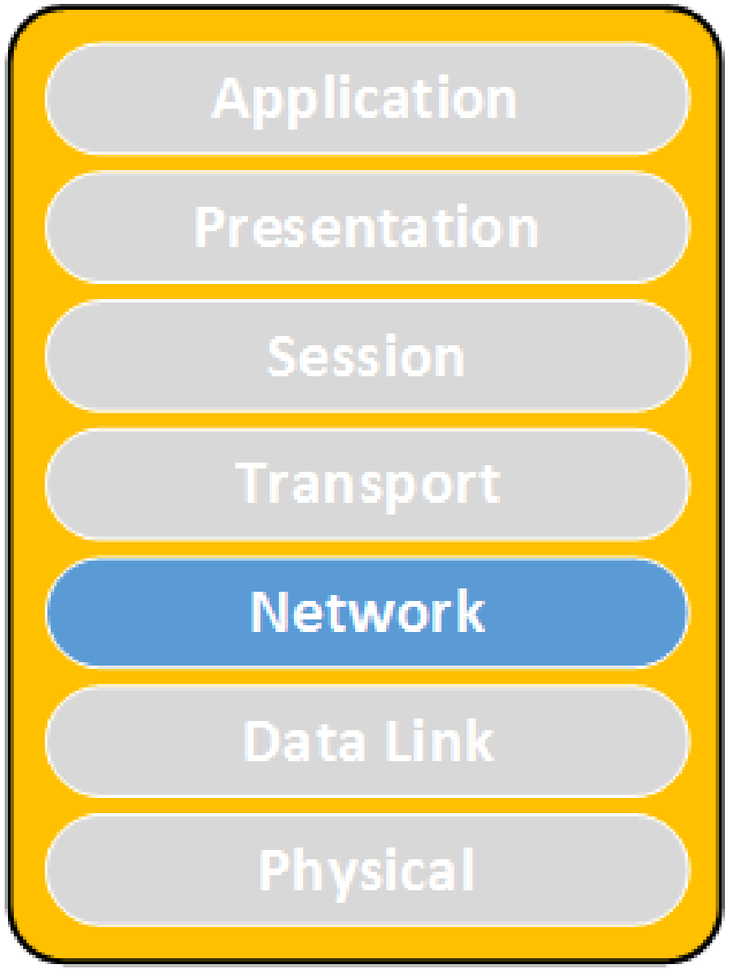
- MAC Spoofing
- MAC flood



Network Layer

- Floods (ICMP)

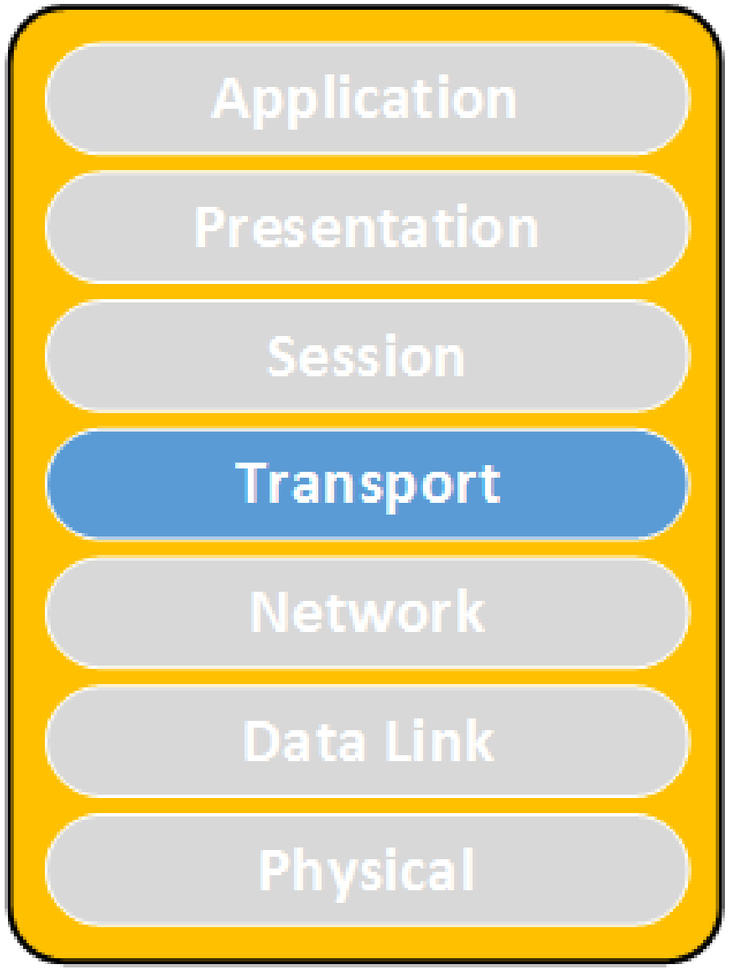
- Teardrop
  (overlapping IP segments)



Transport Layer

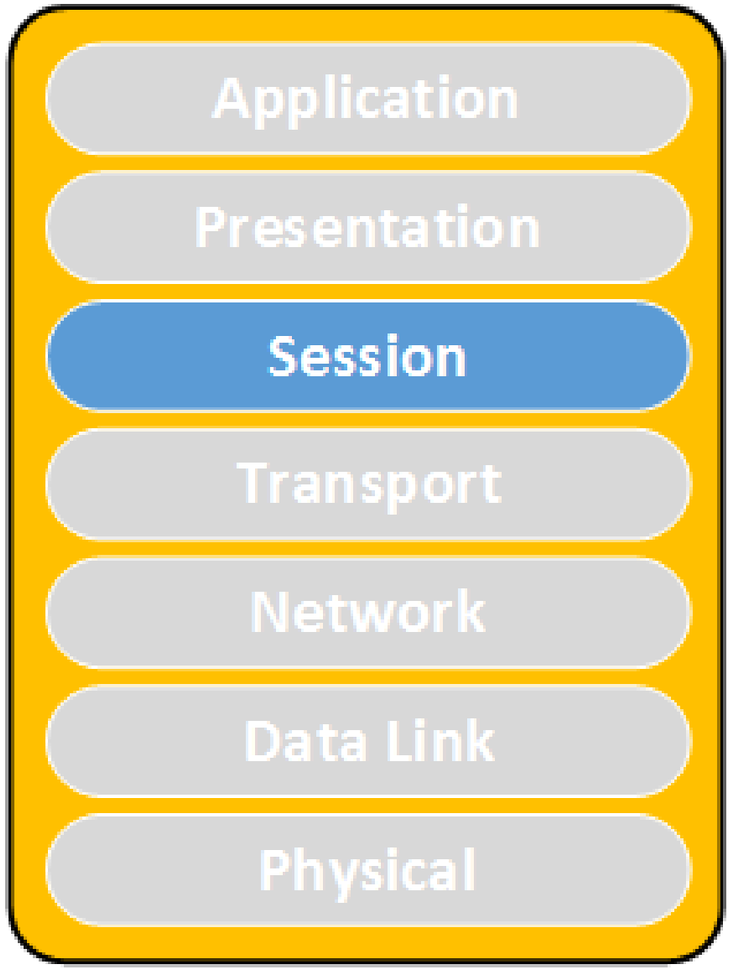
- SYN Flood
- RST Flood
- FIN Flood
- You name it…

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



Session Layer

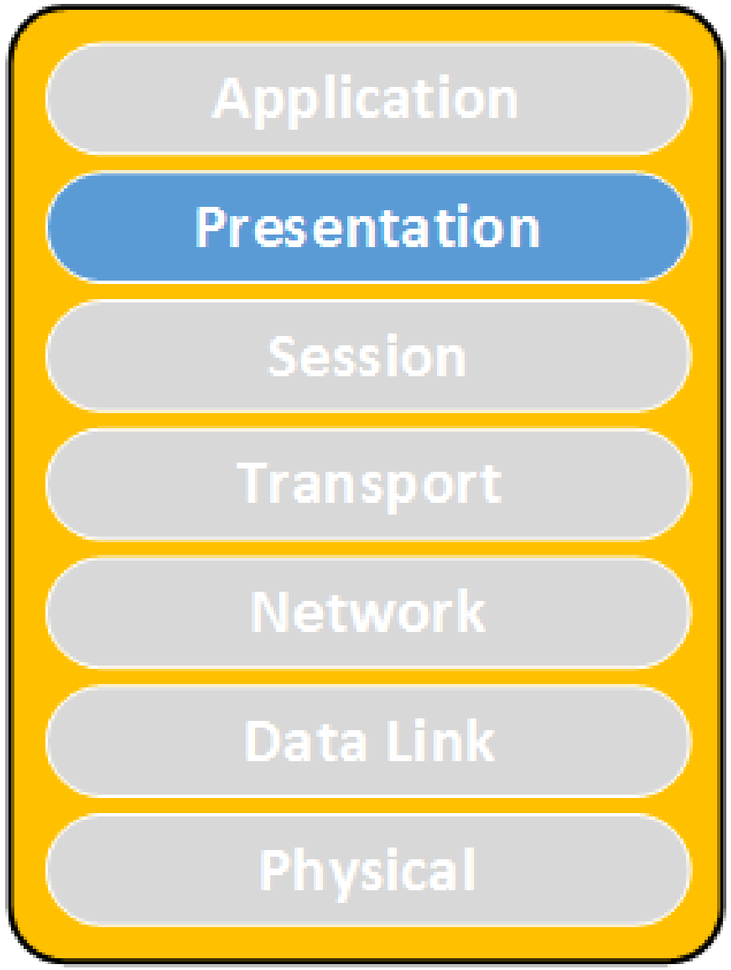
- Slowloris
- Sending data to a port with no NL in it (long headers, long request lines)
- Send data to the server with no CR



Presentation Layer

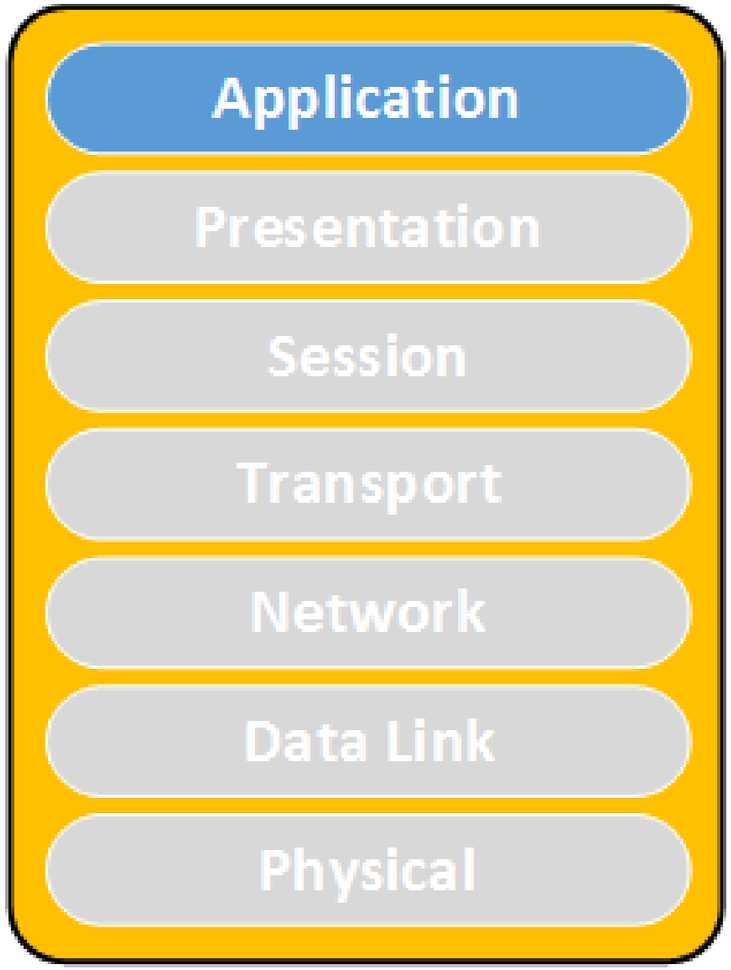
- Expensive queries (repeated many times)
- XML Attacks

