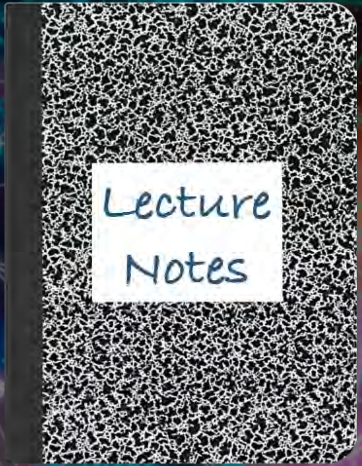
CS 419: Computer Security

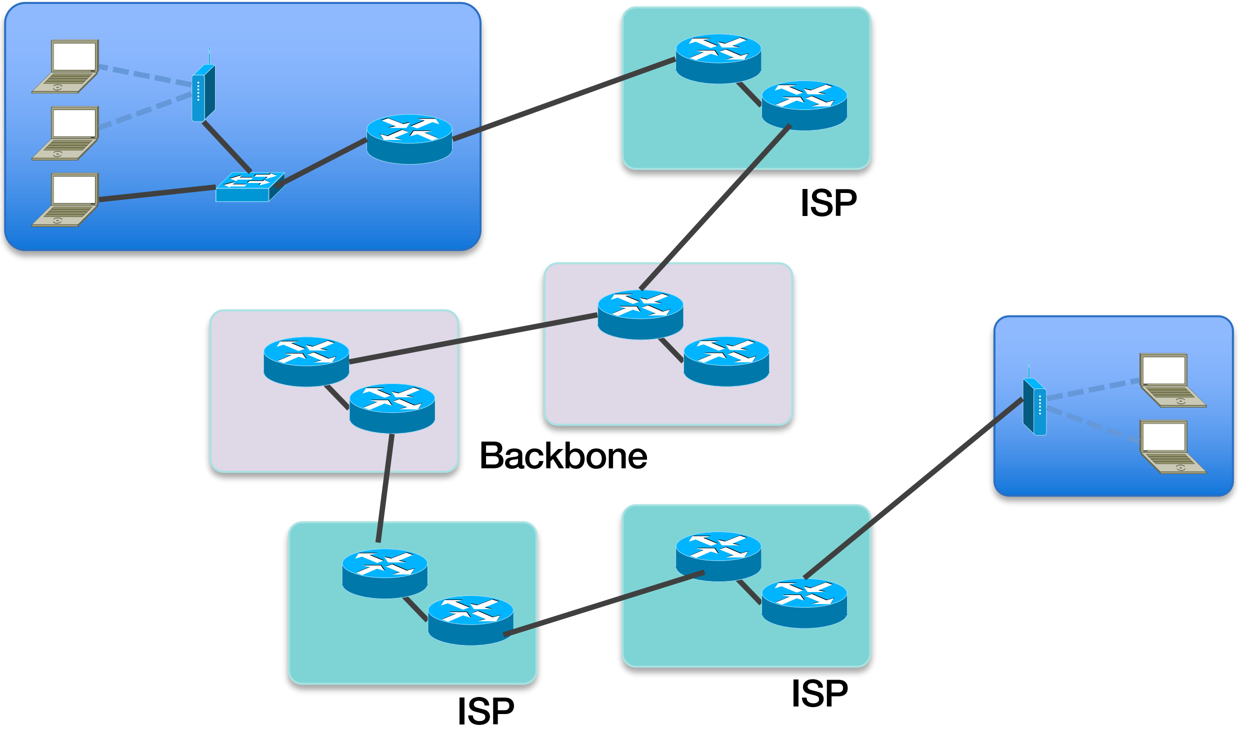
Week 11: Network Security



Paul Krzyzanowski

The Internet

Packet switching: store-and-forward routing across multiple physical networks
... across multiple organizations



The Internet: Key Design Principles

1. **Support interconnection of networks**
   - No changes needed to the underlying physical network
   - IP is a *logical network*

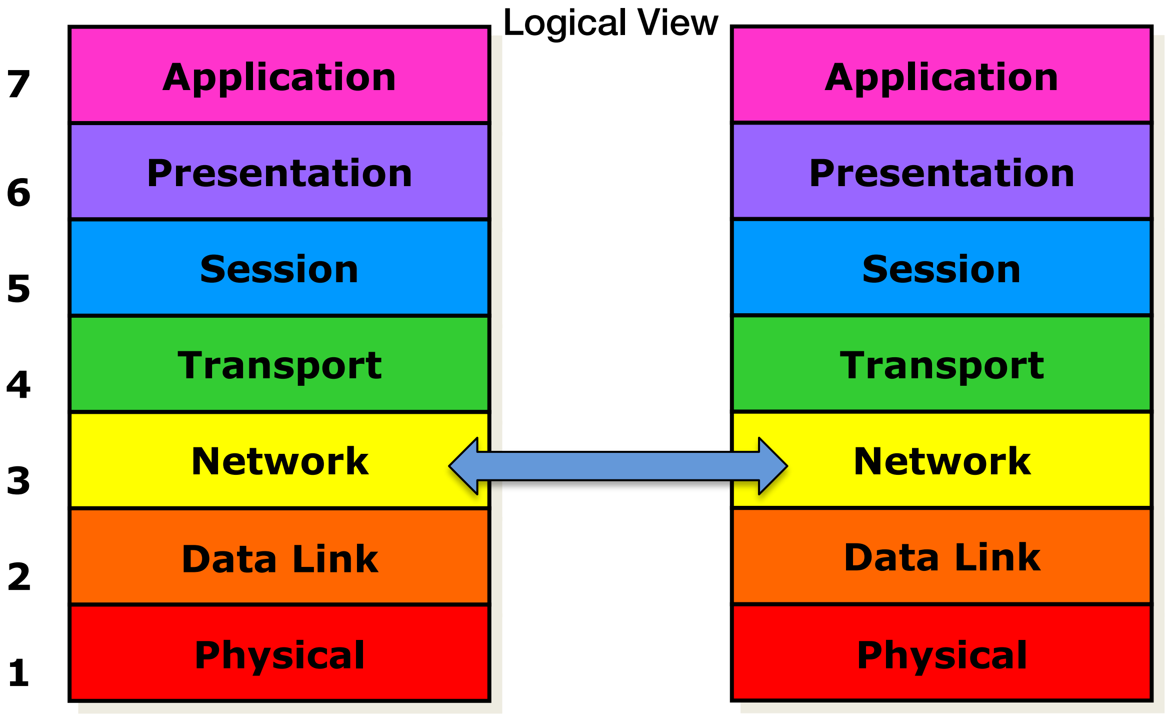
2. **Assume unreliable communication**
   - If a packet does not get to the destination, software on the receiver will have to detect it and the sender will have to retransmit it

3. **Routers connect networks**
   - Store & forward delivery

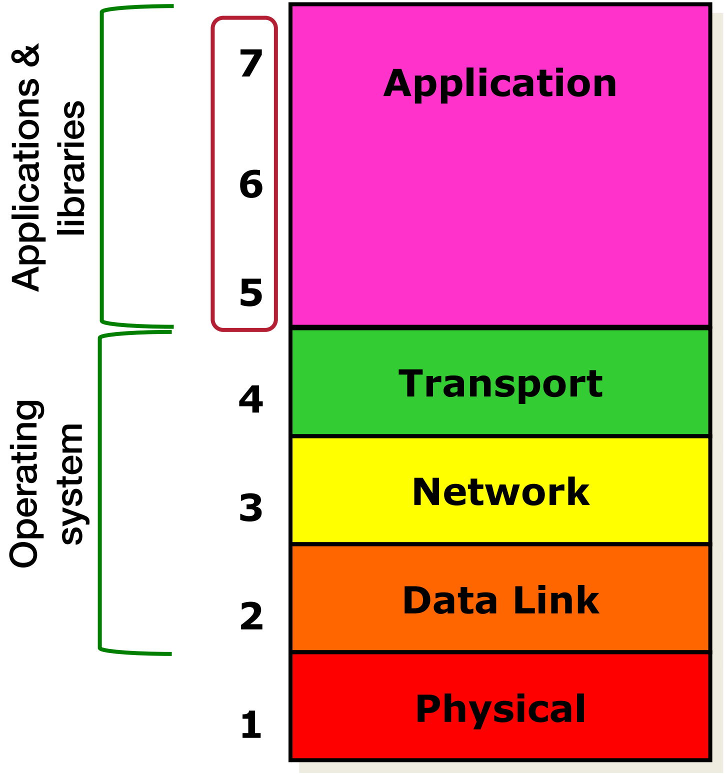
4. **No global (centralized) control of the network**

Network protocol layers

Networks are modular. Protocol layers communicate with their counterparts.
Low-level attacks can affect higher levels.