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



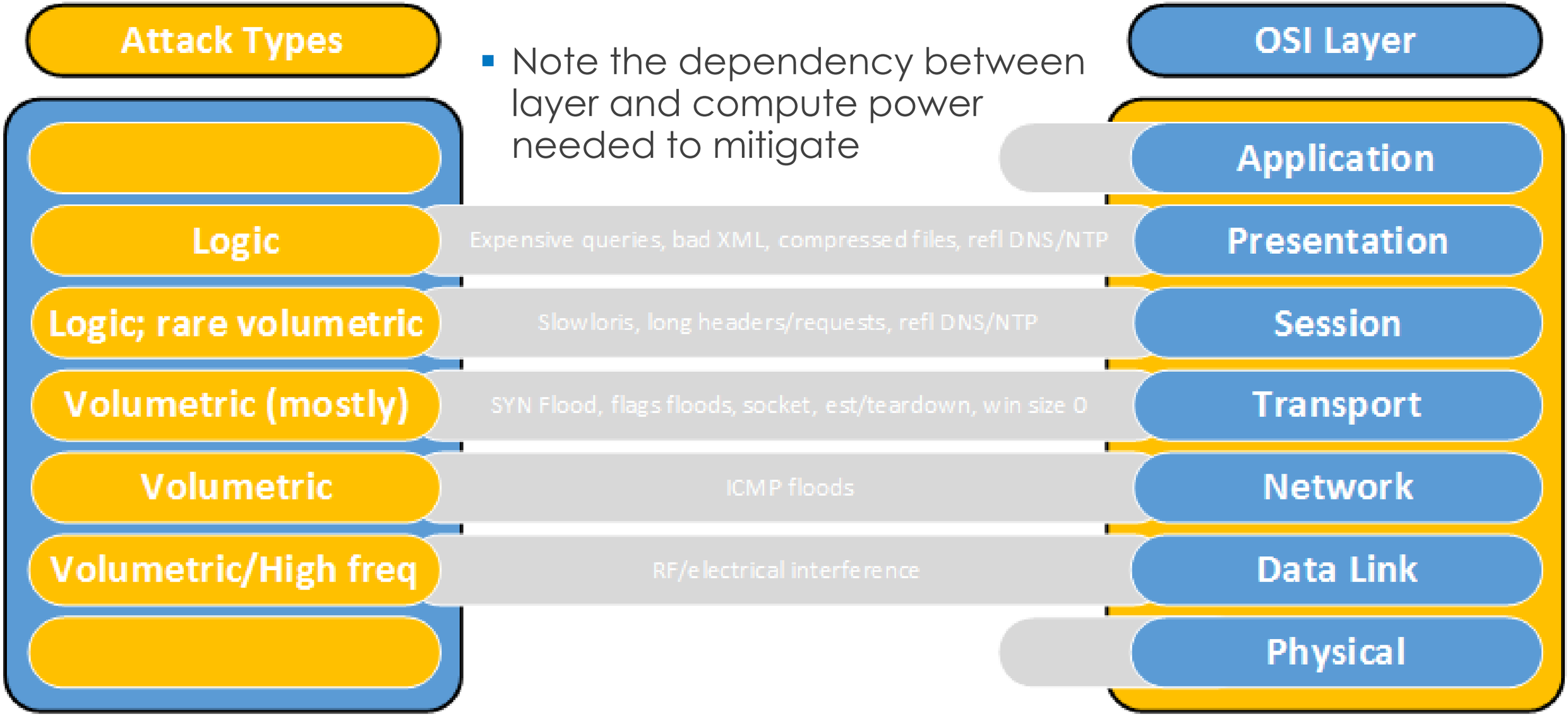
Application Layer

- Depends on the application
- Black fax



Attack summary by layer

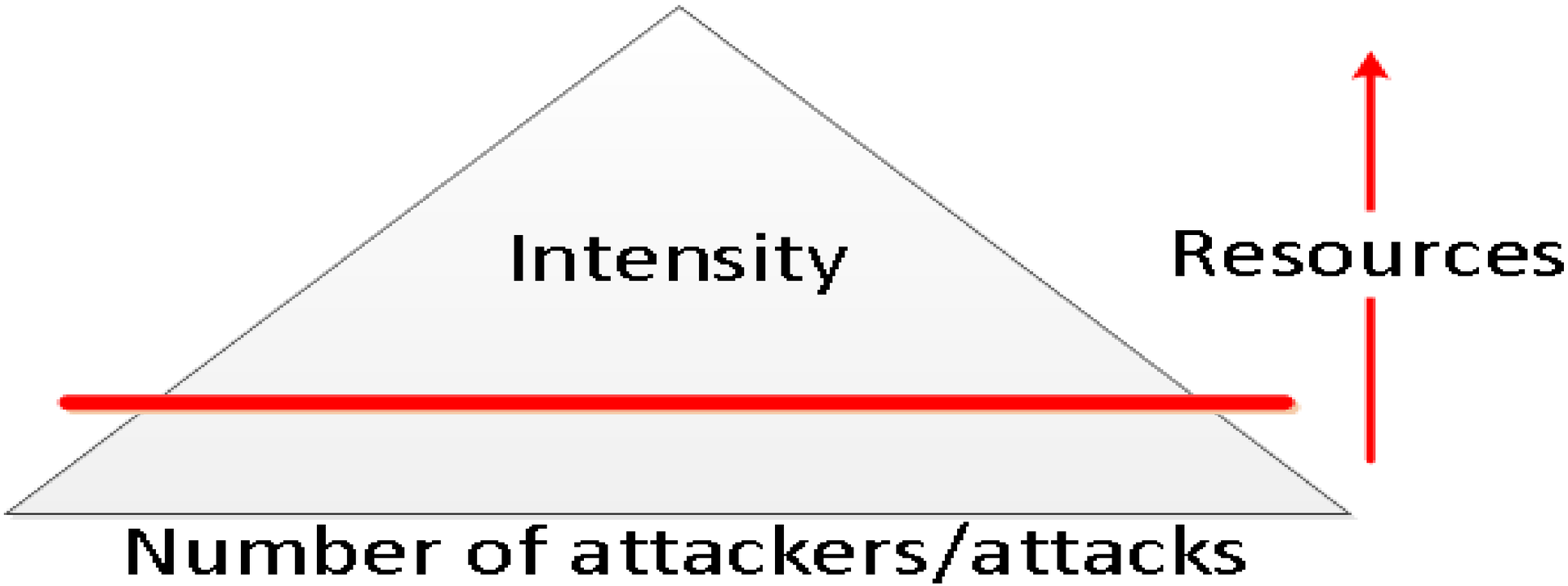
- Note the dependency between layer and compute power needed to mitigate



Questions?

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?

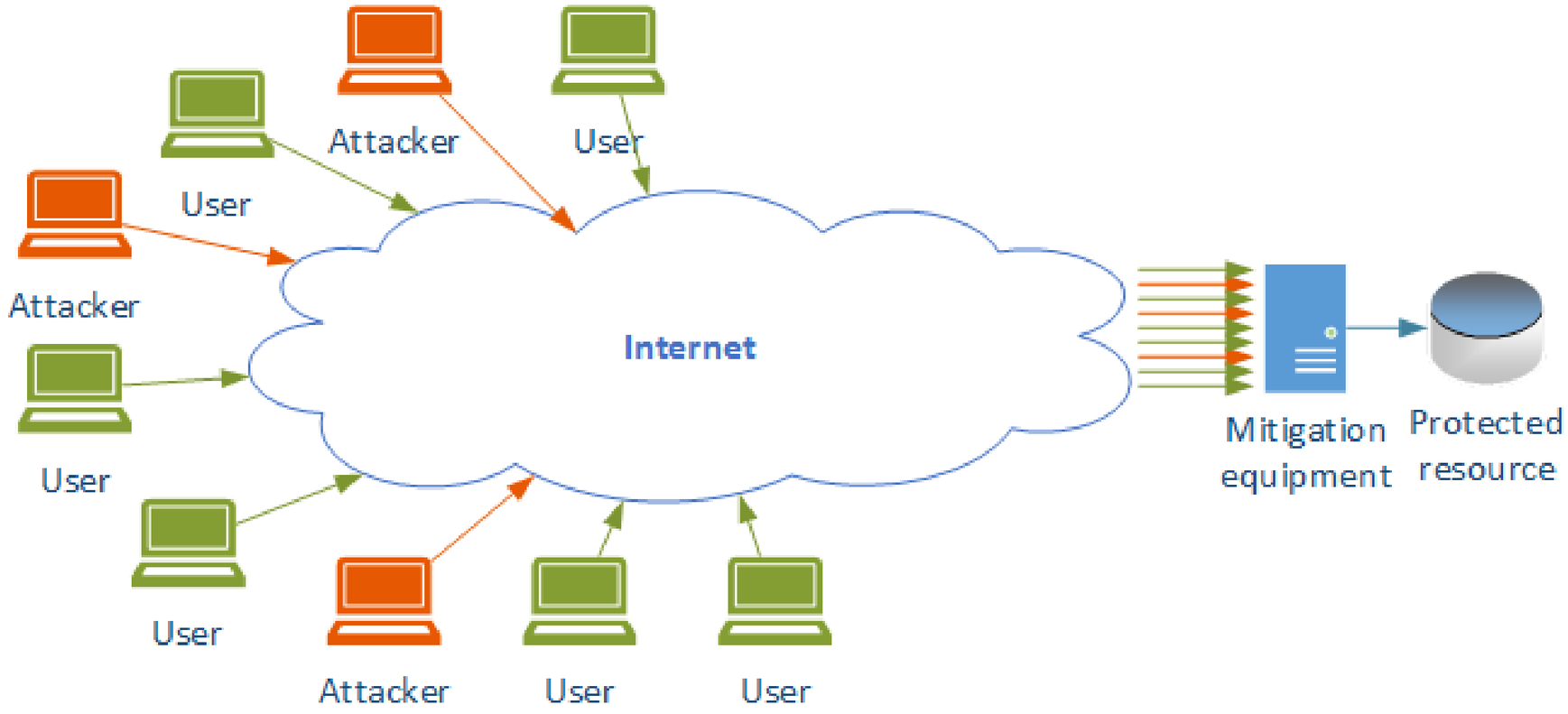
- Are there reasons you need to mitigate on-site vs offsite? Latency?

Mitigation

Different approaches:

- Do it yourself (DIY)
- Outsource/service
- Hybrid

Do it Yourself (On Premise)



DIY: Considerations

- 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

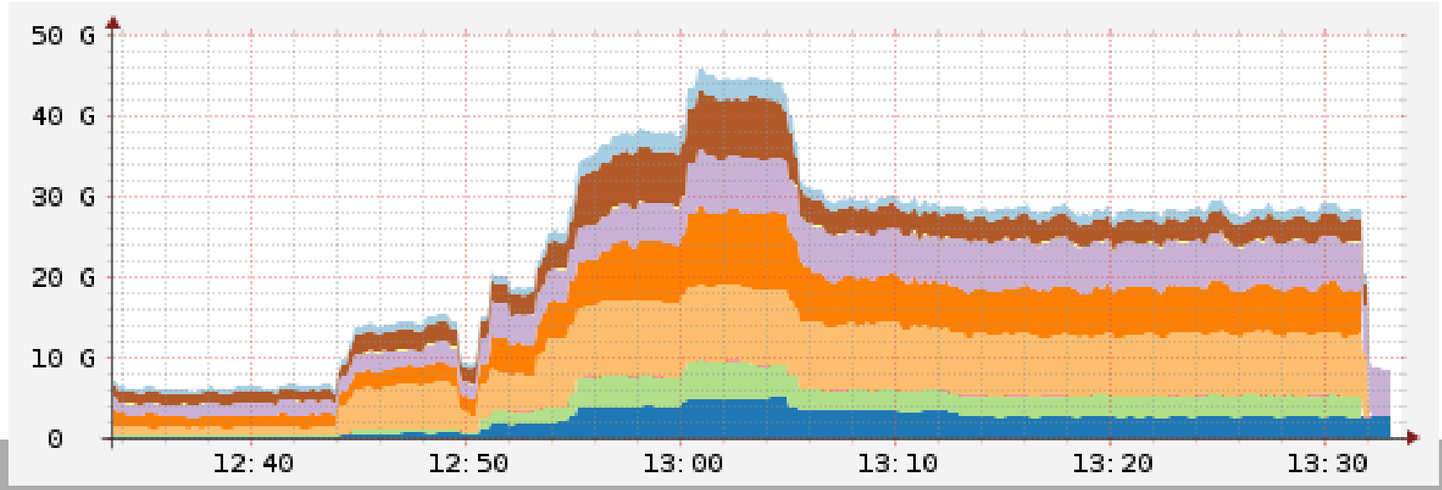
DIY: Benefits

- 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

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

How much bandwidth do you need?
Double, triple, ten times?



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

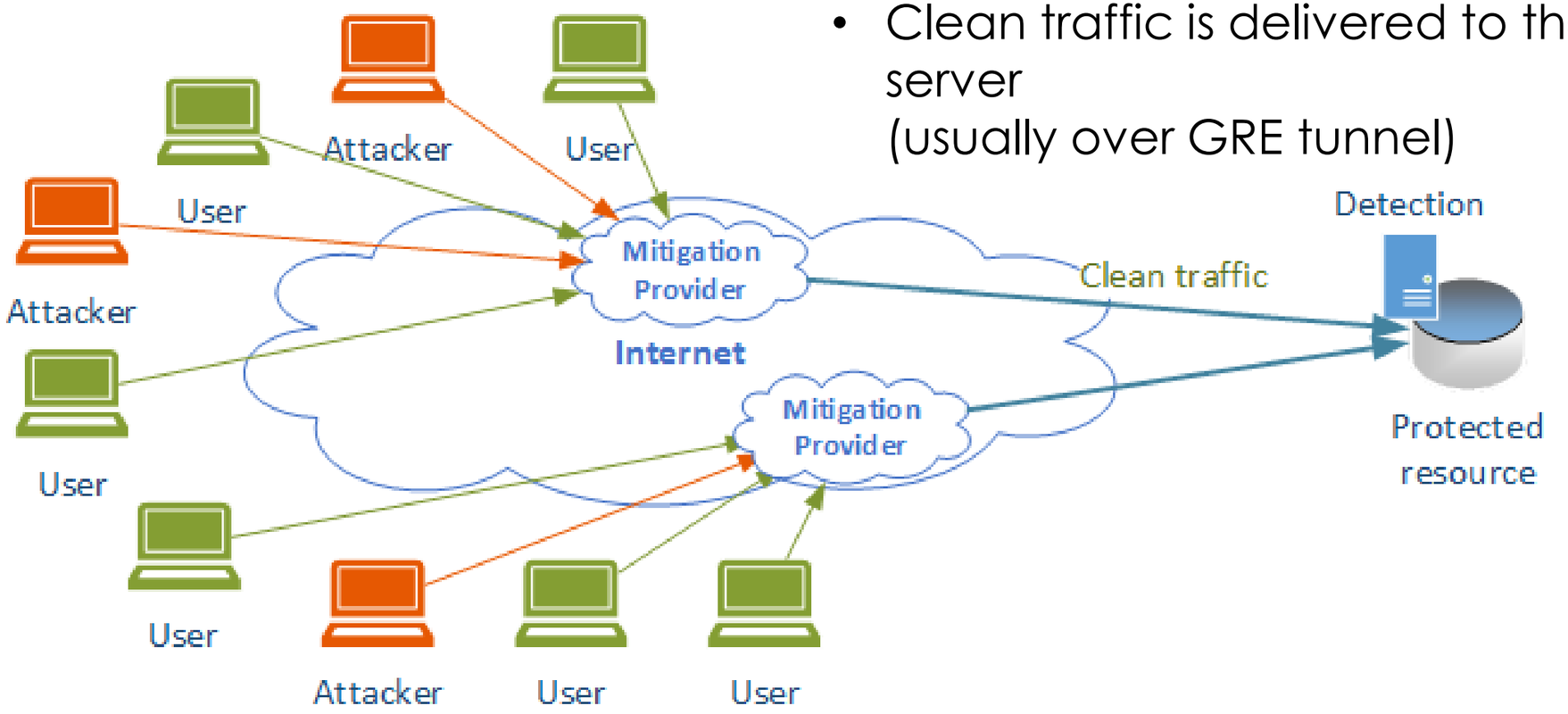
- At present DDoS attacks are at a very large scale but DIY is not easy to scale for small and medium networks
- Leverages economy of scale – requires a large infrastructure
- Infrastructure is very expensive to build and maintain
- Requires significant amount of know-how
- Unless hosting a very large 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
- Can protect most applications from volumetric attacks
- Easier to deploy
- May leave the target vulnerable to bypass

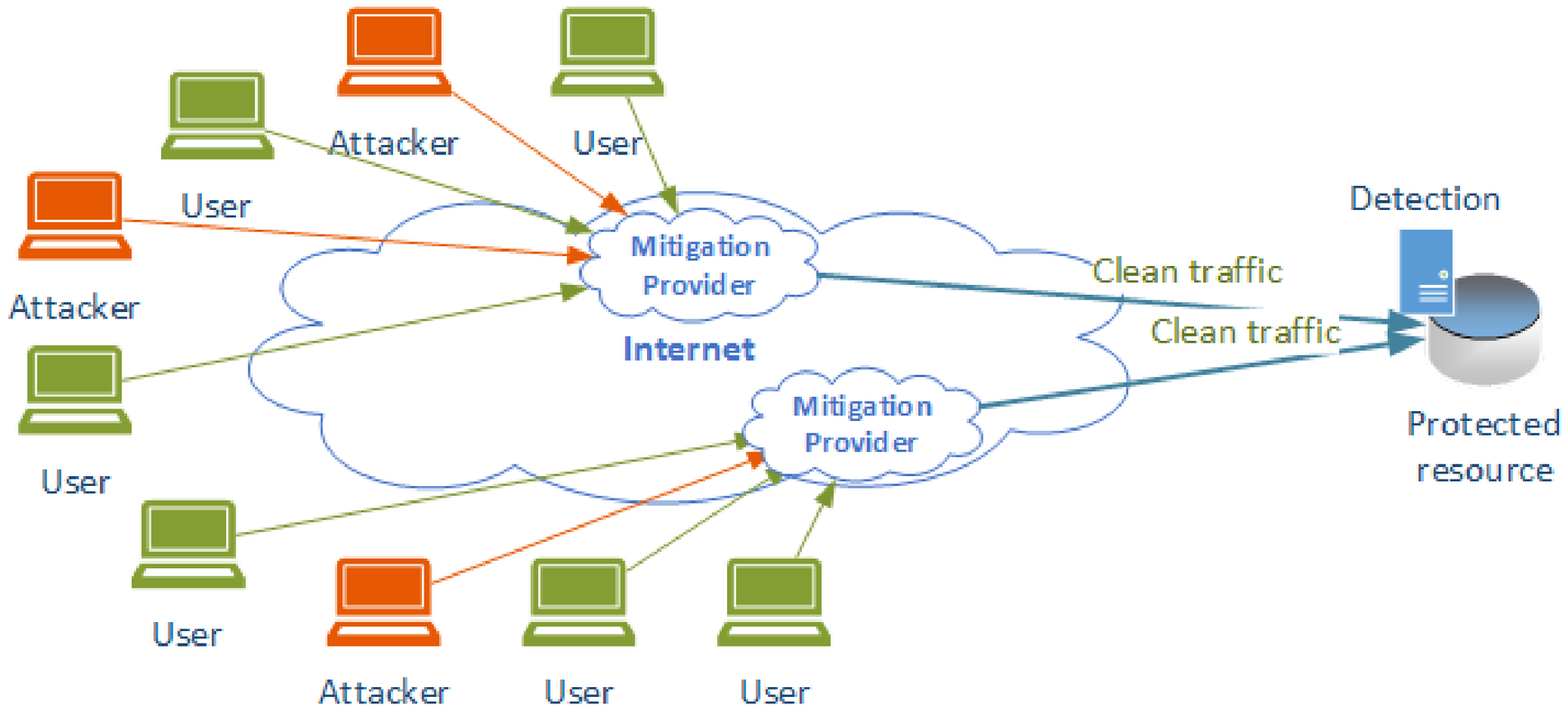
On Demand Mitigation - drawbacks

- 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

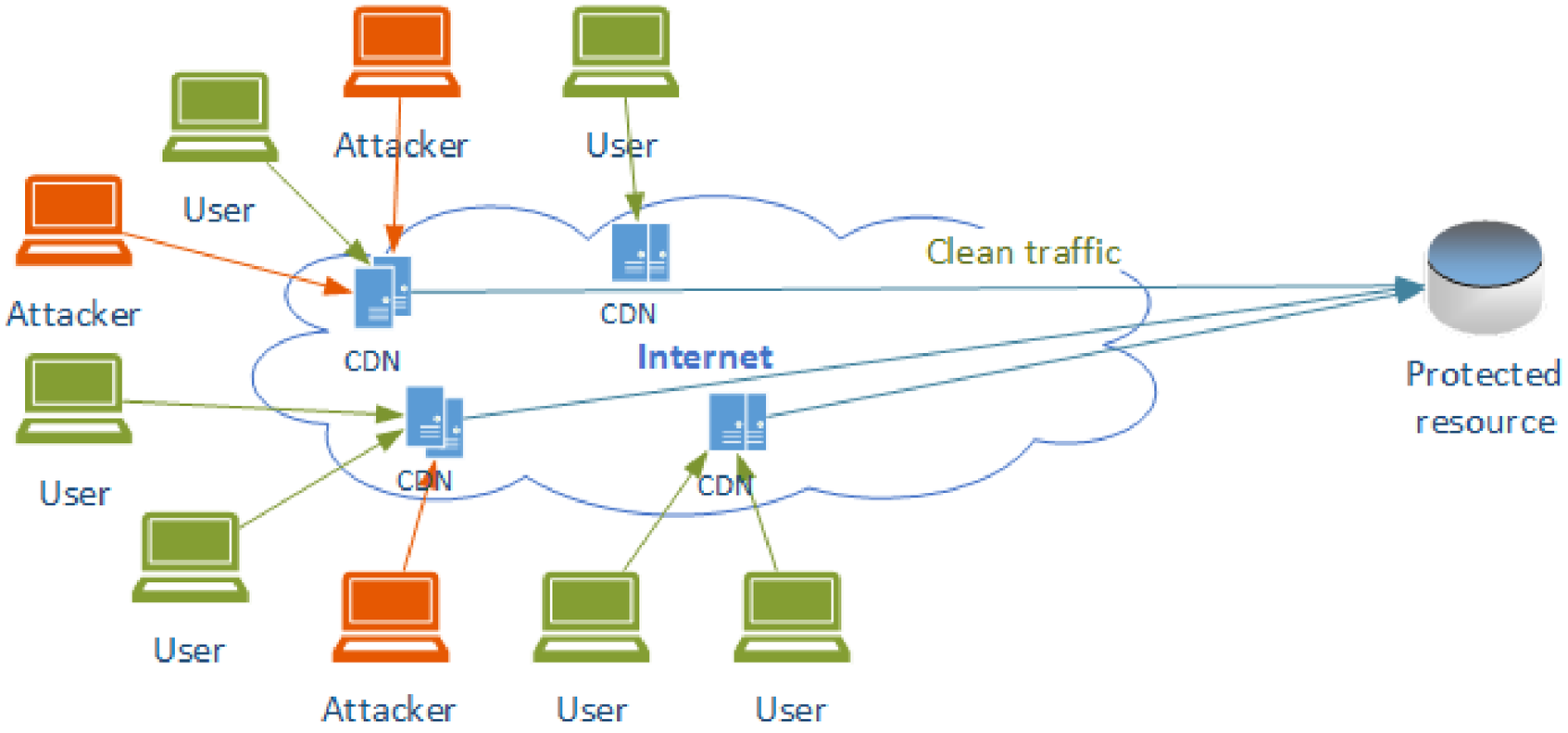
Always On Mitigation

- Permanently advertise 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 may depend on the size of the size more than 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?