IP Protocol Stack



Internet protocol stack

Protocols

SMTP, IMAP, HTTP, FTP, …

BGP, DNS, NTP

TCP, UDP

IP

Ethernet MAC, 802.11, ARP

Connectors, voltage levels, …

Data Link Layer

Data Link Layer (Layer 2)

Layer 2 (Ethernet/Wi-Fi switches) generally has weak security

- MAC Attacks – CAM overflow
- VLAN Hopping
- ARP cache poisoning
- DHCP spoofing

Link Layer: CAM overflow

Monitor all traffic on a LAN

Layer 2: Ethernet Switches

TP-Link Switch

- 8 1-GbE ports

Cisco Nexus 9516 Switch

- 1/10/40 GbE
- 21-rack-unit chassis
- Up to 576 1/10 Gb ports

Ethernet MAC addresses

Ethernet frames are delivered based on their 48-bit MAC* address

- Top 24 bits: manufacturer code assigned by IEEE
- Bottom 24 bits: assigned by manufacturer
- `ff:ff:ff:ff:ff:ff` = broadcast address

Ethernet MAC address ≠ IP address

*MAC = Media Access Control address – used as a link-layer address by Ethernet, Wi-Fi, and Bluetooth

How does an Ethernet switch work?

A switch contains a switch table (MAC address table)

– Contains entries for known MAC addresses & their interface

Forwarding & filtering:

a frame arrives for some destination address D

1. Look up *D* in the switch table to find the interface
2. If found & the interface is the same as the one the frame arrived on
   Discard the frame (**filter**)
3. If found & *D* is on a different interface
   Forward the frame to that interface: queue if necessary
4. If not found
   - **Forward** to ALL interfaces

As attackers, we want this to happen.
That way, we get to see all network traffic

The switch table

A switch is self-learning

- **Switch table (MAC address → interface): initially empty**
- **Whenever a frame is received, associate the interface with the source MAC address in the frame**
- **Delete switch table entries if they have not been used for some time**

Switches must be fast: can't waste time doing lookups

- They use CAM – Content Addressable Memory
- Fixed size table

CAM overflow attack

Exploit size limit of CAM-based switch table

- **Send bogus Ethernet frames with random source MAC addresses**
  - Each new address will displace an entry in the switch table
- **With the CAM table full, legitimate traffic will be broadcast to all links**
  - A host on any port can now see all traffic
  - CAM overflow attack turns a switch into a hub

Countermeasures:

Port security

- Some managed switches let you limit # of addresses per switch port

802.1x support

- All traffic from a port is initially "unauthorized" and redirected to an authentication server

dsniff: collection of tools for network auditing and penetration testing
`https://monkey.org/~dugsong/dsniff/`

Link Layer: VLANs & VLAN hopping

Join VLANs you are not a member of

VLANs

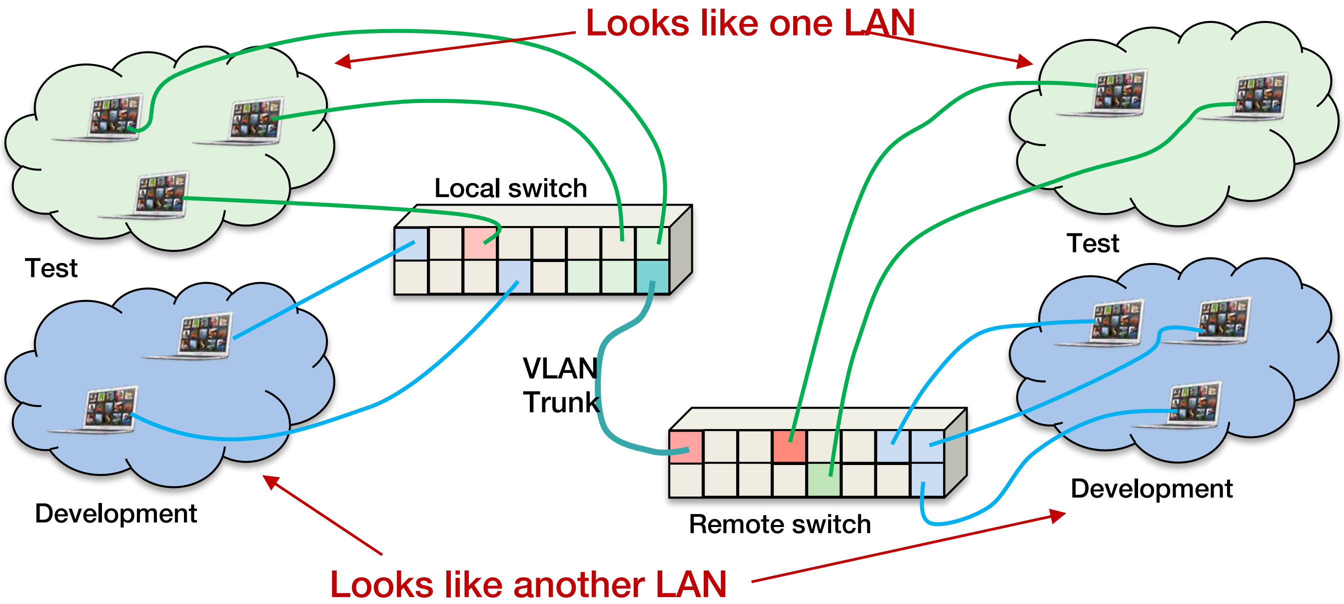
- **A switch & cables creates a local area network (LAN)**
- **We use LANs to**
  - Isolate broadcast traffic from other groups of systems
  - Isolate users into groups
  - What if users move? What if switches are inefficiently used?

- **Virtual Local Area Networks (VLANs)**
  - Create multiple virtual LANs over one physical switch infrastructure
  - Network manager can assign a switch's ports to a specific VLAN
  - Each VLAN is a separate broadcast domain

VLAN Trunking

VLANs across multiple locations/switches

– VLAN Trunking: a single connection between two VLAN-enabled switches carries all traffic for all VLANs

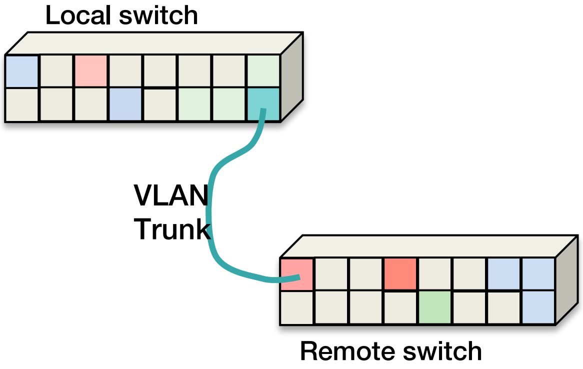


VLAN Hopping Attack

- **VLAN trunk carries traffic for *all* VLANs**
- **Extended Ethernet frame format**
  - 802.1Q for frames on an Ethernet trunk = Ethernet frame + VLAN tag
  - Sending switch adds VLAN tag for traffic on the trunk
  - Receiving switch removes VLAN tag and sends traffic to appropriate VLAN ports based on VLAN ID

Attack: **switch spoofing**

Devices can spoof themselves to look like a switch with a trunk connection and become a member of all VLANs



Avoiding VLAN Hopping

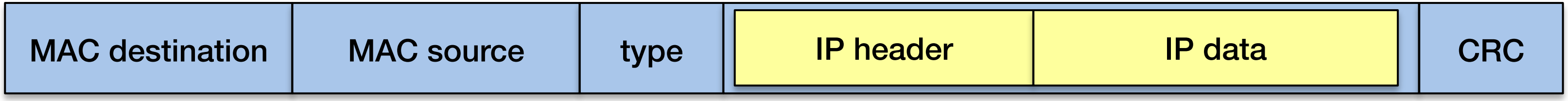
Disable	**Disable unused ports & assign them to an unused VLAN** • Stops an attacker from plugging a device into an unused port
Disable	**Disable auto-trunking** • Stops an attacker from masquerading as a switch
Configure	**Explicitly configure trunking on switch ports that are used for trunks** • Allows legitimate connected switches to work

ARP Cache Poisoning (ARP Spoofing)

Intercept traffic for other IP addresses

Find MAC address given an IP address

- **We need to send a datagram to an IP address**
- **It is encapsulated in an Ethernet frame and a MAC address**



How do we know what MAC address to use?

ls.cs.rutgers.edu:
IP address: `128.6.13.171`
MAC address: `40:b0:34:f6:cd:0f`

ilab1.cs.rutgers.edu:
IP address: `128.6.4.101`
MAC address: `ee:4f:34:13:19:78`

Address Resolution Protocol (ARP)