Sockets Overview

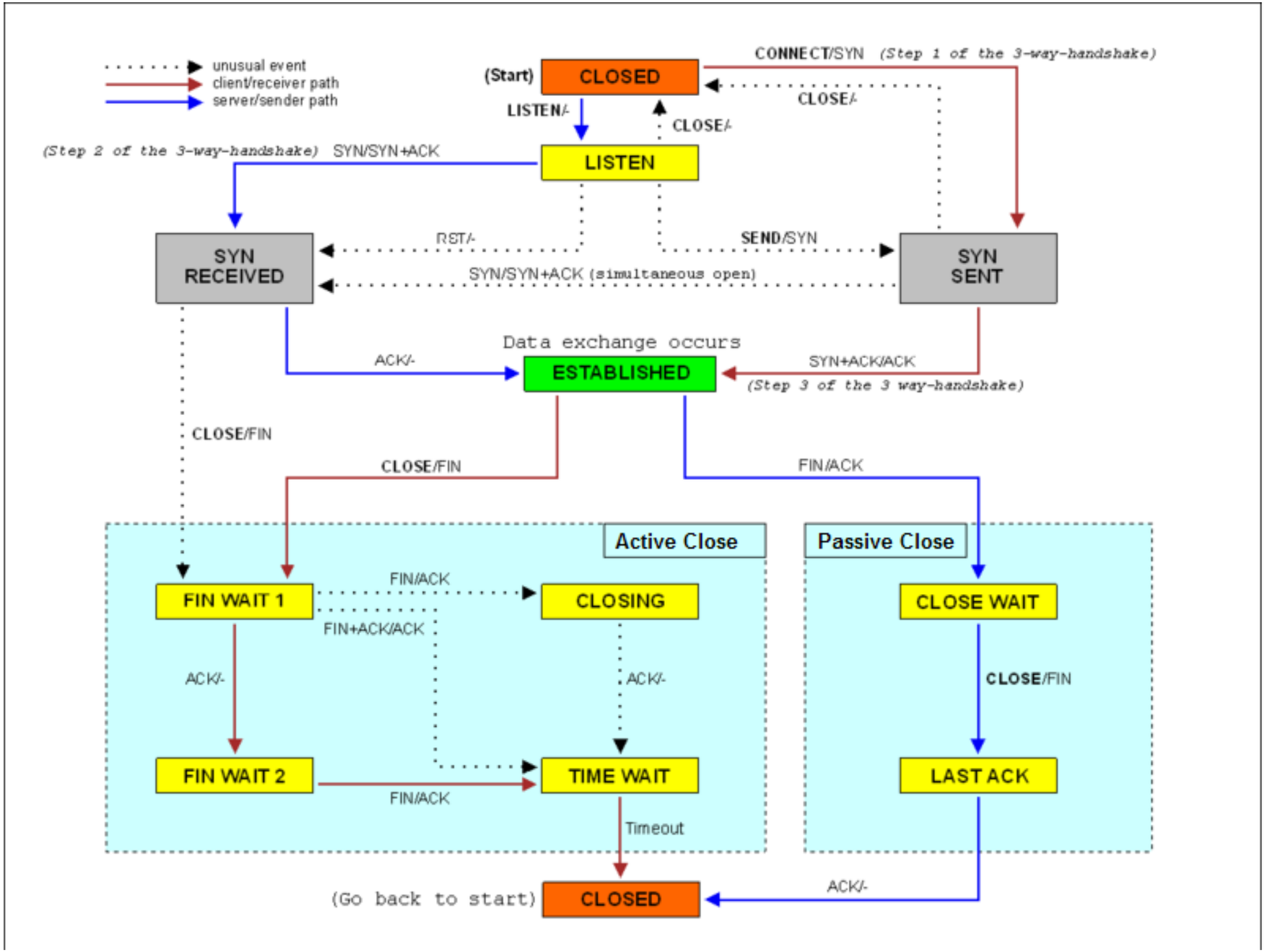
Sockets

- 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

Socket State Diagram

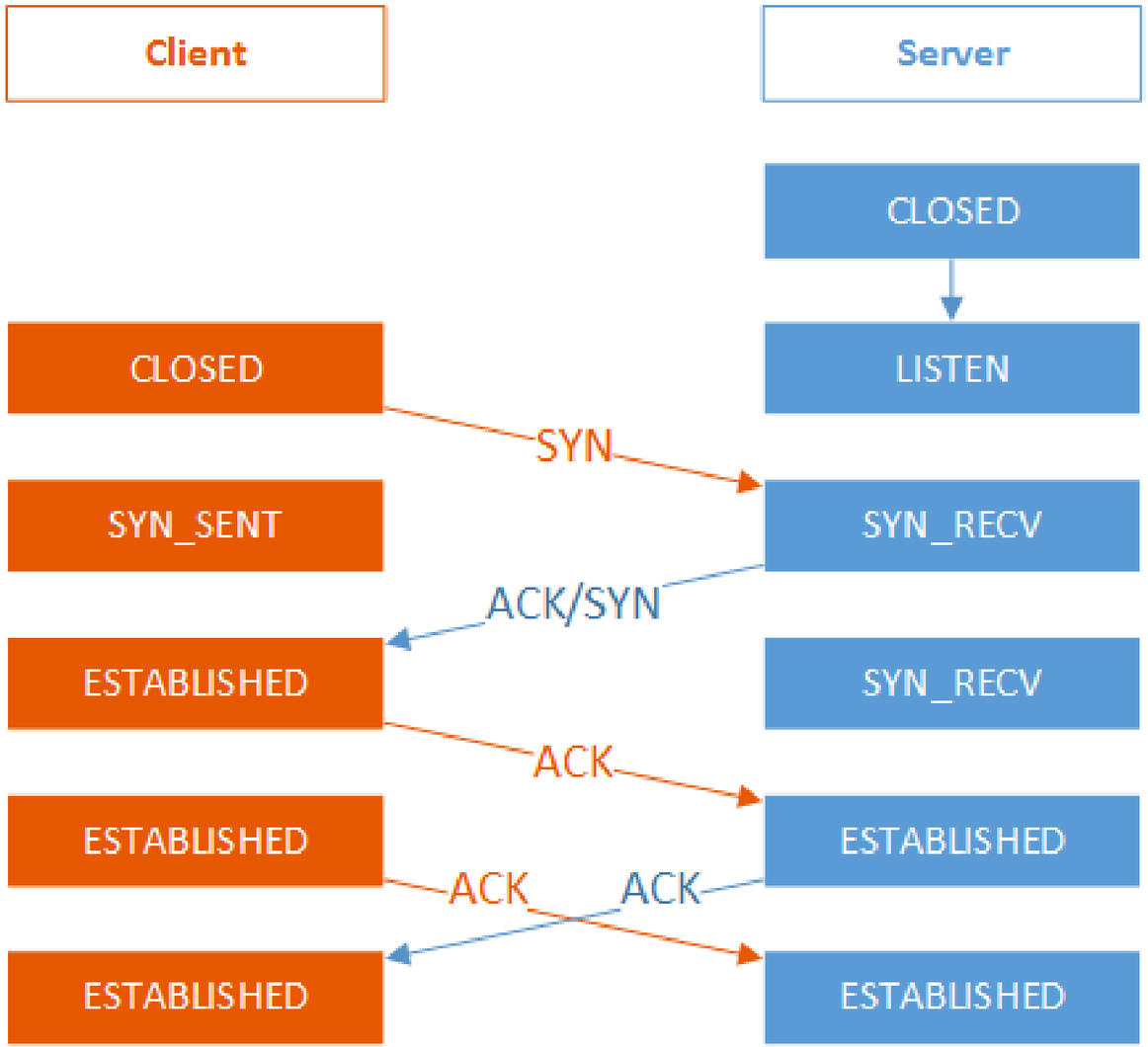


Source: Wikipedia

Opening a TCP connection

Let's review the sequence for opening a connection

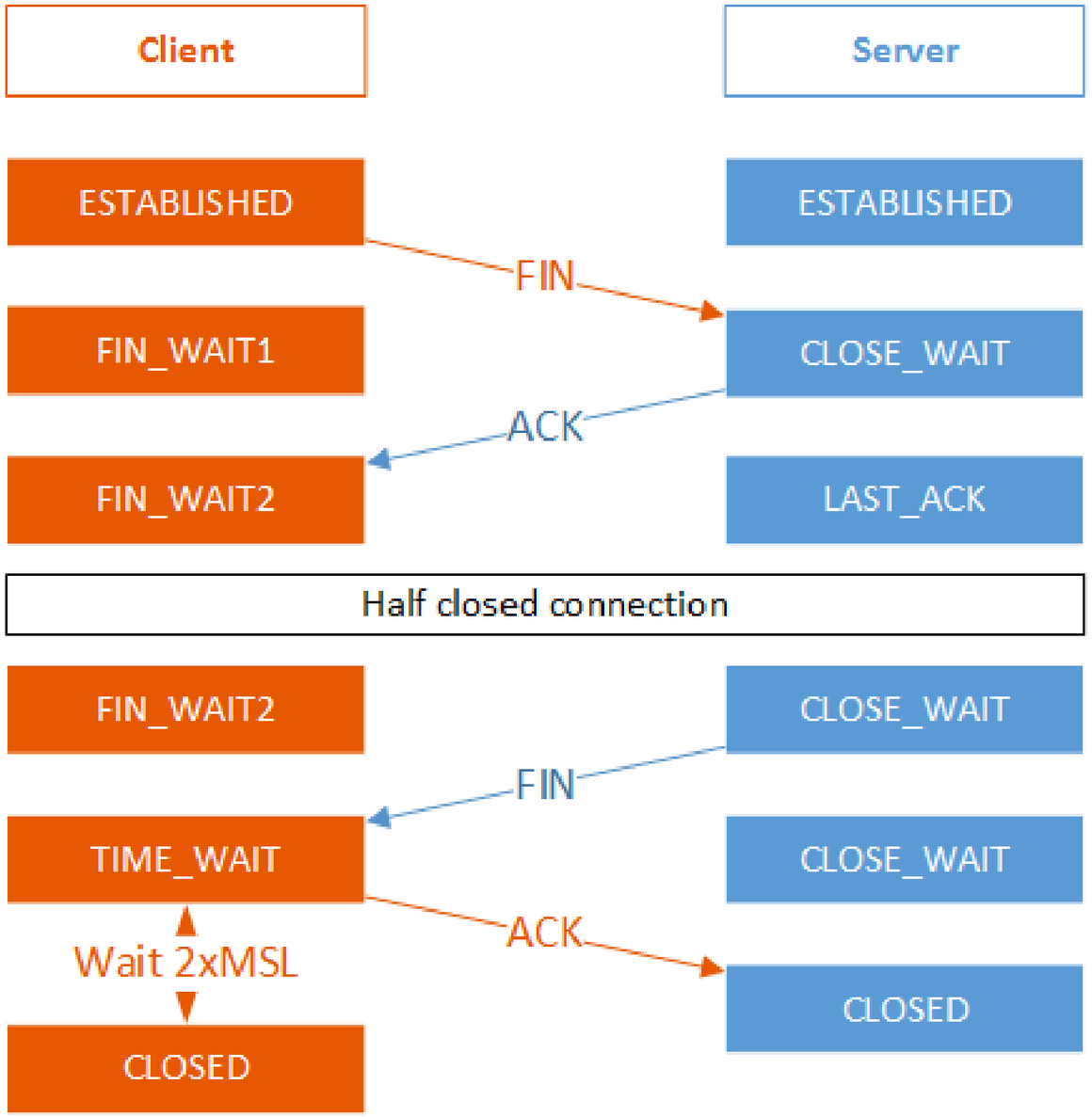
- Server side opens a port by changing to LISTEN state
- Client sends a SYN packet and changes state to SYN_SENT
- Server responds with SYN/ACK and changes state to SYN_RECV. For the client this is ESTABLISHED connection
- Client has to ACK and this completes the handshake for the server
- Packet exchange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

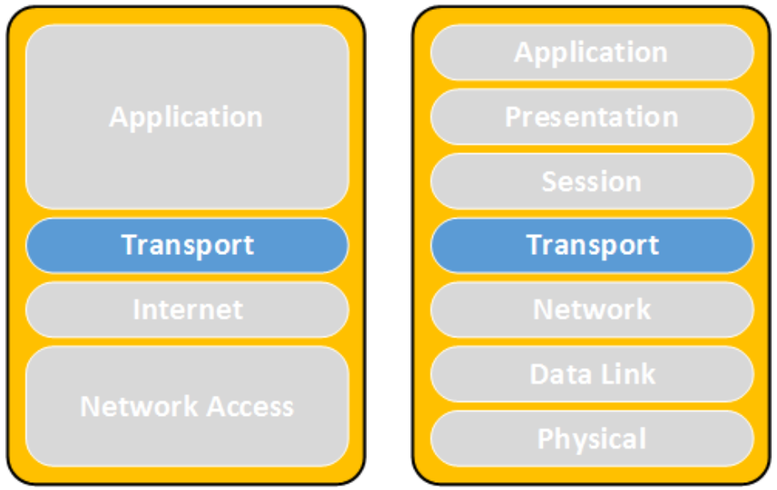


Use of netstat for troubleshooting

```
[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]#
```

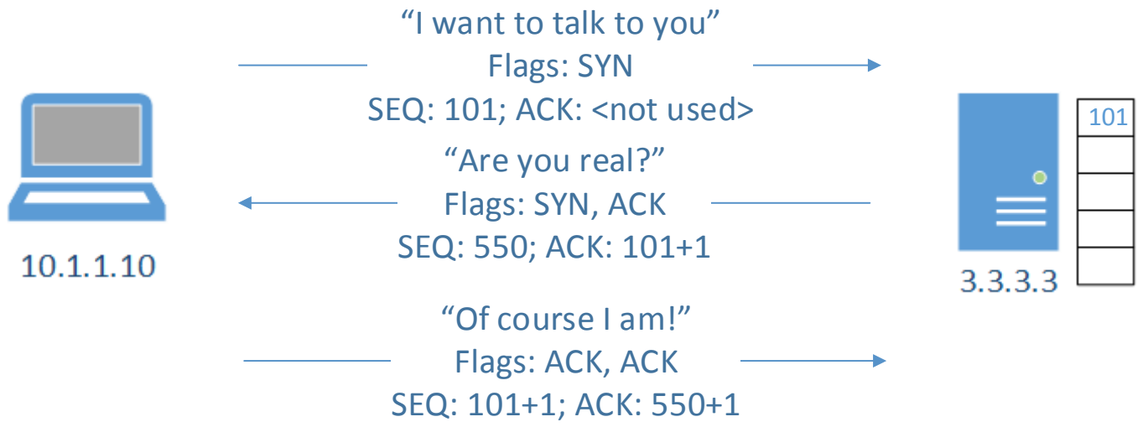
Attack types and terminology

SYN Flood


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

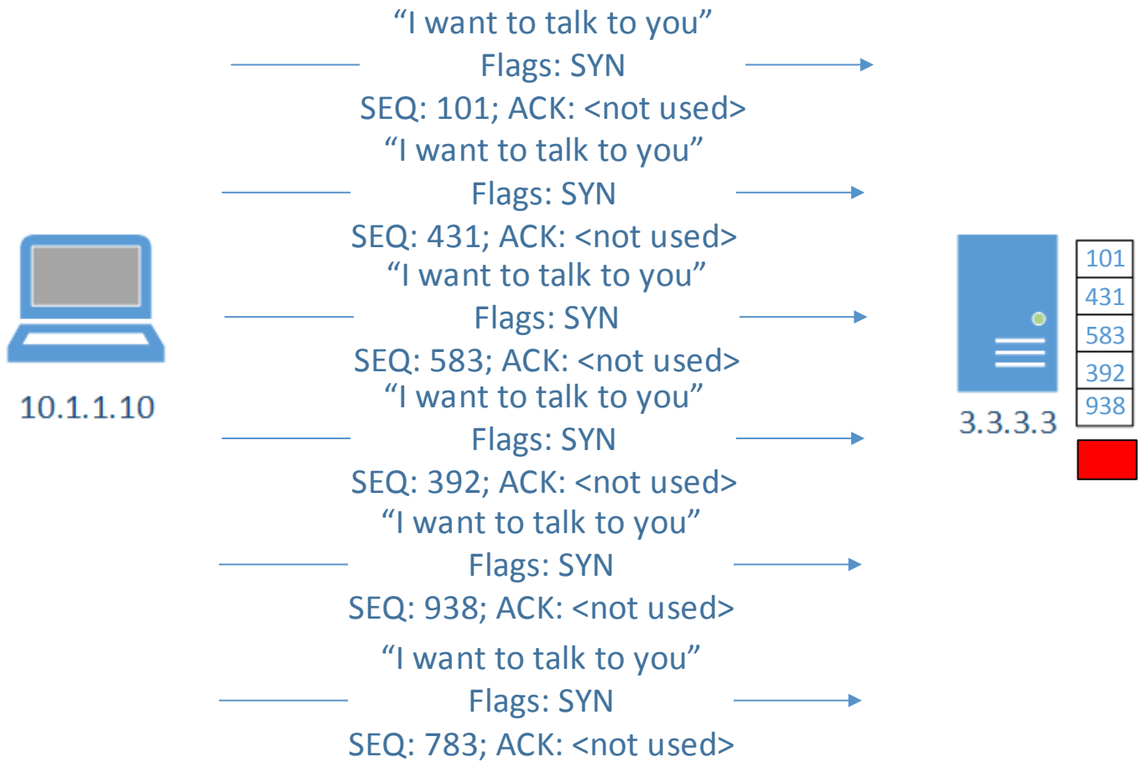
What is a SYN flood?

- What is a 3-way handshake?



SYN flood

- Exploits the limited slots for pending connections
- Overloads them

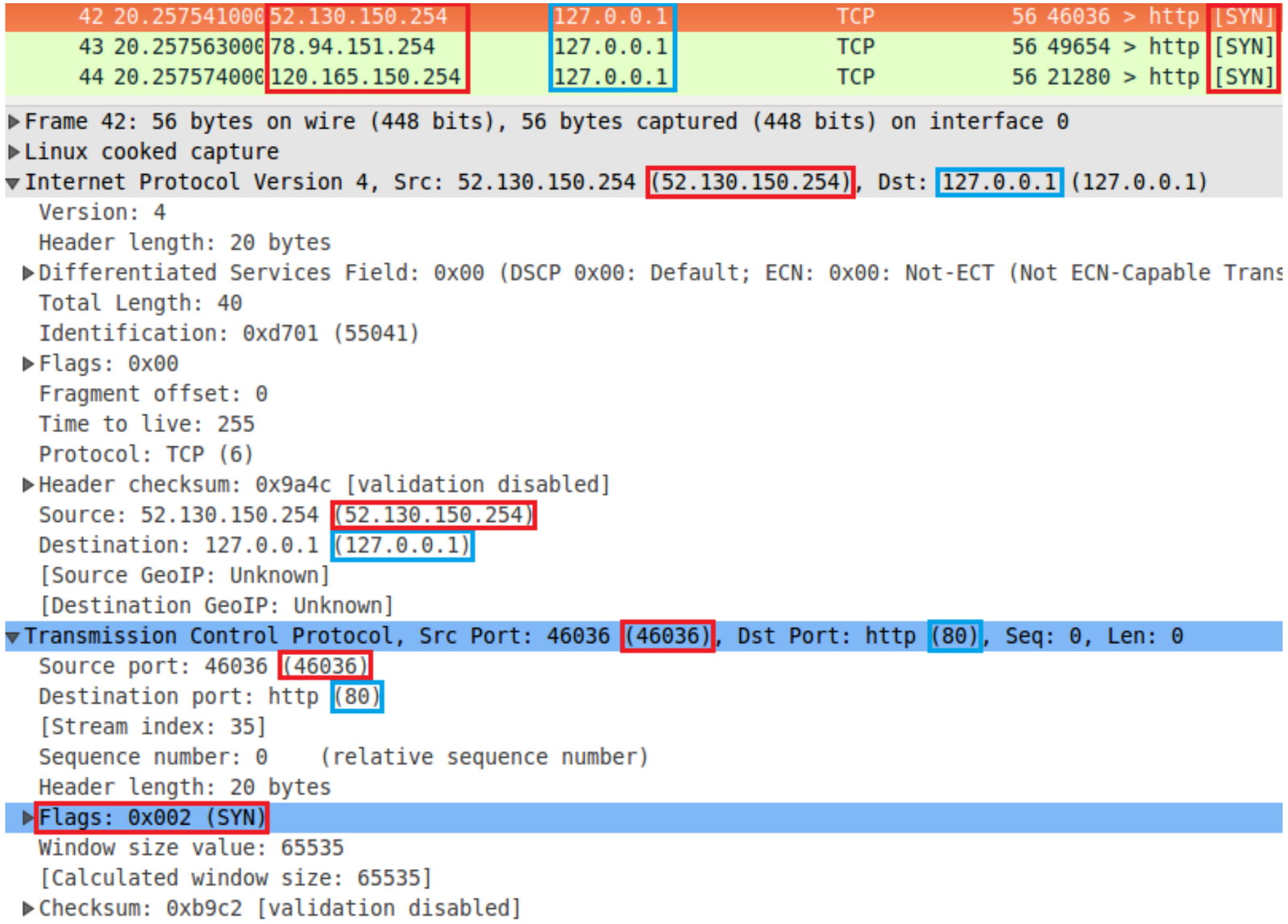


SYN flood through the eyes of netstat

- 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



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

SYN flood mitigation

- Technology
  - SYN Cookies
  - Whitelists

What is a SYN cookie?