ARP Table

– Kernel table mapping IP addresses & corresponding MAC addresses
– OS uses this to fill in the MAC header given an IP destination address
– *What if the IP address we want is not in the cache?*

ARP Messages

– A host creates an ARP query packet & broadcasts it on the LAN
  - Ethernet broadcast MAC address: `ff:ff:ff:ff:ff:ff`
– All adapters receive it
  - If an adapter's IP address matches the address in the query, it responds
  - Response is sent to the MAC address of the sender

HW Protocol (ethernet)	Protocol type (e.g., IPv4)	MAC addr length	query/ response	sender MAC addr	sender IP addr	target MAC addr	target IP addr

ARP packet structure

see the *arp* command on Linux/BSD/Windows/macOS

ARP Cache Poisoning

- **Network hosts cache any ARP replies they see … even if they did not originate them … on the chance that they might have to use that IP address**
- **Any client is allowed to send an *unsolicited* ARP reply**
  - Called a gratuitous ARP
- **ARP replies will overwrite older entries in the ARP table … even if they did not expire**
- **An attacker can create fake ARP replies**
  - Containing the attacker's MAC address and the target's IP address
  - This will direct any traffic meant for the target to the attacker
  - Enables man-in-the-middle or denial of service attacks

See *Ettercap* – a multipurpose sniffer/interceptor/logger
https://github.com/Ettercap/ettercap

Defenses against ARP cache poisoning

- **Ignore replies that are not associated with requests**
  - But you have to hope that the reply you get is a legitimate one
- **Use static ARP entries**
  - But can be an administrative nightmare
- **Enable Dynamic ARP Inspection**
  - Validates ARP packets against DHCP Snooping database information or static ARP entries

DHCP Server Spoofing

Configure hosts with your chosen network settings

DHCP (Dynamic Host Configuration Protocol)

Computer joins a network – needs to be configured

- *Broadcasts* a *DHCP Discover* message

A DHCP server picks up this request and sends back a response

- IP address
- Subnet mask
- Default router (gateway)
- DNS servers
- Lease time

Attack:

Spoof responses that would be sent by a valid DHCP server

DHCP Spoofing

- **Anybody can pretend to be a DHCP server**
  – Spoof responses that would be sent by a valid DHCP server
  – Provide:
    - False gateway address
    - False DNS server address
- **Attacker can now direct traffic from the client to go anywhere**
- **The real server may reply too**
  – If the attacker responds first, he wins
  – Attack the server first – delay or disable the real server: denial of service attack

Defenses

Some switches (Cisco, Juniper) support DHCP snooping

- Switch ports can be configured as "trusted" or "untrusted"
- Only specific machines are allowed to send DHCP responses
- The switch will use DHCP data to track client behavior
  - Ensure hosts use only the IP address assigned to them
  - Ensure hosts do not fake ARP responses

Network Layer (IP) vulnerabilities

Network Layer: IP

Responsible for end-to-end delivery of packets

- **No guarantees on message ordering or delivery**
- **Key functions**
  - Routing
    - Each host knows the address of one or more connected routers (gateways)
    - The router knows how to route to other networks
  - Fragmentation & reassembly
    - An IP fragment may be split if the MTU size on a network is too small
    - Reassembled at its final destination
  - Error reporting
    - ICMP messages sent back to the sender (e.g., if packet is dropped)
  - Time-to-live
    - Hop count avoids infinite loops; packet dropped when TTL = 0

Source IP address

No source IP address authentication

- **Clients are *supposed* to use their own source IP address**
  - Can override with raw sockets
  - Responses will be sent to the forged source IP address
- **Enables**
  - Anonymous DoS attacks
  - DDoS attacks
    - Send lots of packets from many places that will cause routers to generate ICMP responses
    - All responses go to the forged source address

Attacks on routers

- **Routers are just special-purpose computers**
  - People may keep default passwords or not use strong passwords
  - Router OS & supporting services may be buggy or not be kept up to date
- **Subject to attacks:**
  - Denial of Service (DOS)
    - Flood the router (e.g., lots of ICMP packets from lots of sources)
  - Routing table poisoning
    - Either by breaking into a router or by sending modified routing data update packets

Router manufacturer	Most common username	Most common password	Confidence
Arris default credentials	admin	password	86%
Aruba Networks default credentials	admin	admin	100%
ASUS default credentials	admin	admin	98%
Belkin default credentials	admin	password	50%
Cisco default credentials	cisco	cisco	38%
D-Link default credentials	admin	-	52%
Hitron default credentials	cusadmin	password	100%
Huawei default credentials	admin	admin	57%
Linksys default credentials	-	admin	70%
MikroTik default credentials	admin	-	97%
Netgear default credentials	admin	password	96%

https://www.router-reset.com/default-router-password-lookup

? TOTOLINK default Password List (Valid March 2022)

Model	Default Username	Default Password	Default IP address
A3 A3 default factory settings	admin	admin	192.168.0.1
A3002RU A3002RU default factory settings	admin	admin	192.168.0.1
A3100R A3100R default factory settings	admin	admin	192.168.0.1
A6004NS A6004NS default factory settings	admin	admin	192.168.1.1
A7000R A7000R default factory settings	admin	admin	192.168.0.1
A810R A810R default factory settings	admin	admin	192.168.0.1
N150RT N150RT default factory settings	admin	admin	192.168.1.1
N600R N600R default factory settings	admin	admin	192.168.0.1

https://www.router-reset.com/default-password-ip-list/TOTOLINK

Just a few recent headlines (`routersecurity.org`)

Mar 26 2024	TheMoon malware infects 6,000 ASUS routers in 72 hours for proxy service
Feb 27 2024	Hackers backed by Russia and China are infecting SOHO routers like yours, FBI warns
Feb 28, 2024	That home router botnet the Feds took down? Moscow's probably going to try again
Feb 15, 2024	Feds dismantle Russian GRU botnet built on 1,000-plus home, small biz routers
Jan 31, 2024	FBI disrupts Chinese botnet by wiping malware from infected routers
Oct 9, 2023	IZ1H9 Campaign Enhances Its Arsenal with Scores of Exploits
Oct 18, 2023	D-Link clears up 'exaggerations' around data breach
Sep 27, 2023	People's Republic of China-Linked Cyber Actors Hide in Router Firmware