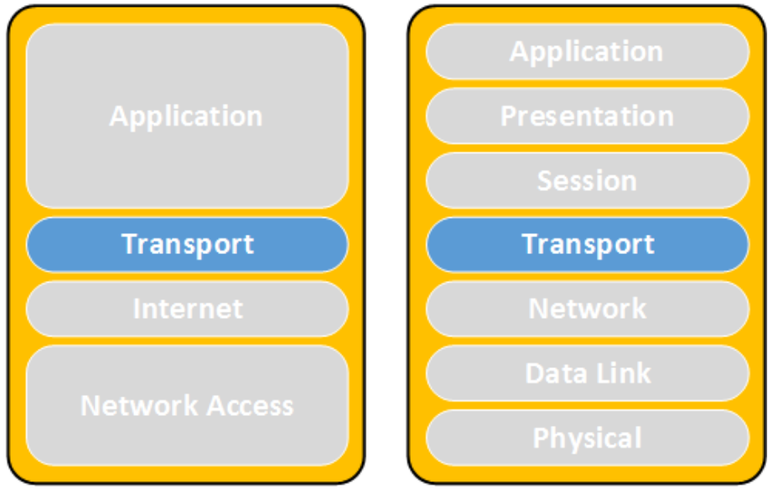
- Hiding information in ISN (initial sequence number)

- 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)

Slowloris


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

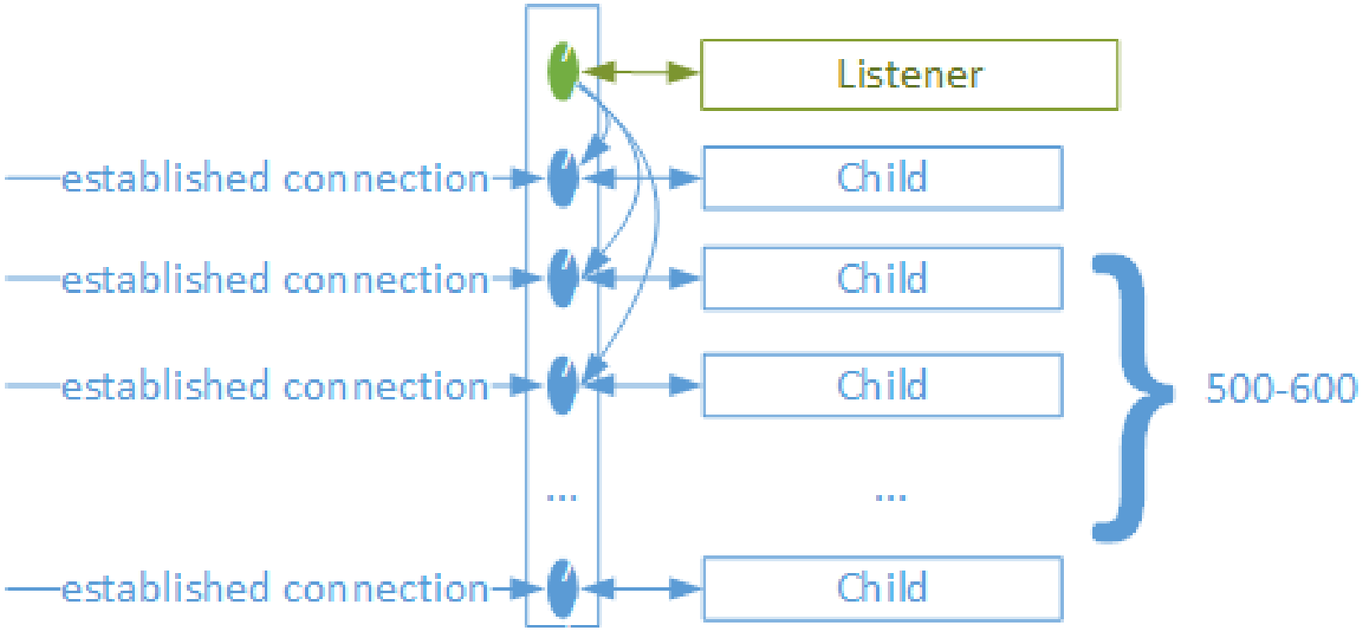
Connection handling architectures

- Process based connection handling?
  - Think “Apache”

- Event based connection handling?
  - Think “nginx”

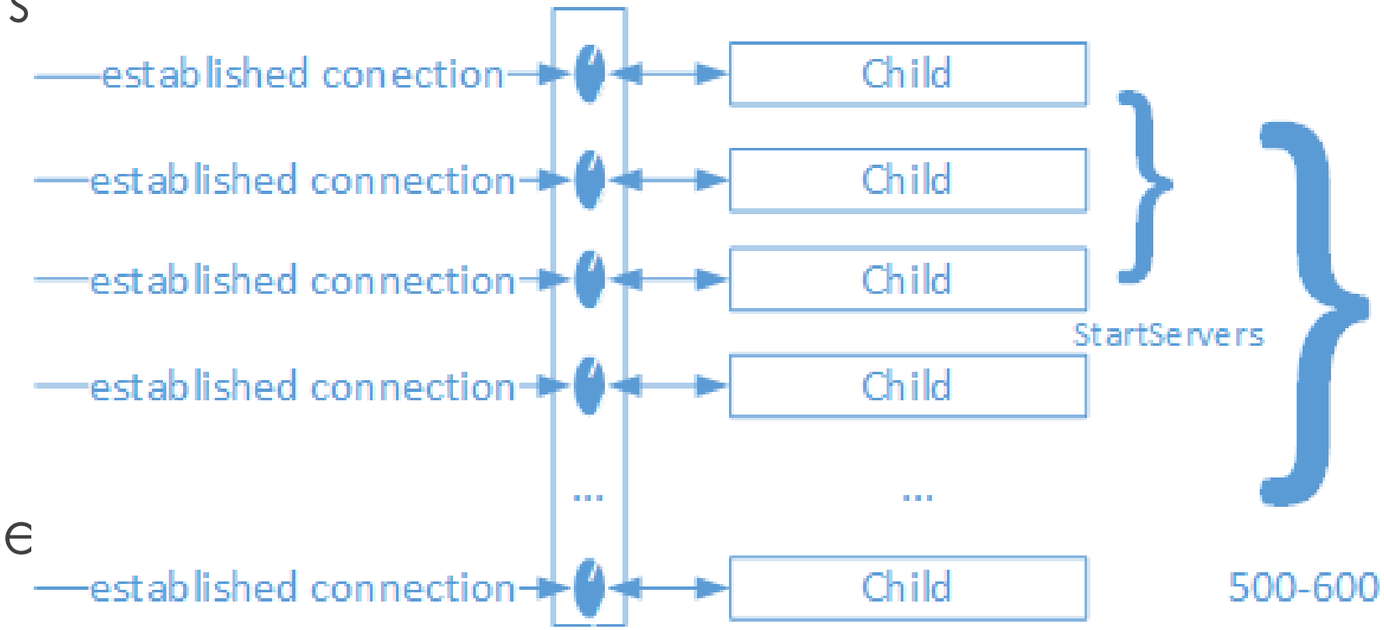
Process oriented explained

- 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 s
- 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 processe are spawned (forked)
- …and so on…
- Up to about 500-600
- The initial set is defined by StartServers

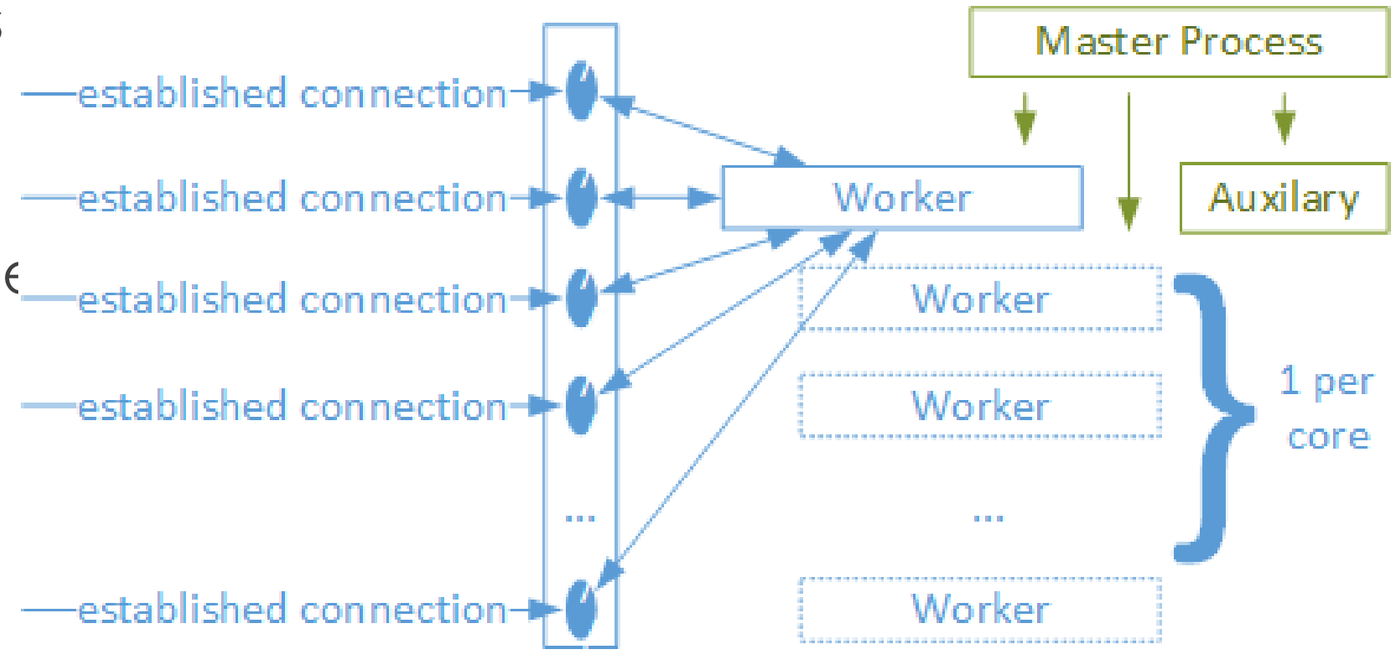


Nginx (simplified)

- Master Process controls logistics
- Support processes (cache management)
- Worker processes process conne
- 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

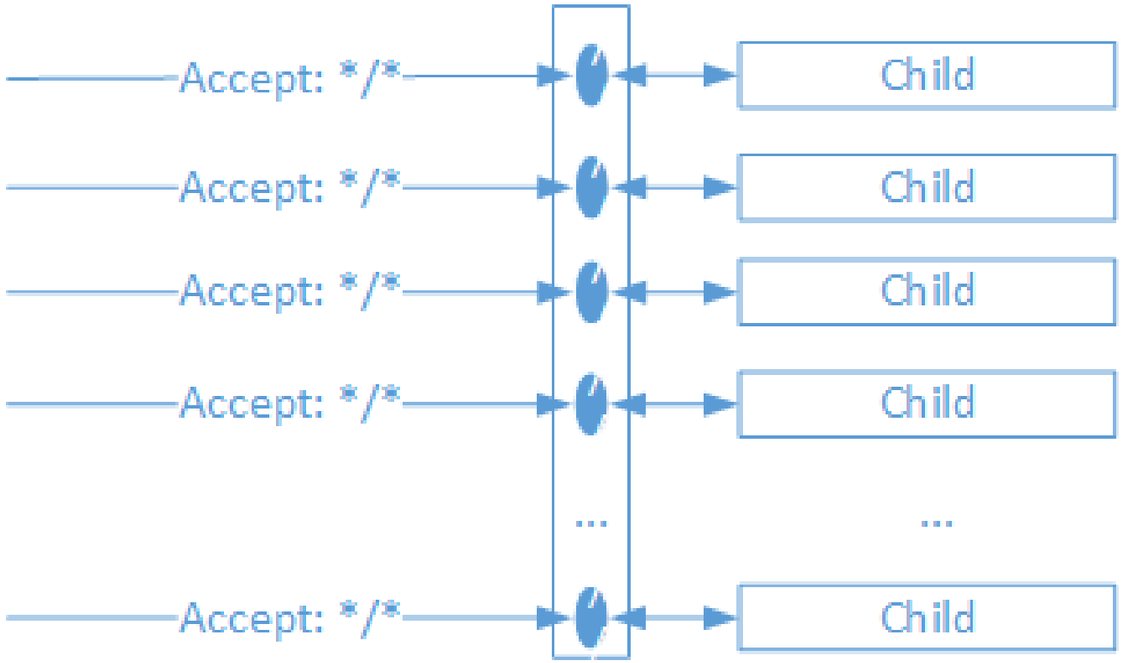
wait...