https://routersecurity.org/RouterNews.php https://routersecurity.org/bugs.php

Transport Layer (UDP, TCP) vulnerabilities

TCP & UDP

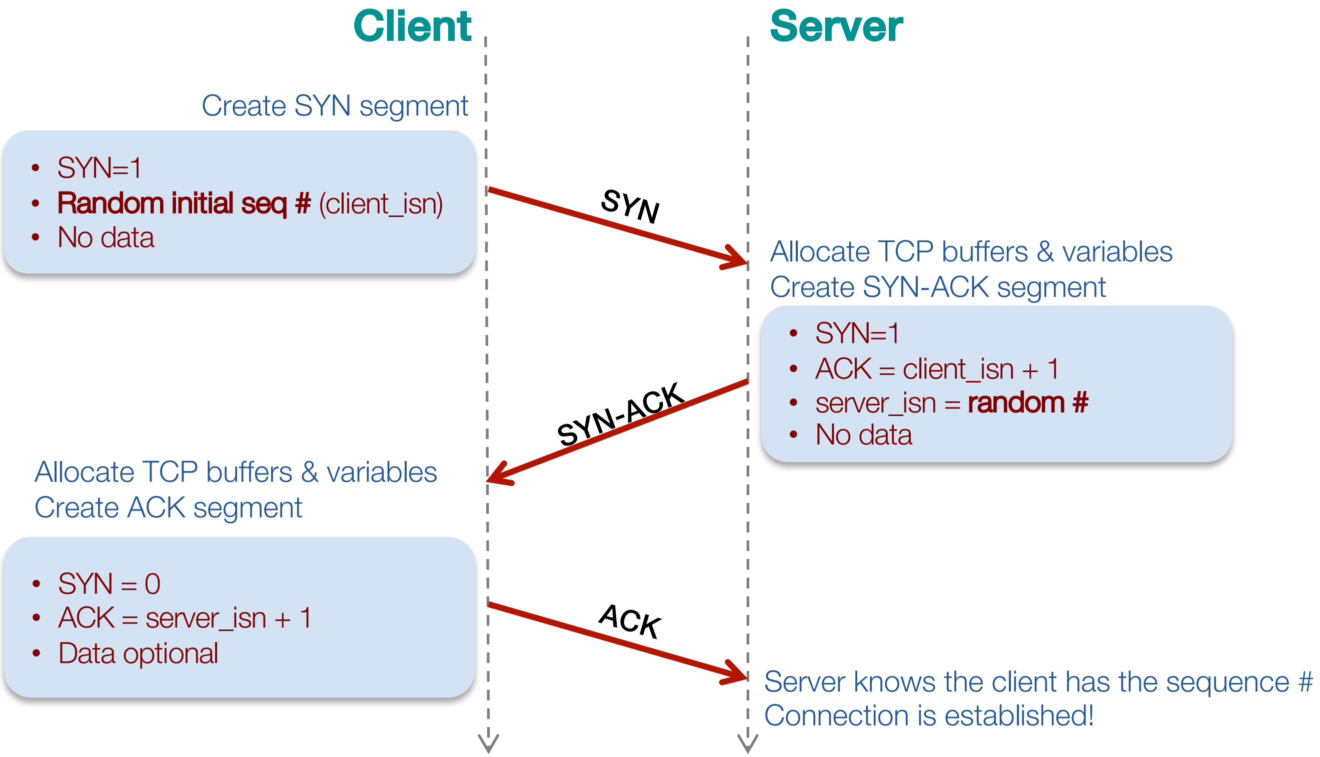
UDP: User Datagram Protocol

- Stateless, connectionless & unreliable
- Anyone can send forged UDP messages

TCP: Transmission Control Protocol

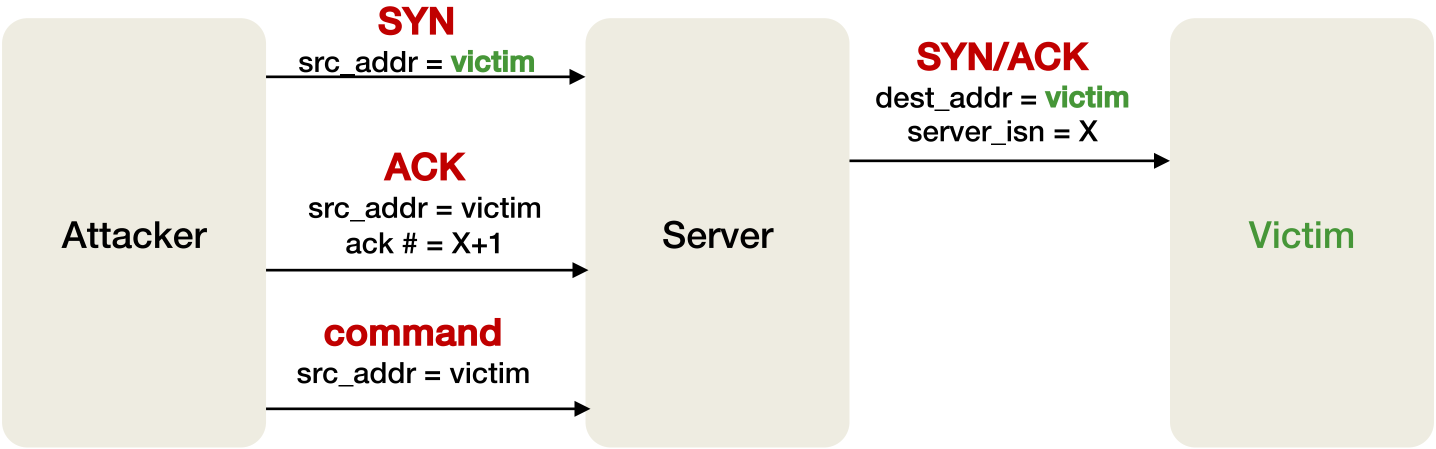
- Stateful, connection-oriented & reliable
- Every packet contains a sequence number (byte offset)
  - Receiver assembles packets into correct order
  - Sends acknowledgements
  - Missing packets are retransmitted

TCP connection setup: three-way handshake



Why random initial sequence numbers?

If predictable, an attacker can create a TCP session on behalf of a forged source IP address



Random numbers make this attack harder – especially if the attacker cannot sniff the network

Denial of service: SYN Flooding

An OS will allocate only a finite # of TCP buffers

- **SYN Flooding attack**
  - Send lots of SYN segments but never complete the handshake
  - The OS will not be able to accept connections until those time out
- **SYN Cookies: Dealing with SYN flooding attacks**
  - Do not allocate buffers & state when a SYN segment is received
  - Create initial sequence # =
    hash(src_addr, dest_addr, src_port, dest_port, SECRET)
  - When an ACK comes back, validate the ACK #
    Compute the hash as before & add 1
  - If valid, then allocate resources necessary for the connection & socket

Denial of service: Reset

- **Attacker can send a RESET (RST) packet to an open socket**
- **If the server sequence number is correct, then the connection will close**
- **Sequence numbers are 32 bits**
  - Chance of success is $1/2^{32} \approx$ 1 in 4 billion
  - But many systems allow for a large range of sequence numbers
  - Attacker can send a flood of RST packets until the connection is broken

Network Routing Protocols

IP Routing Protocols

Network operators (autonomous systems) need to know how to route packets within their network and the best connection to use for packets that are routed outside their network

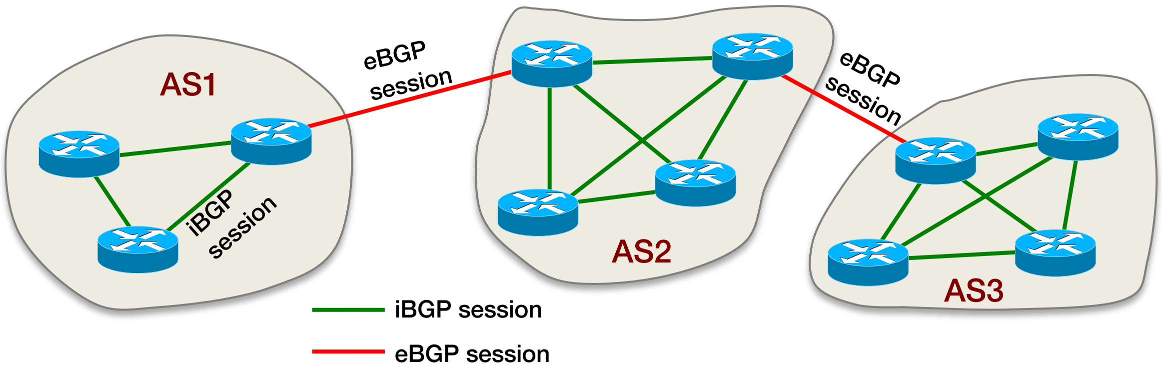
OSPF: Open Shortest Path First

- Interior Gateway Protocol (IGP) within an autonomous system (AS)
- Uses a link state routing algorithm (Dijkstra's shortest path)

BGP: Border Gateway Protocol

- Exterior Gateway Protocol (EGP) between autonomous systems (AS)
- Network operators exchange routing and reachability information
  - Each sends a list of blocks of addresses they can route to and the distance to each block
  - Identifies the owner and AS route to reach the owner
- Distance vector routing protocol

BGP sessions maintained via TCP links

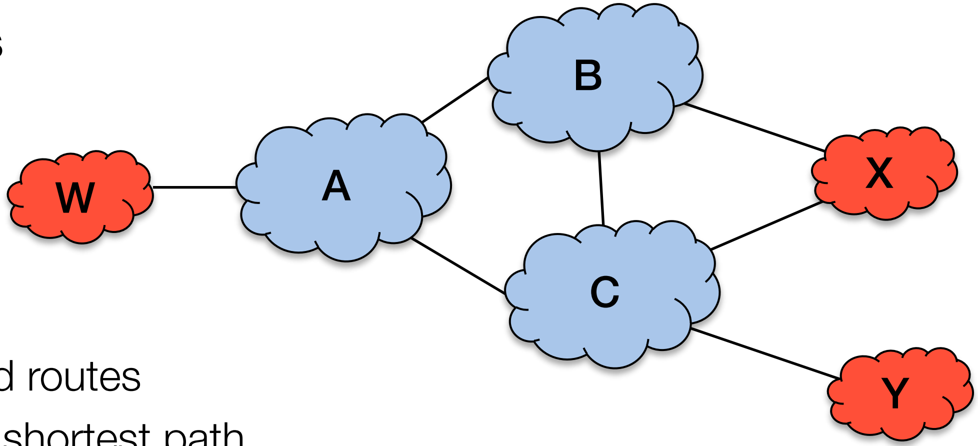


Pairs of routers exchange information via semi-permanent TCP connections

- One connection for each link between gateway routers
  - External BGP (eBGP) session
- Also, BGP TCP connections between routers *inside* an AS
  - Internal BGP (iBGP) session

Route selection

- **A, B, C: transit ASes – ISPs & backbone**
- **W, X, Y: stub ASes – customers**



BGP route selection

- Policies allow selection of preferred routes
- Otherwise, pick the route with the shortest path
- If there’s a tie, choose the shortest path with the closest router

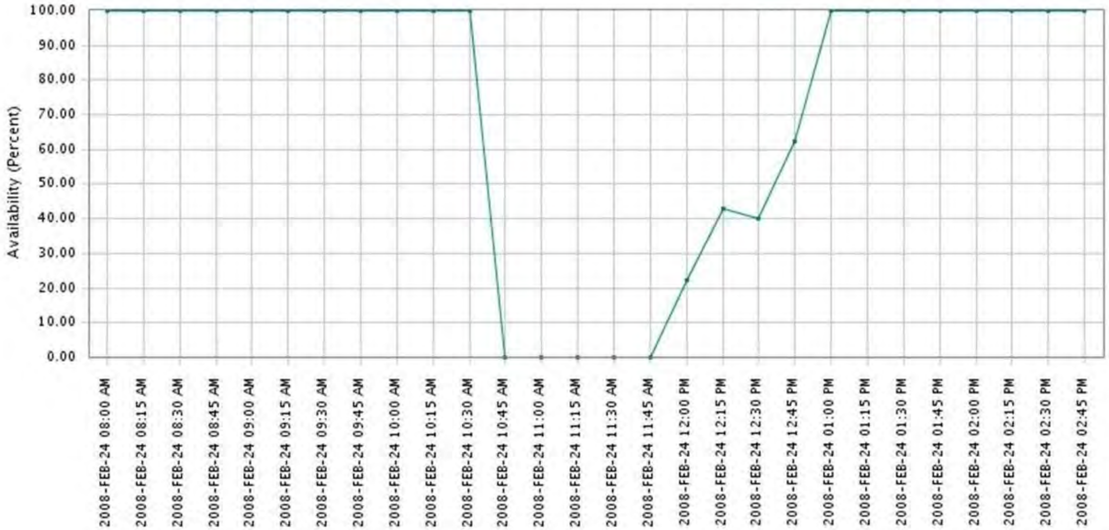
BGP Hijacking

- **BGP was built based on trust**
  - Each network operator trusts others & believes the information it receives is accurate
  - The trust is a chain: a network operator sends route advertisements that are built from data it received from other network operators

- **Route advertisements are not authenticated**
  - A malicious network operator can inject advertisements for arbitrary routes
  - Information will propagate throughout the Internet
  - Can be used for DoS (dropping packets), eavesdropping, man-in-the-middle attacks, or redirecting traffic to malicious computers

Pakistan's attack on YouTube in 2008

- **YouTube service was cut off the global web for over an hour**
- **Pakistan Telecom received a censorship order from the telecommunications ministry to block YouTube**
  – The company sent spoofed BGP messages claiming to be the best route for YouTube's range of IP addresses



Pakistan's attack on YouTube in 2008

- **Pakistan Telecom sent BGP advertisements that it was the correct route for 256 addresses in YouTube's 208.65.153.0 network**
  - Advertise a /24 network
- **That is a more specific destination than YouTube's broadcast, which covered 1024 addresses**
  - YouTube advertised a /22 network
  - Within minutes, all YouTube traffic started to flow to Pakistan
- **YouTube immediately tried countermeasures**
  - Narrowed its broadcast to 256 addresses … but too late
  - Then tried an even more specific group: 64 addresses

    Advertise a /26 network ⇒ priority over /24 routes

    - Routes for more specific addresses overrule more general ones
  - Route updates were finally fixed after 2 hours

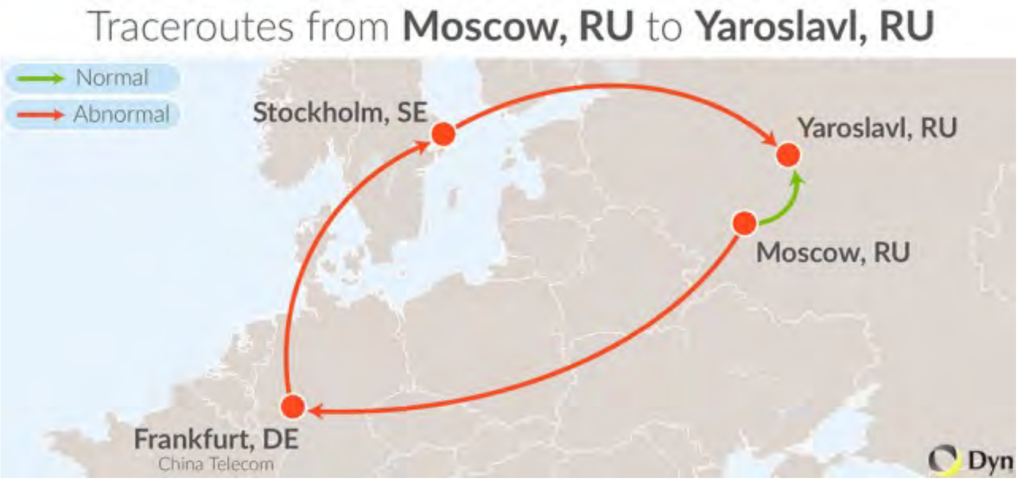
2013 – Repeated attacks

- **38 events observed where traffic to 1,500 blocks of IP addresses was redirected to Iceland or Belarus**
  - Redirection ranged from a few minutes to several days
  - Over 60 days of man-in-the-middle attacks observed
- **Data targeted to 150 cities was intercepted**

URL: https://arstechnica.com/information-technology/2013/11/repeated-attacks-hijack-huge-chunks-of-internet-traffic-researchers-warn/

2014 – Russian traffic routed through China

- **Russian domestic traffic was repeatedly rerouted to routers operated by China Telecom**
- **Occurred after Russian mobile provider Vimpelcom and China Telecom signed a peering agreement to carry traffic over each other's network at no cost**
- **The rerouting could have been a configuration error**
  - But could also have been espionage or hacking



https://arstechnica.com/information-technology/2014/11/wtf-russias-domestic-internet-traffic-mysteriously-passes-through-china/

2017 – Selected traffic routed to Russian ISP

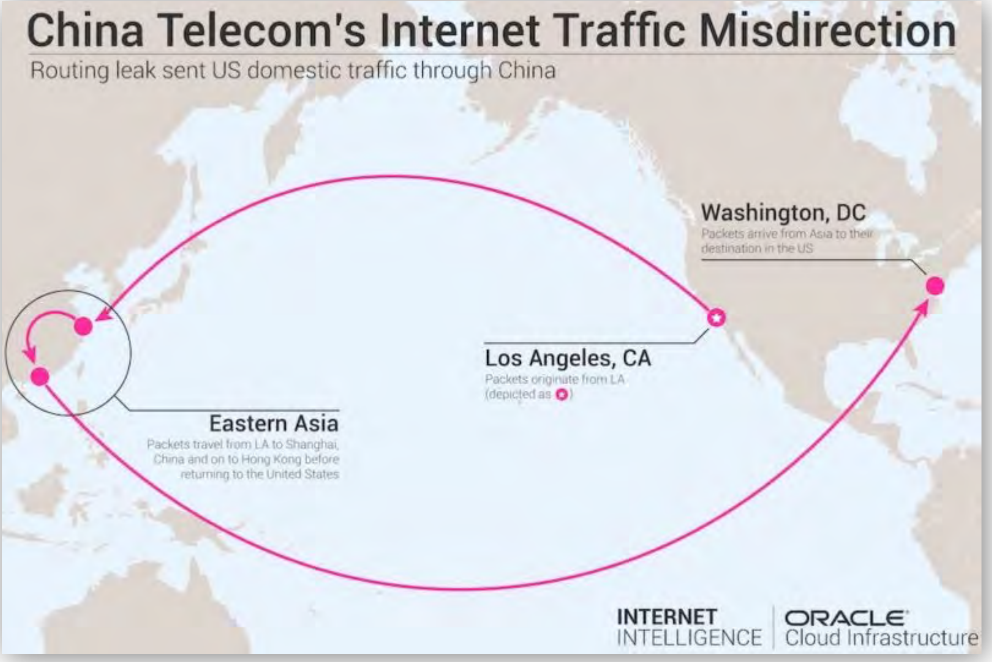
- **Traffic belonging to Google, Facebook, Apple, Microsoft, Twitch, and Riot Games was routed through a Russian Internet provider**
  - Eight months earlier, traffic for MasterCard, Visa, and more than two dozen other financial services was routed through a Russian government-controlled telecom
- **Considered suspicious & not a configuration error**
  - Targeted very specific companies
  - Advertised IP address blocks were broken into small chunks
    - BGP prioritizes more specific blocks of addresses; this ensures they get selected over broader advertisements for the same groups of addresses

https://arstechnica.com/information-technology/2017/12/suspicious-event-routes-traffic-for-big-name-sites-through-russia/

https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/

2015-2018 – Traffic redirected to China

- **China Telecom redirected large chunks of Internet traffic through their routers**
- **This took place for 2.5 years**
- **China Telecom was incorrectly advertising routes to Verizon's Asia-Pacific AS (AS703)**
  - Packets would be routed to China
- **Could have been a configuration error**



https://arstechnica.com/information-technology/2018/11/strange-snafu-misroutes-domestic-us-internet-traffic-through-china-telecom/