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

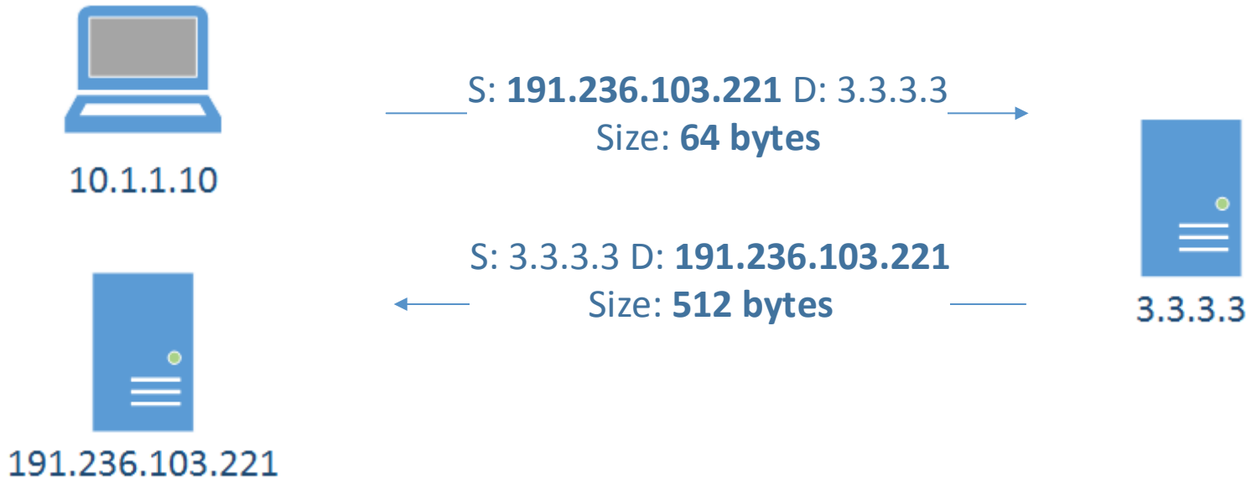
Reflective attacks

- 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

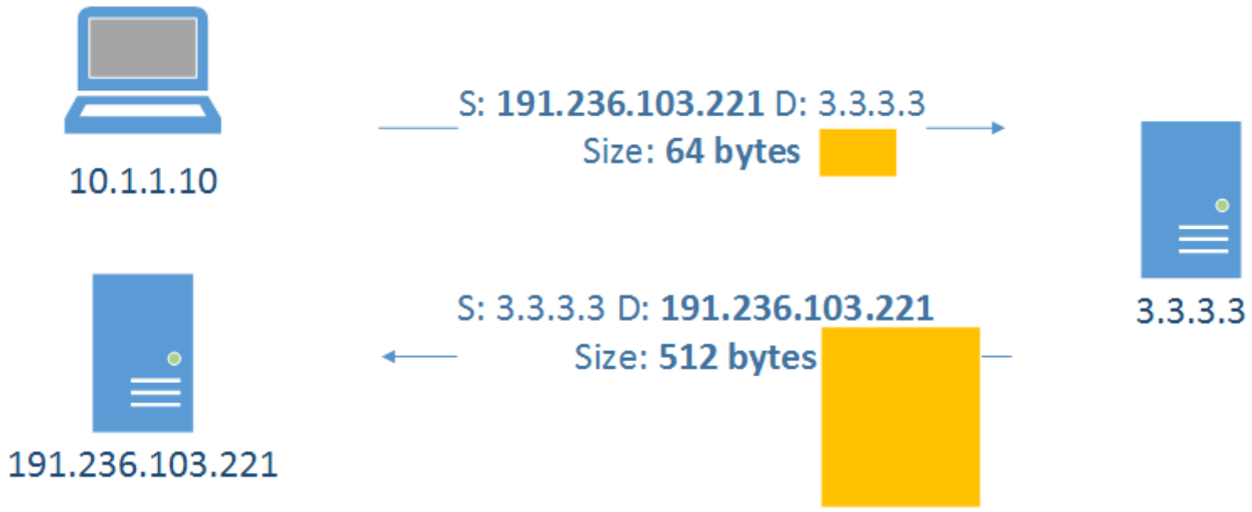
The ones that are of interest are:

- 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

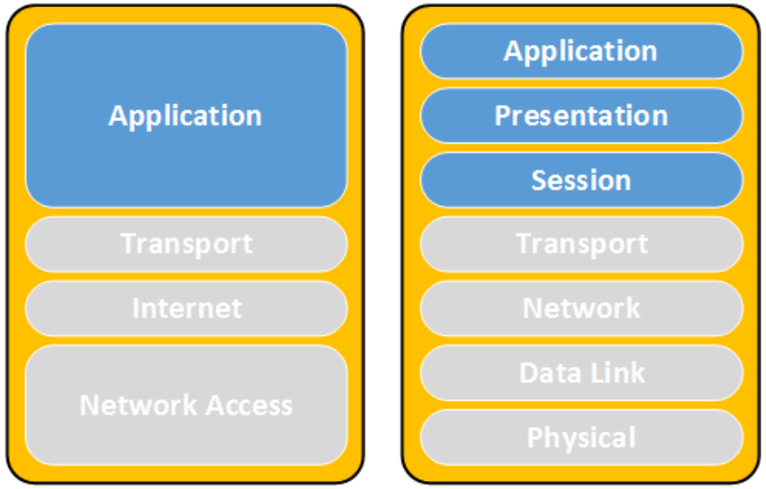
Amplification quotients

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

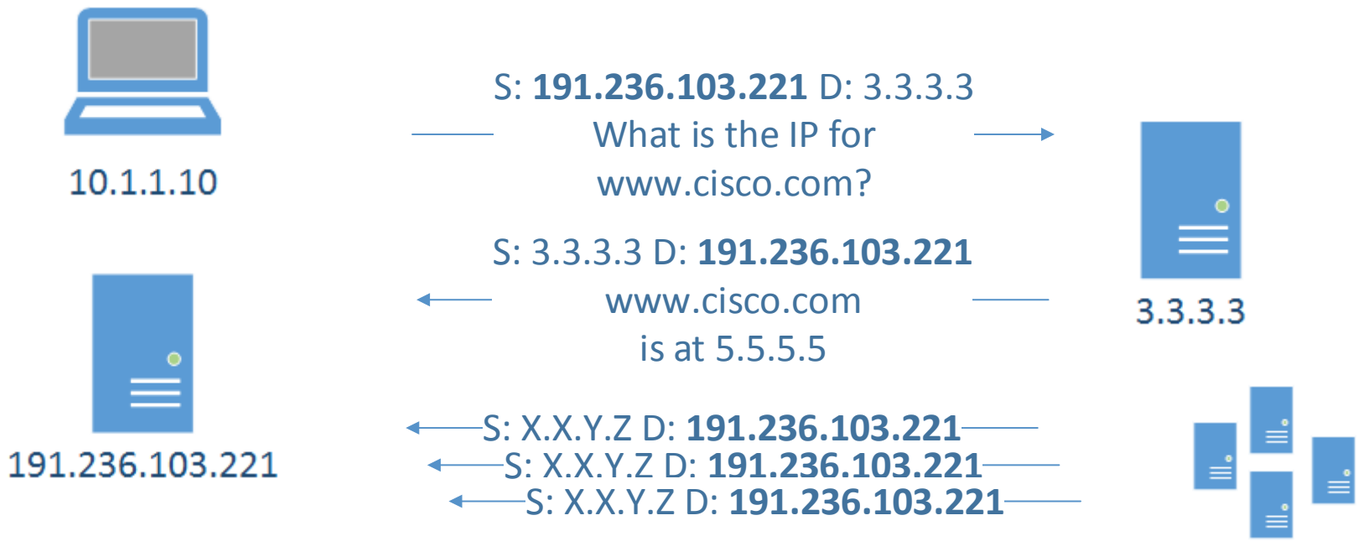
Questions?

DNS 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/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+hvYAUGHo7WwRmxGRaipS8l9+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/Qx5RiNbPCIw+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 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

Reflection and Amplification

10.1.1.10

191.236.103.221

S: **191.236.103.221** D: 3.3.3.3

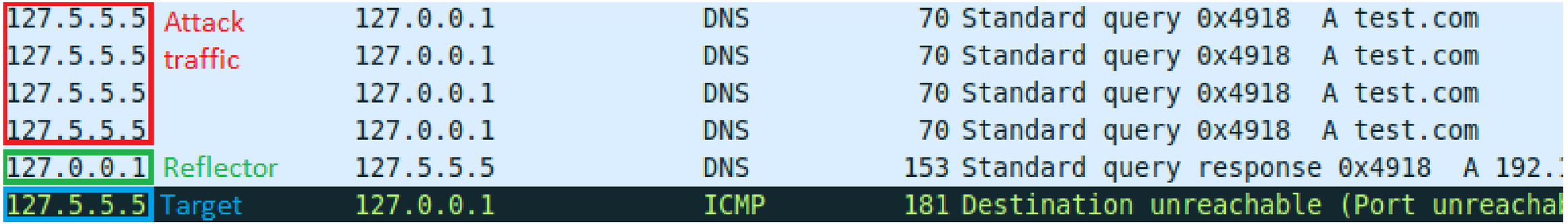
What is ANY isc.org

S: 3.3.3.3 D: **191.236.103.221**

3.3.3.3

```
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. KHMs09DaFMw416/7xXhaD9By0NrqCiQ4kBnqi6oq2VocZRREAbUHHAY
KydlgKO5vOavvd1Fy86/oiODkk3yyHspciwdJvjefu4PktdUnd1IQixW791q/
jWgHBL5iQiQlgBYv7Z5lfY1ENn+6fPOchAyvvNqEBYcdqVNBpzzOjtzxIU=
isc.org.            481     IN      DS      12892 5 2
F1E184C0E1D615D20EB3C223ACED3B03C773DD952D5F0EB5C777586DE18DA6B5
isc.org.            481     IN      DS      12892 5 1
98211D08B4C8A1D9F8AEE1E2237AEF69F3F9759
isc.org.            5725    IN      RRSIG   A 5 2 7200 20130620134150 20130521134150 50012
isc.org. iCBy1JjPP6mXVYpSc62JCrZWHnVYAUGHb7WxRmxGRoip38I9+LOvRl
2erglomkBP79m9ohnFOxNEAoueA6TIHCiGxOkgnk3hBfMfjLB9rhvichn
uxC02D8gc1DJDU5egfpJCF2ftFhEvNkeMvt6GGNvdcWWkBsFHCxM7Fms= 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+Konkdt8c4y3WlJHVs5TjanvhdEyHI4qPh/cHh
+y1vA6+gAwTHiX+GpscctNxEIvoSvWu3m9Nocnivu//AZGloL/SyDgEsIbJMY
X+ZXY5qrgGrV2grOcKAAA91Bus8beIhYQiZTsdaH2TS+AKjKINEgvm
yQi5xNEodzE3p0ygfPq4eMNO4FRT9LQDhTRD3v8zfxFNXKvBsQWZGBH0
5+QicbC6xnGyn1bBptJEEGnCBG01ncJ+IMCyEf96VGHKJfeowORIrDQ3
cjJRFPTCOkA8n4jBvnsimUP/TGI+Mg4ufAZpE96jJhvFBsdcC/iOo6i XicGVIA==
isc.org.            5725    IN      RRSIG   DNSKEY 5 2 7200 20130620130130 20130521130130
50012 isc.org. o18F3KIRkYedFRwIe5MP4qDo3vi8g0XK95WCYDT5oGhs9R5eyc/6KEW
Se4IZXRhf6d77xXierMYOrafh/GHdJPRoE1xL/nzH/hTBJA9XDbC5I/
EUpFIGVLVdGy43XKtyvm0J2nycSMdGa2VeLKo+hHTmH3S+3pGRVJp2IK5Z0=
isc.org.            5725    IN      DNSKEY  257 3 5
BEAAAAOhHQDBrhGb+phgq2wGUpEQ5t4DtUHkoMMFu2hWLDMvoOMRXjGr
hhCeFvAZh7yJHfBZGRMN6hd38hXG/xyIYCO6KrpbdoJwx6YMKLA5/kA+
u50WL8ZR1R6KTbzYMf/Qx5RNbPCIwtvT+L8eXEJmO20j81ULgqy3 47cBB1zMvnnz/
4LuJpA0do9QbKjBA2547515aNIMctwsB8/2+2E63/zZrGzBkDBrN/
9BexjpkaBjRhZo+EaXn3dTy47R09Uix5WcJ++xaqZ7+yayL
KOOed839Z7SDmsn2eA0FKKqpwA6LXeG2wtjxmvu9oA8IVUgEf/reCr/bB yBNsO70aEFTd
isc.org.            5725    IN      DNSKEY  256 3 5 BGEAAAABwuHt9Cem0BJ0JGTO7C/
a8MtR6hMbuFja1dfG/inaJpYv7vH XTrAOm/MeKp+/x6eT4GLru0KaZkvZJnqTI8JyaRTv2OMV
HBfh/hL2ImCft2O7n3MfeqY+tjPnY7dWghYW4aVfH7VVEGm958a9nfi79532Qekkxh
x8pXWdeAaRU=
isc.org.            5725    IN      DNSKEY  256 3 5 BGEAAAABwuHt9Cem0BJ0JGTO7C/
a8MtR6hMbuFja1dfG/inaJpYv7vH XTrAOm/MeKp+/x6eT4GLru0KaZkvZJnqTI8JyaRTv2OMV
HBfh/hL2ImCft2O7n3MfeqY+tjPnY7dWghYW4aVfH7VVEGm958a9nfi79532Qekkxh
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
g.root-servers.net.  297708  IN      A       192.112.36.4
h.root-servers.net.  298308  IN      A       128.63.2.53
h.root-servers.net.  416776  IN      AAAA    2001:500:1::803f:235
i.root-servers.net.  297617  IN      A       192.36.148.17
```

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)

```
35820 128.1479010€ 127.5.5.5    127.0.0.1    DNS     70 Standard query 0x4918  A test.com
35821 128.1479080€ 127.5.5.5    127.0.0.1    DNS     70 Standard query 0x4918  A test.com
35822 128.1479150€ 127.5.5.5    127.0.0.1    DNS     70 Standard query 0x4918  A test.com
35823 128.1479410€ 127.0.0.1    127.5.5.5    DNS    153 Standard query response 0x4918  A 192.1
35824 128.1479440€ 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)