Internet Society

Mutually Agreed Norms for Routing Security (MANRS) 25 April 2018

Another BGP Hijacking Event Highlights the Importance of MANRS and Routing Security

By Megan Kruse

Another BGP hijacking event is in the news today. This time, the event is affecting the Ethereum cryptocurrency. Users were faced with an insecure SSL certificate. Clicking through that, like so many users do without reading, they were redirected to a server in Russia, which proceeded to empty the user's wallet. …

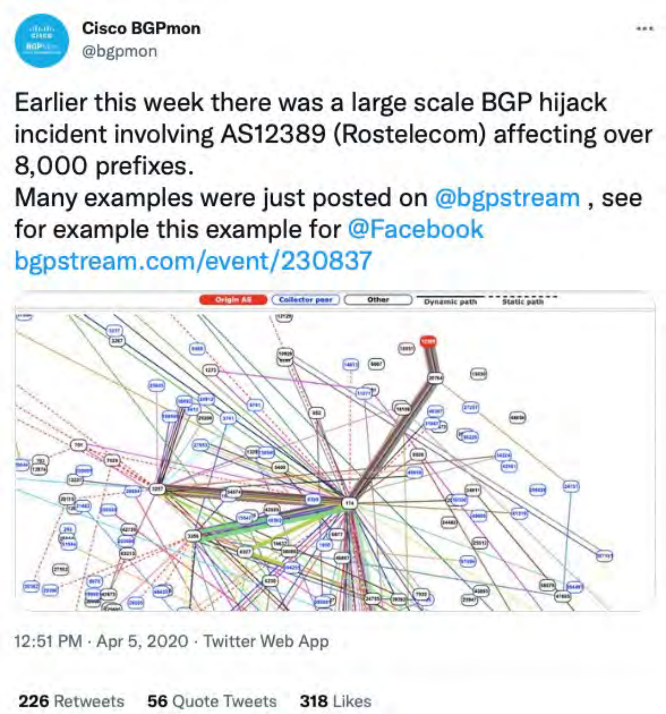
In this case specifically, the culprit re-routed DNS traffic using a man in the middle attack using a server at an Equinix data center in Chicago. Cloudflare has put up a blog post that explains the technical details. From that post:

"This [hijacked] IP space is allocated to Amazon(AS16509). But the ASN that announced it was eNet Inc(AS10297) to their peers and forwarded to Hurricane Electric(AS6939)."

https://www.internetsociety.org/blog/2018/04/another-bgp-hijacking-event-highlights-the-importance-of-manrs-and-routing-security/

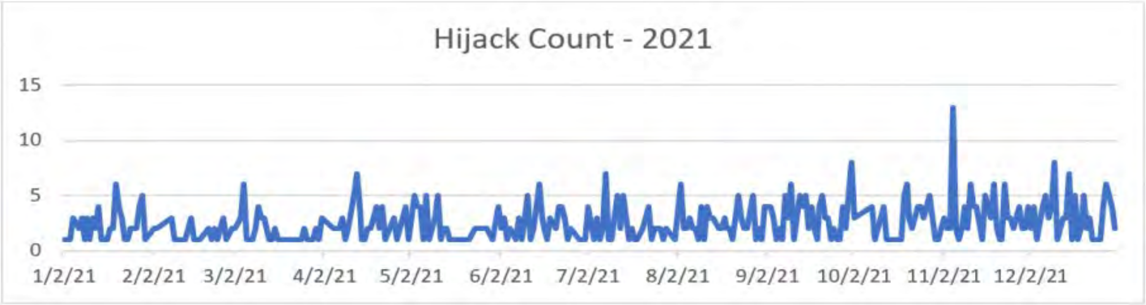
2020 – Traffic from >200 providers redirected to Russia

- Traffic for content delivery networks and cloud providers was redirected through Rostelecom, Russia's state-owned telecom provider
- Affected over 8,800 routes from over 200 networks
- Lasted for an hour
- Companies affected included Google, Amazon, Facebook, Akamai, and Cloudflare
- Could have been a mistake

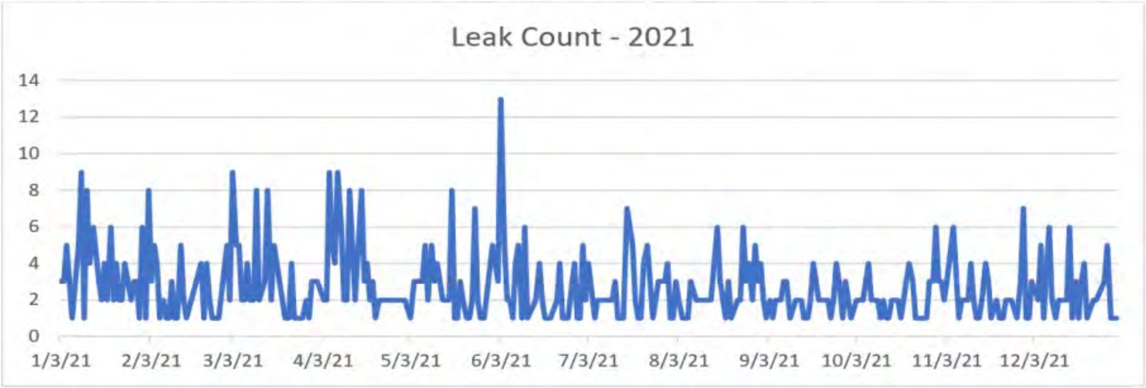


https://www.zdnet.com/article/russian-telco-hijacks-internet-traffic-for-google-aws-cloudflare-and-others/

BGP Hijacking continued through 2021



775 incidents flagged as possible BGP hijacks (likely malicious)



830 incidents flagged as possible BGP leaks (likely configuration errors)

Cisco BGPStream.com data

https://www.manrs.org/2022/02/bgp-security-in-2021/

2022 – Klayswap crypto attack

- **Hackers stole almost $2M from South Korean cryptocurrency platform KLAYswap**
- **Used a rogue autonomous system**
  - Advertised IP address for `developers.kakao.com`
  - `developers.kakao.com` hosts the Kakao SDK used by third-party developers
- **Attackers hijacked the address and served a malicious version of KakaoTalk's JavaScript SDK file**

  `https://developers.kakao.com/sdk/js/kakao.min.js`
- **Users thought they were downloading it from the official site, but it came from the attacker's servers**
- **Code waited for a transaction and transferred funds to an attacker's wallet**
- **Attack lasted two hours and incurred 407 transactions across 325 customer wallets**

https://therecord.media/klayswap-crypto-users-lose-funds-after-bgp-hijack/

Some Twitter traffic briefly funneled through Russian ISP, thanks to BGP mishap

arstechnica

Despite the timing, the 45-minute hijacking was most likely an error, not an attack.

Dan Goodin • March 28, 2022

Some Internet traffic in and out of Twitter on Monday was briefly funneled through Russia after a major ISP in that country misconfigured the Internet's routing table, network monitoring services said.

The mishap lasted for about 45 minutes before RTCOMM, a leading ISP in Russia, stopped advertising its network as the official way for other ISPs to connect to the widely used Twitter IP addresses. Even before RTCOMM dropped the announcement, safeguards prevented most large ISPs from abiding by the routing directive.

A visualization of what the event looked like is illustrated on this page from BGPStream.