```
35820 128.1479010( 127.5.5.5      127.0.0.1      DNS      70 Standard query 0x4918  A test.com
35821 128.1479080( 127.5.5.5      127.0.0.1      DNS      70 Standard query 0x4918  A test.com
35822 128.1479150( 127.5.5.5      127.0.0.1      DNS      70 Standard query 0x4918  A test.com
35823 128.1479410( 127.0.0.1      127.5.5.5      DNS     153 Standard query response 0x4918  A 192.
35824 128.1479440( 127.5.5.5      127.0.0.1      ICMP    181 Destination unreachable (Port unreacha
```

```
▶ User Datagram Protocol, Src Port: domain (53), Dst Port: 24058 (24058)
▼ Domain Name System (response)
    [Request In: 34402]
    [Time: 0.017424000 seconds]
    Transaction ID: 0x4918
  ▶ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 2
  ▼ Queries
    ▼ test.com: type A, class IN
        Name: test.com
        Type: A (Host address)
        Class: IN (0x0001)
  ▼ Answers
    ▶ test.com: type A, class IN, addr 192.168.1.1
  ▼ Authoritative nameservers
    ▶ test.com: type NS, class IN, ns localhost
  ▼ Additional records
    ▶ localhost: type A, class IN, addr 127.0.0.1
    ▶ localhost: type AAAA, class IN, addr ::1
```

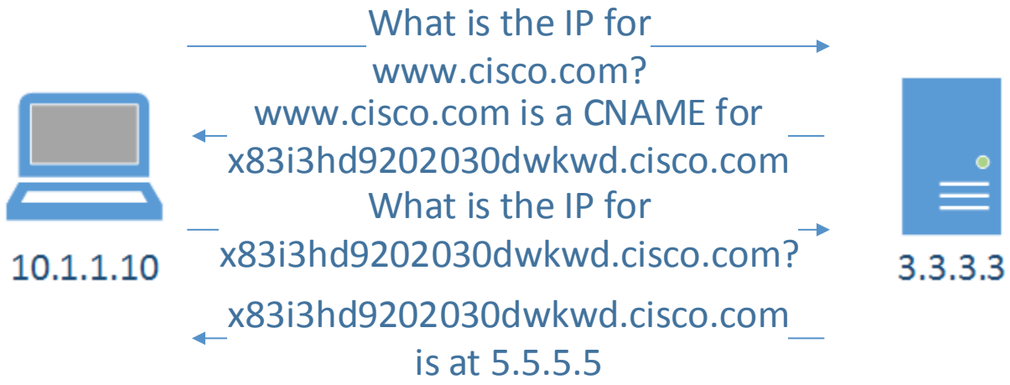
- Reflector (127.0.0.1) responds to the query to the victim (127.5.5.5)
- Note the number of records in the answer

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/

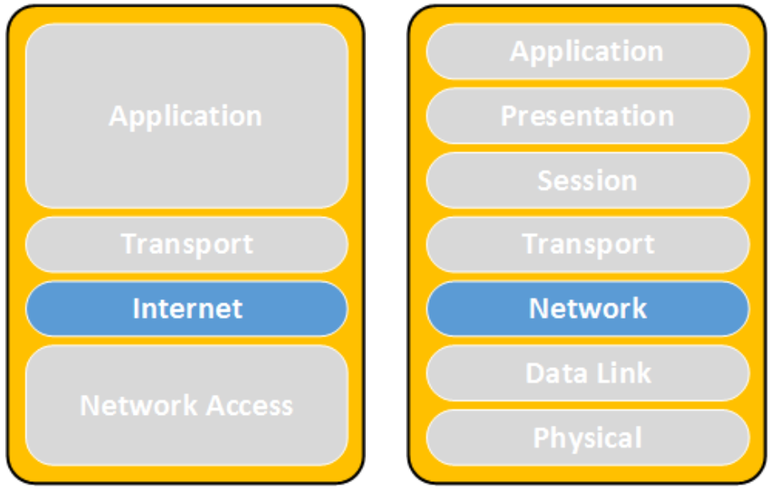
DNS attacks mitigation (victim - DNS challenge)

- What is a DNS challenge?



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

Backscatter



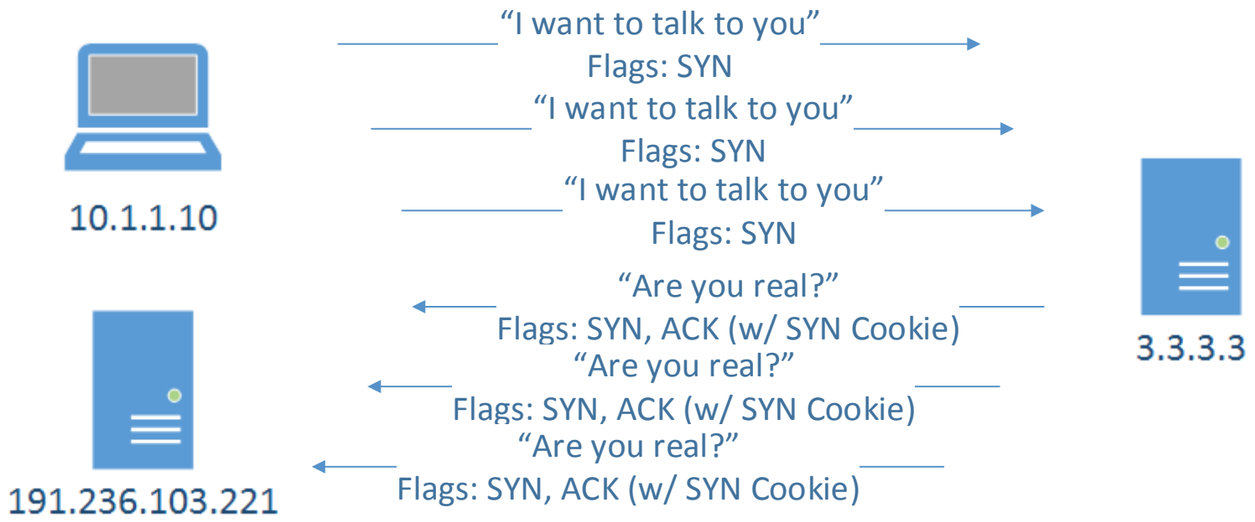
Backscatter

- 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 X