Remember BGP

The border gateway protocol is the means by which ISPs in one geographical region locate and connect to ISPs in other areas. The system was designed in the early days of the Internet, when operators of one network knew and trusted their peers running other networks. Typically, one engineer would use BGP table to "announce" that their network—known as an "autonomous system" in BGP parlance—was the correct path to send and receive traffic to specific networks.

https://arstechnica.com/information-technology/2022/03/absence-of-malice-russian-isps-hijacking-of-twitter-ips-appears-to-be-a-goof/

Attempts to defend against BGP Hijacking

RPKI (Resource Public Key Infrastructure) framework

Standardized in 2012
See RFC 6480

- Provides a way to validate that the AS that is making the route announcement is authorized to do so for the addresses it is advertising – it secures the origin of route announcements.
  - Each AS obtains an X.509 certificate from the Regional Internet Registry (RIR)
  - AS admin creates a Route Origin Authorization (ROA) – signed by the AS's private key
  - The ROA identifies which network operator is allowed to announce an organization's IP addresses using BGP
  - Route advertisements without a valid, signed ROA are ignored
- Only about 43% of advertised routes use ROA as of May 2023
- Doesn't stop all hijacks: a malicious AS can still send BGP *UPDATE* messages claiming it's connected to the legitimate owner to capture traffic to that owner because of its low hop count

Standardized in 2017
See RFC 8206

See: https://labs.ripe.net/author/niklas-vogel/crashing-the-party-vulnerabilities-in-rpki-relying-party-software/

Attempts to defend against BGP Hijacking

Standardized in 2017
See RFC 8206

BGPsec

- Security enhanced version of the BGP protocol
- Protects integrity of BGP update messages – each AS adds its signature to the advertised route
- Downgrade attacks possible if all ASes don't support BGPsec
- Requires all ~71,000 ASes to be able to send *UPDATE* messages to each other

See: https://labs.ripe.net/author/niklas-vogel/crashing-the-party-vulnerabilities-in-rpki-relying-party-software/

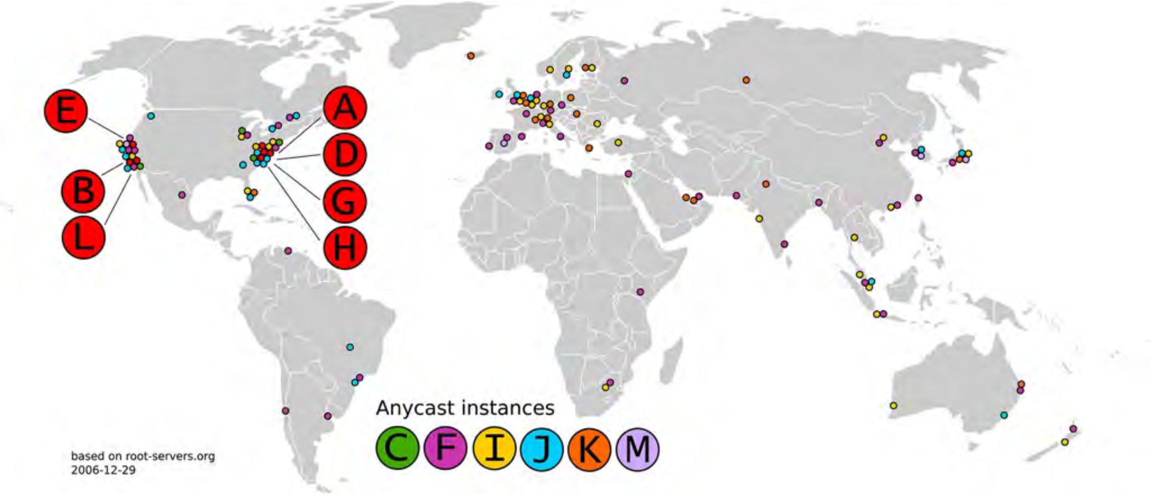
Domain Name System (DNS) Vulnerabilities

Domain Name System

- **Hierarchical service to map domain names to IP addresses**
- **How do you find the DNS Server for rutgers.edu?**
  - That's what the domain registry keeps track of
  - When you register a domain
    - You supply the addresses of at least two **DNS servers** that can answer queries for your zone
    - You give this info to the **domain registrar** (e.g., Namecheap, GoDaddy) who updates the database at the **domain registry** (e.g., Verisign for .com, .net, .edu, .gov, … domains)
      - **Domain registrar:** Sells domain names to the public
      - **Domain regsitry:** Maintains the top-level domain database
- ***So how do you find the right DNS server?***
  - Start at the root

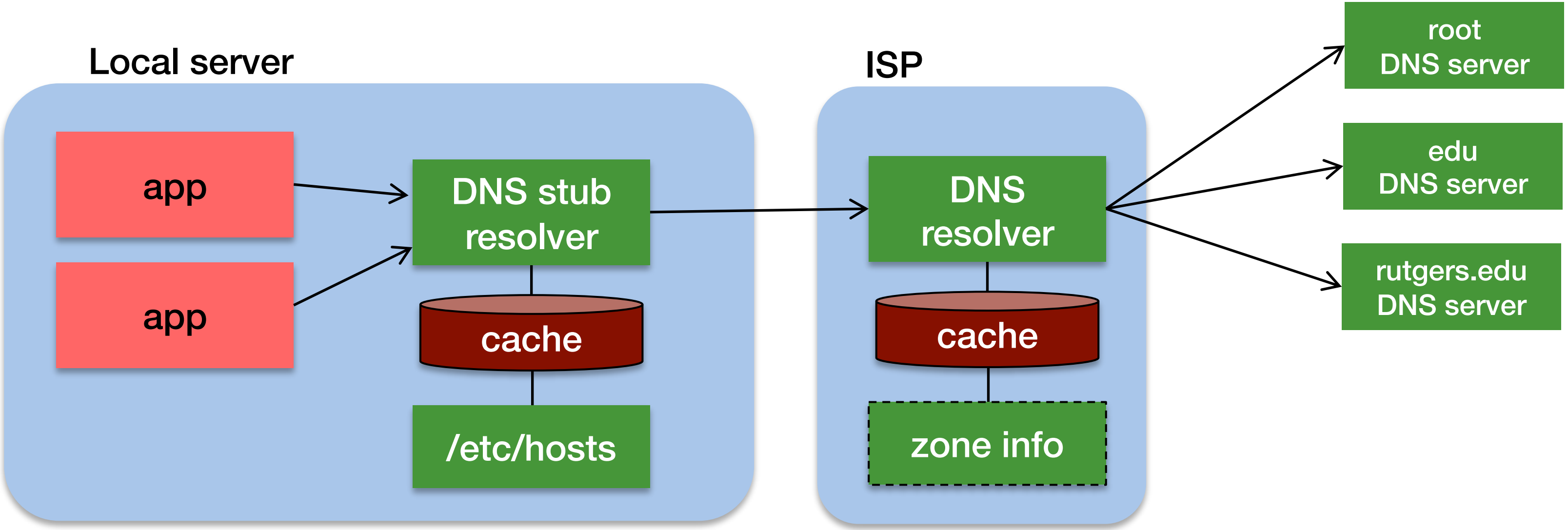
Root name servers

- The **root name servers** provide lists of authoritative name servers for top-level domains
- 13 root name servers
  - `A.ROOT-SERVERS.NET`, `B.ROOT-SERVERS.NET`, …
  - Each has redundancy (via *anycast* routing or load balancing)
    - Each server is really a set of machines



Download the latest list at http://www.internic.net/domain/named.root

DNS Resolvers in action



<u>Local stub resolver</u>:
- check local cache
- check local hosts file
- send request to external resolver

<u>External resolver</u>:
- Running at ISP, Cloudflare, Google Public DNS, OpenDNS, etc.

E.g., on Linux: resolver is configured via the `/etc/resolv.conf` file

DNS Vulnerabilities

Programs (and users) trust the host-address mapping

- This is the basis for some security policies
  - Browser same-origin policy, URL address bar

- **But DNS responses can be faked**
  - If an attacker gives a DNS response first, the host will use that
  - Malicious responses can direct messages to different hosts
  - A receiver cannot detect a forged response

- **DNS resolvers cache their results (with an expiration)**
  - If it gets a forged response, the forged results will be passed on to any systems that query it

Pharming attack

Redirect traffic to an attacker's site by modifying how the DNS resolver gets its information

Forms of attack

1. Use malware or social engineering to modify a computer's *hosts* file

   This file maps *names→IP addresses* and avoids DNS queries

2. Attack the router & modify its DNS server setting

   Direct traffic to the attacker's DNS server, which will give the wrong IP address for certain domain names

DNS spoofing attack

Redirect traffic to an attacker via DNS **cache poisoning**

- **An attacker sends the wrong DNS response**
  - The DNS resolver requesting it will cache it and provide that to anyone else who asks in the near future
- **How do we prevent spoofed responses?**
  - Each DNS query contains a 16-bit Query ID (**QID**) – only 65,536 to guess
    - Response from the DNS server must have a matching QID
  - DNS uses UDP and this was created to make it easy for a system to match responses with requests
- **An attacker will have to guess the QID number**
  - But Query IDs were sequential and not hard to guess
  - Fix by using random Query IDs

DNS spoofing via Cache Poisoning

What happens?

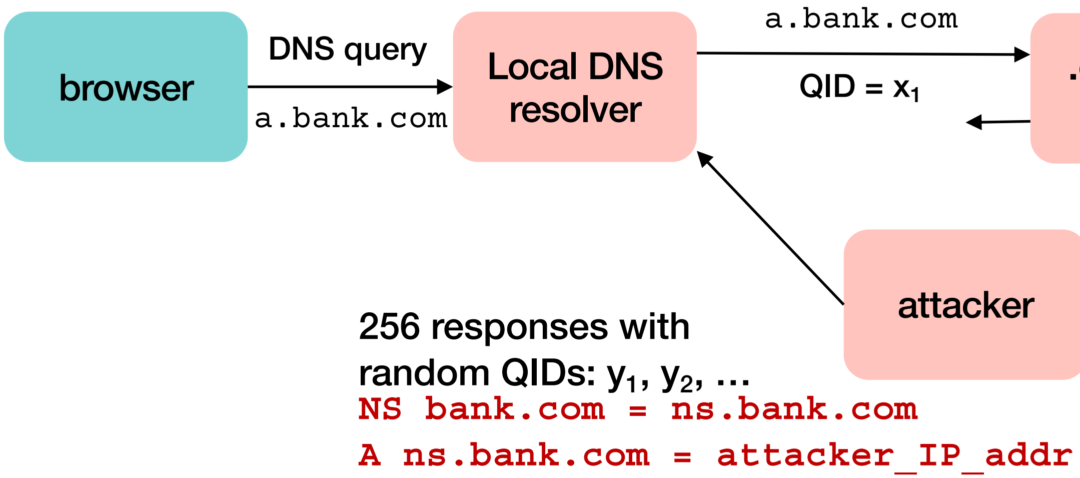
- Malicious JavaScript on a web page causes the client to try to look up `a.bank.com`, `b.bank.com`, etc.
- At the same time, the attacker is sending a stream of DNS "responses" hoping that one will have a matching query ID (QID)

If the attacker is successful, one of the responses matches up

- But we expect the victim to go to **bank.com**, not **f.bank.com**
- However….
  The DNS response can also define a new DNS server for bank.com!
- This overwrites any saved DNS info for `bank.com` that may be cached
- The attacker can take over any requests to bank.com!

DNS spoofing via Cache Poisoning

JavaScript on a website may launch a DNS attacker



If there is some j such that $x_1 = y_j$ then the response will be cached

All future DNS queries for anything at bank.com will go to attacker_IP_addr

If it doesn't work … try again with `b.bank.com`, `c.bank.com`, etc.

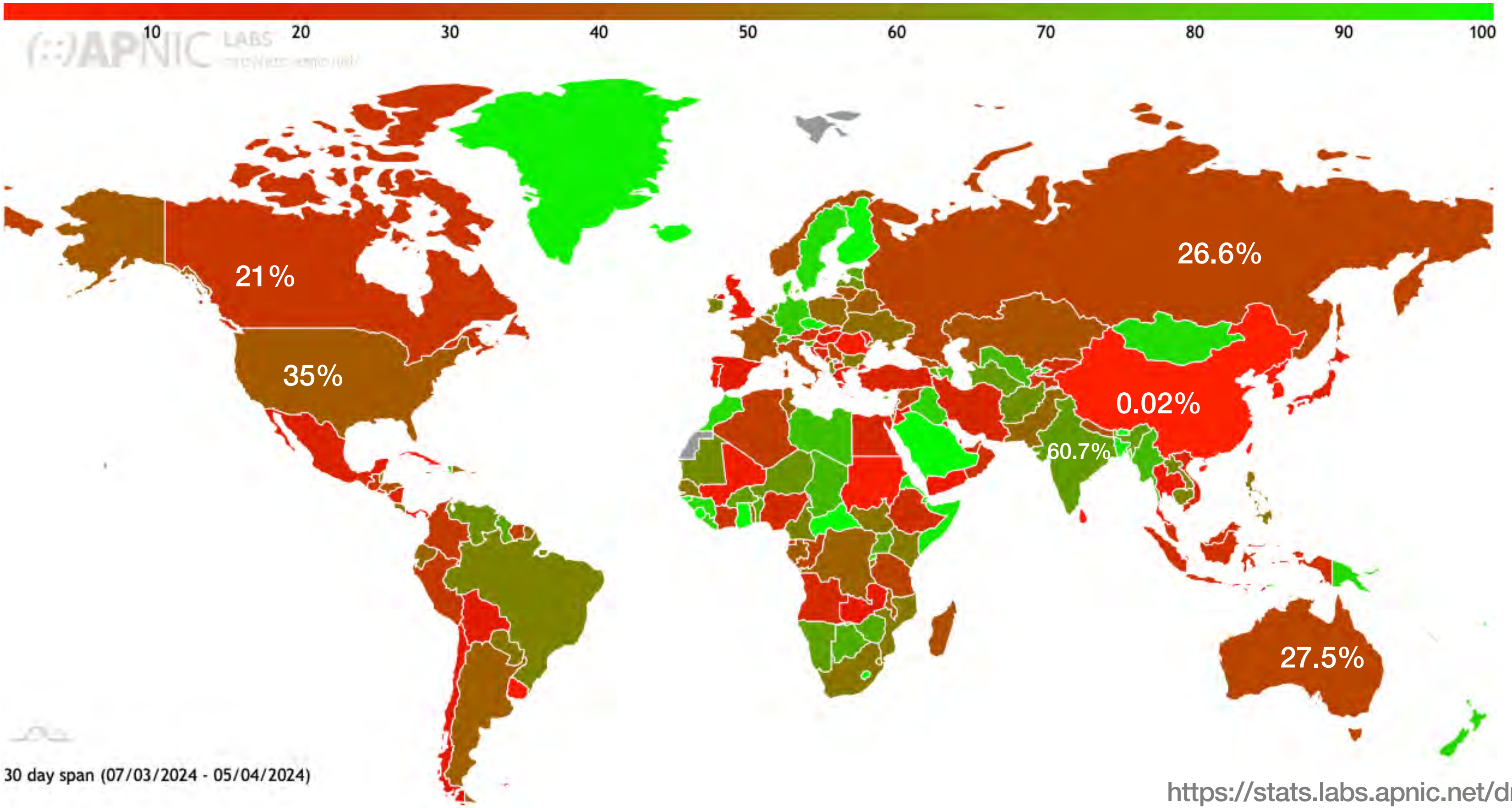
Defenses against DNS cache poisoning

- **Query IDs used to be predictable**
  - Easy to guess
  - Have a web page make a DNS query to a domain under the attacker's control & look at the QID
  - The attacker can then guess the next one
- **Randomize source port # – *where DNS queries originate***
  - Attack will take several hours instead of a few minutes
  - Will have to send responses to a range of ports
  - But this is tricky in real environments that use NAT (network address translation) and may limit the exposed UDP ports
- **Issue double DNS queries**
  - Attacker will have to guess the Query ID twice (32 bits)

Defenses against DNS cache poisoning

- **Use TCP instead of UDP for DNS queries**
  - It's much harder to inject a response into a TCP stream
  - But
    - Much higher latency
    - Much more overhead at the DNS resolver
- **The better long-term solution: DNSSEC**
  - Secure extension to DNS that provides authenticated responses
  - Responses contain a digital signature
  - But
    - Adoption has been very slow
    - DNSSEC response size is much bigger than a DNS response, which makes it more powerful for DoS attacks

Current DNSSEC Deployment



https://stats.labs.apnic.net/dnssec

DNS Rebinding

DNS Rebinding

Attack that allows attackers to run a script to attack other systems on the victim's private network

- **The web's security model relies on comparing domain names**
- **If we can change the underlying address:**
  - We can send messages other systems
    … while the browser thinks it's still going to the same domain
  - This can let us access private machines in the user's local area network
  - Example: access local web services, cameras, thermostats, printers, …

DNS Rebinding

- **Attacker**
  - Registers a domain (`attacker.com`)
  - Sets up a DNS server
  - DNS server responds with very short TTL values – response won't be cached
- **Client (browser)**
  - Script on page causes access to a malicious domain
  - Attacker's DNS server responds with IP address of a server hosting malicious client-side code
  - Malicious client-side code makes additional references to the same domain name
    - This is allowed under the web's **same-origin policy**
      - Scripts in a page may access data in another page only if both pages have the same origin (protocol, address, port)
    - Because of the short TTL, the script causes the system to issue a new DNS request
    - The attacker's DNS server replies with a new IP address (e.g., a target somewhere in the victim's LAN)
    - The script can continue to access content in the same domain
      - But it really isn't in the same domain!

Defending against DNS rebinding

- **Force minimum TTL values**
  - This may affect some legitimate dynamic DNS services
- **DNS pinning: refuse to switch the IP address for a domain name**
  - This is similar to forcing minimum TTL values
  - But this can mess up load balanced or other dynamic services
- **Have the local DNS resolver make sure DNS responses don't contain private IP addresses**
- **Server-side defense within the local area network**
  - Reject HTTP requests with unrecognized `Host` headers
  - Authenticate users

The End