    - What is the traffic to X going to look like?

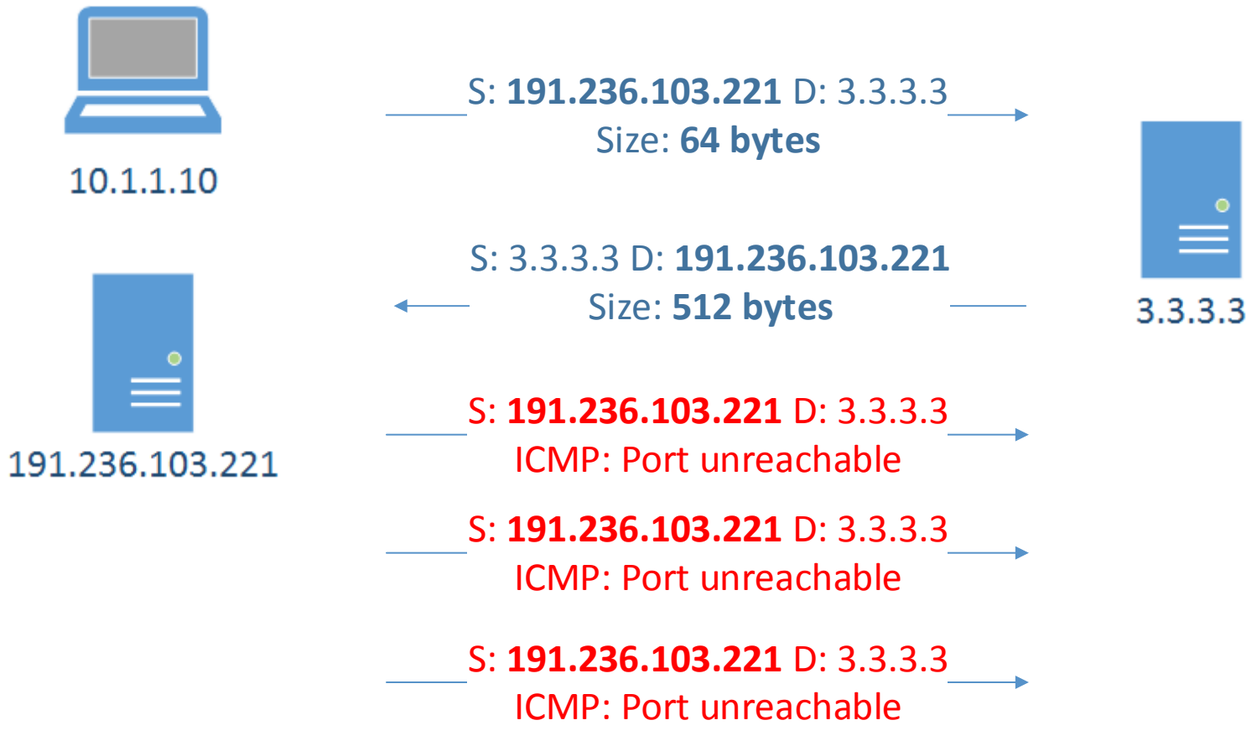
SYN Flood Backscatter?

- Cookie flood ☺



Are you a reflector? (Backscatter)

- In some cases return traffic/backscatter



Back scatter on the wire

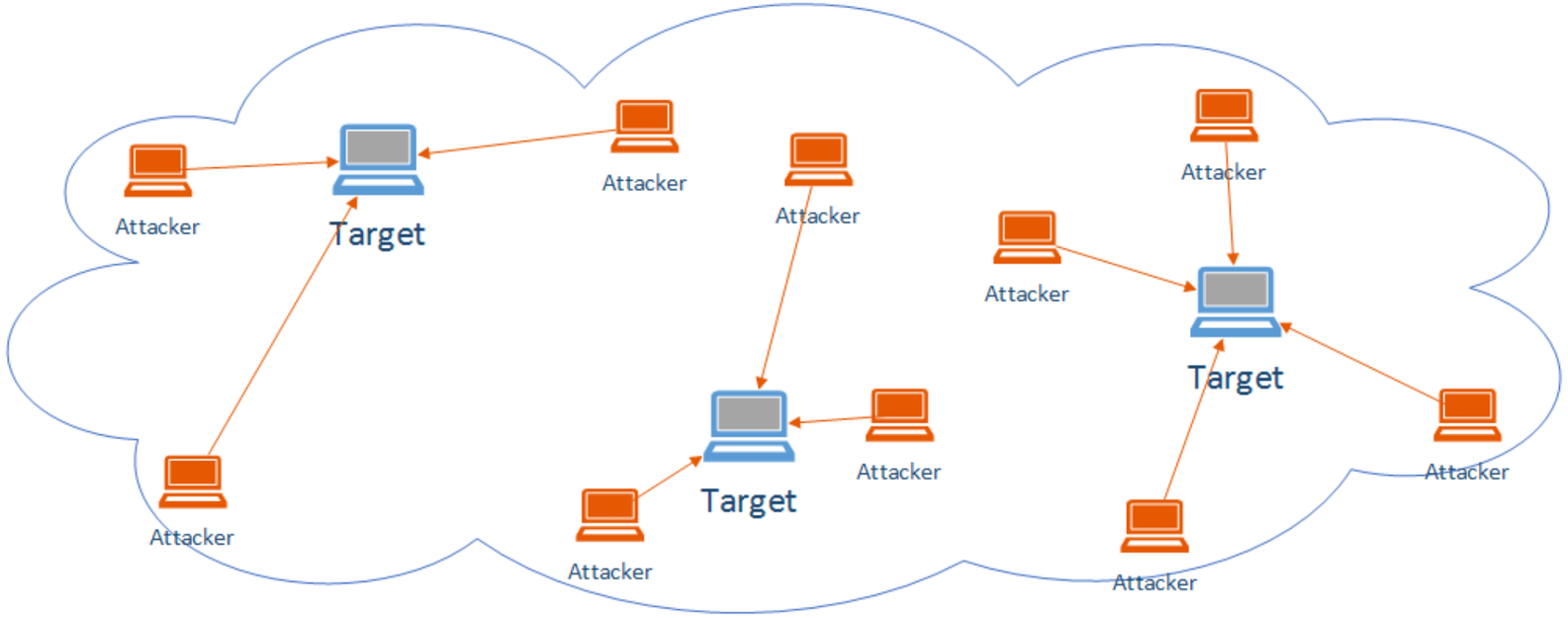
```
  20021 1.756892000 127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
  20022 1.756900000 127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
  20023 1.756907000 127.5.5.5     127.0.0.1     DNS      70 Standard query 0x4cb1  A test.com
  20024 1.756915000 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]
```

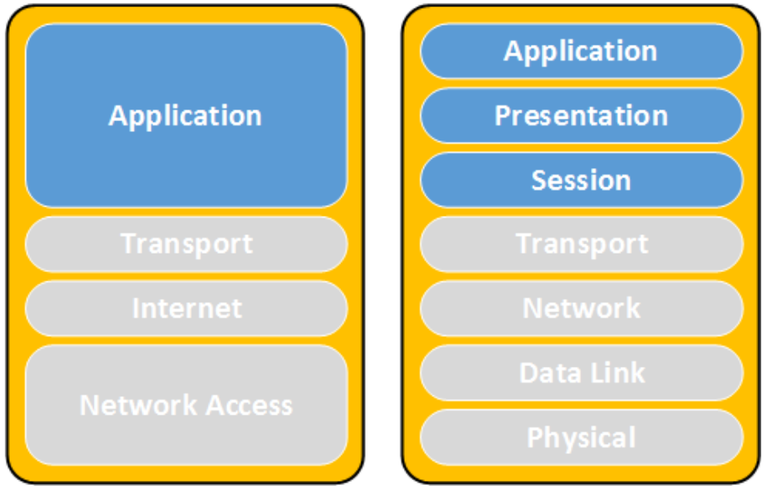
- The victim (127.5.5.5) sends and ICMP port unreachable to the reflector (127.0.0.1)

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



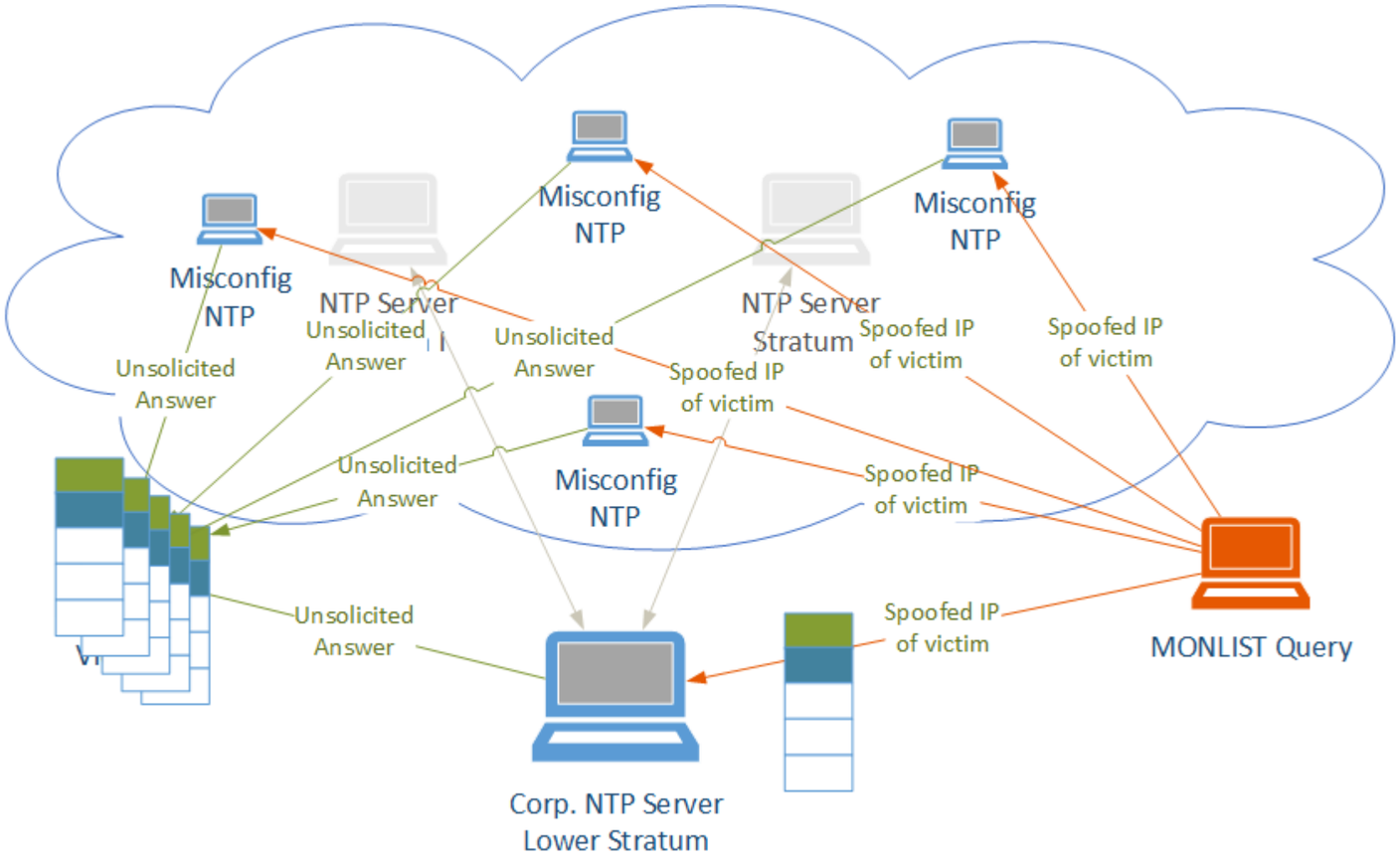
Network Time Protocol (NTP)



NTP reflection attack

- Stratum servers
- NTP queries

- MONLIST command
  - provides
    a list of clients that have
    time readings

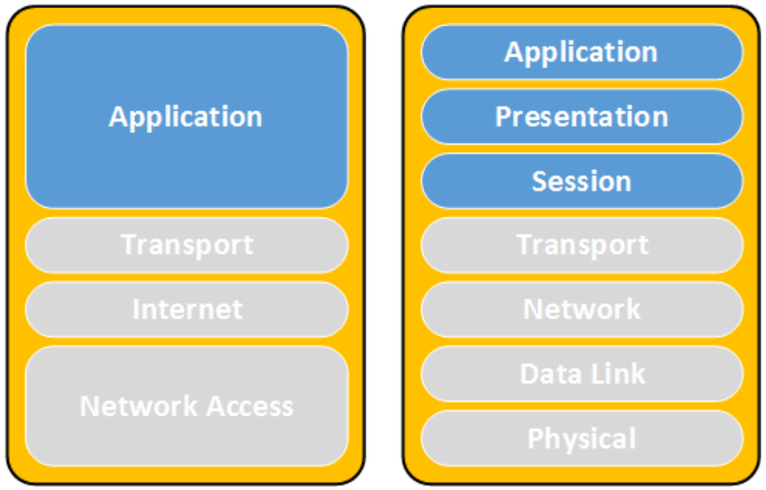


NTP server configuration

- 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/

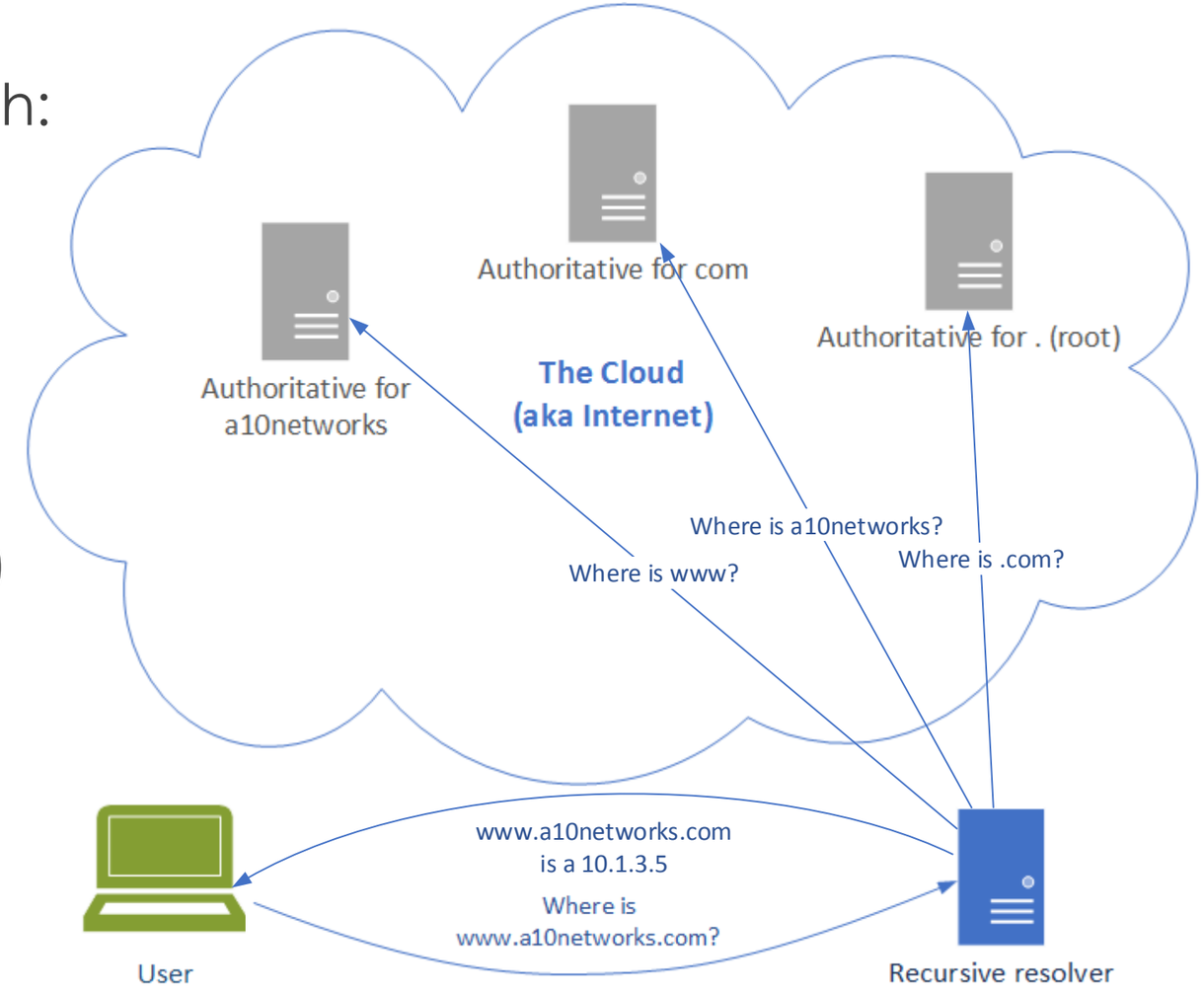
Questions?

Cache busting (back to DNS)



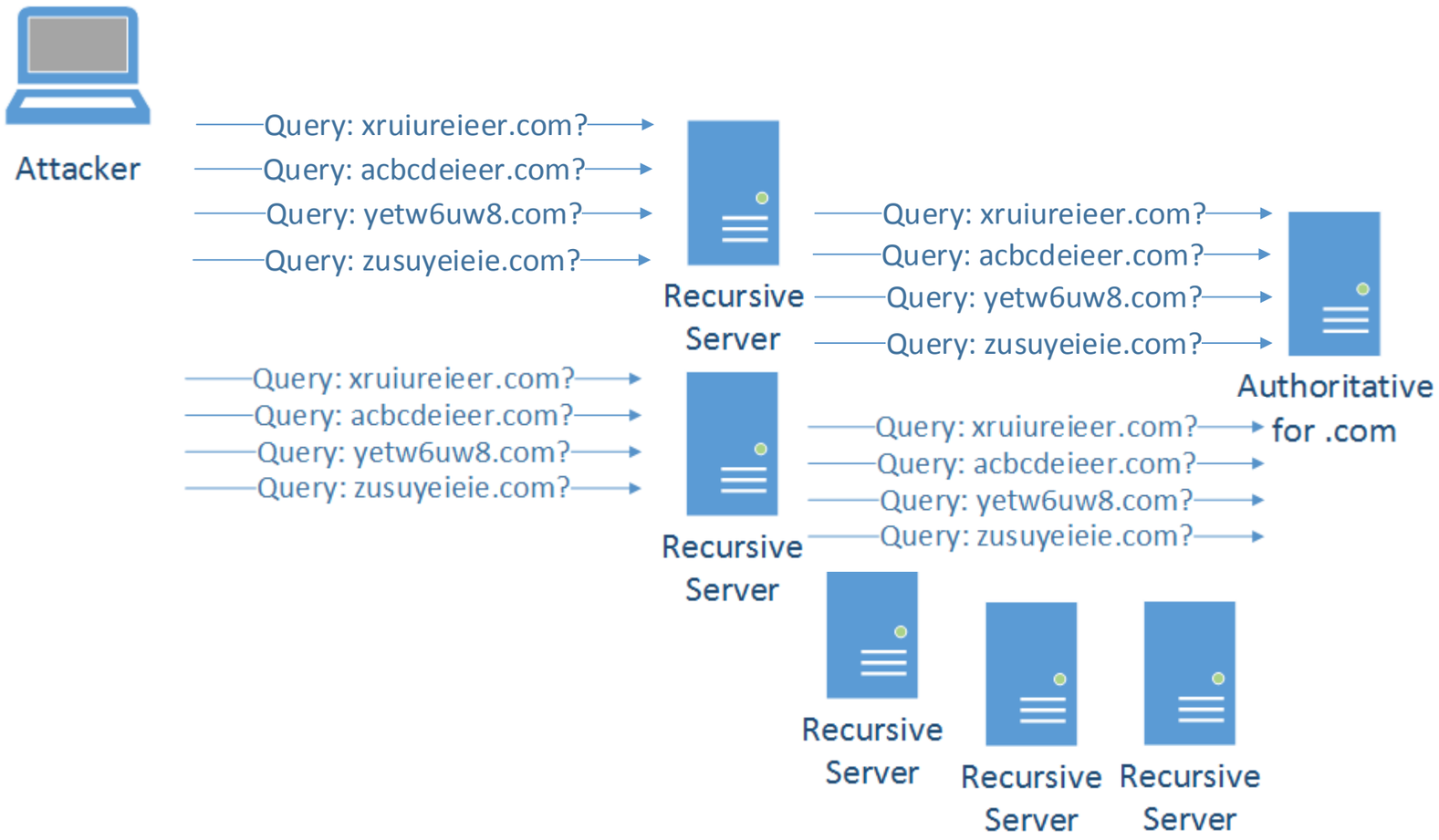
DNS resolution (rehash)

- 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?

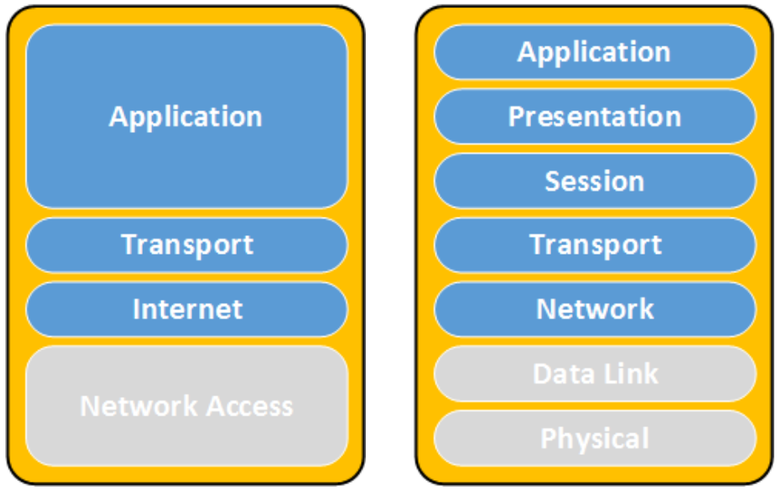
- Attacker sends a query to recursive/reflector
- Recursive forwards the query
- And so on…
- Imagine one more recursive resolver
- Rinse and repeat…



Questions?

Questions?

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