Lecture 11: Network Attacks

These notes were scribed by students in CSE 127 during Winter 2021. They have been lightly edited but may still contain errors.

1 Introduction

These are three basic security properties that come up often in classical information security. They are often referred to the CIA triad.

1. **C**onfidentiality: Permission is required to read data and communications.
2. **I**ntegrity: Permission is required to manipulate data and communications.
3. **A**vailability: Data and communications are accessible to the user.

It will be useful to keep these properties in mind as we discuss different types of network attacks.

Network attacks can be categorized into a few broad types as listed below from most local to least local:

1. **Physical Access**: The attacker has physical access to the network.
2. **In Path/Man in the Middle**: The attacker is between a communicator and the intended recipient and can see, add, and block packets.
3. **On Path/Man on the Side**: The attacker can only see, add, and potentially copy packets.
4. **Passive**: The attacker can only see network traffic.
5. **Off Path**: The attacker can't see network traffic, but can potentially send data.

Here are the layers of the OSI model that we'll be focusing on:

1. **Application** (e.g. DNS, HTTP, HTTPS)
2. **Transport** (e.g. TCP, UDP)
3. **Network** (e.g. IP, BGP)

4. **Data Link** (e.g. Ethernet, WiFi, ARP)

5. **Physical** (e.g. physical wires, photons, RF modulation)

Each layer is vulnerable to its own classes of attacks and layers can sometimes interact with each other.

2 Physical and Link Layer Threats

2.1 Eavesdropping

Eavesdropping is an example of an attack on the physical layer. It is a passive attack that violates confidentiality.

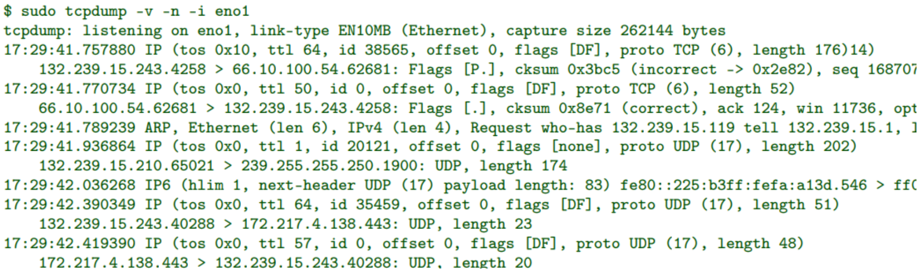
By design, the network infrastructure (routers, switchers, access points) must observe all the traffic passing through in order to route it to the correct destination. An attacker with access to the network infrastructure can take advantage of this by listening in on this traffic. This allows an attack to view the metadata and contents of packets sent over the network.

Examples of ways an attacker can gain access to the network include:

1. Connecting to an unprotected WiFi network.

2. An attacker exploits a well known security vulnerability in WPA2 to learn the network password and connect to the network.

3. An attacker connects to a classical "shared" Ethernet network which broadcasts traffic to everyone connected to the same hub.



```
$ sudo tcpdump -v -n -i eno1
tcpdump: listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:29:41.757880 IP (tos 0x10, ttl 64, id 38565, offset 0, flags [DF], proto TCP (6), length 176)14)
    132.239.15.243.4258 > 66.10.100.54.62681: Flags [P.], cksum 0x3bc5 (incorrect -> 0x2e82), seq 168707
17:29:41.770734 IP (tos 0x0, ttl 50, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    66.10.100.54.62681 > 132.239.15.243.4258: Flags [.], cksum 0x8e71 (correct), ack 124, win 11736, opt
17:29:41.789239 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 132.239.15.119 tell 132.239.15.1, l
17:29:41.936864 IP (tos 0x0, ttl 1, id 20121, offset 0, flags [none], proto UDP (17), length 202)
    132.239.15.210.65021 > 239.255.255.250.1900: UDP, length 174
17:29:42.036268 IP6 (hlim 1, next-header UDP (17) payload length: 83) fe80::225:b3ff:fefa:a13d.546 > ff0
17:29:42.390349 IP (tos 0x0, ttl 64, id 35459, offset 0, flags [DF], proto UDP (17), length 51)
    132.239.15.243.40288 > 172.217.4.138.443: UDP, length 23
17:29:42.419390 IP (tos 0x0, ttl 57, id 0, offset 0, flags [DF], proto UDP (17), length 48)
    172.217.4.138.443 > 132.239.15.243.40288: UDP, length 20
```

Figure 1: Network eavesdropping using tcpdump

Tools used for eavesdropping on a network include command line utilities like tcpdump (Figure 1) and GUI utilities like Wireshark. Eavesdropping is illegal in United States. These tools should not be used on public networks without permission.

In 2006, whistle blower Mark Klein reported that the NSA was using AT&T's network infrastructure (Figure 2) to observe the Internet traffic passing through. Figure 3 is part of the documentation for a fiber-optic splitter used in this

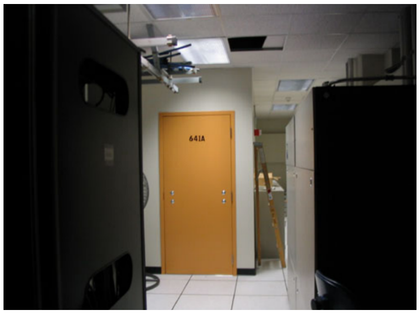
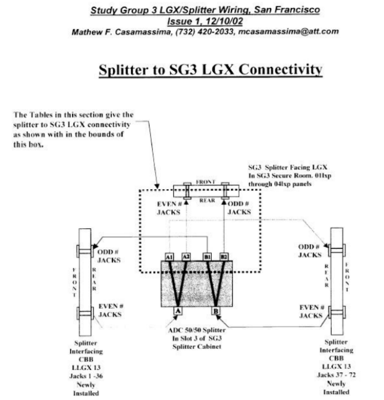


Figure 2: A photo of a room used by the NSA to spy on traffic handled by AT&T in the same building



Study Group 3 LGX/Splitter Wiring, San Francisco
Issue 1, 12/10/02
Mathew F. Casamassima, (732) 420-2033, mcasamassima@att.com

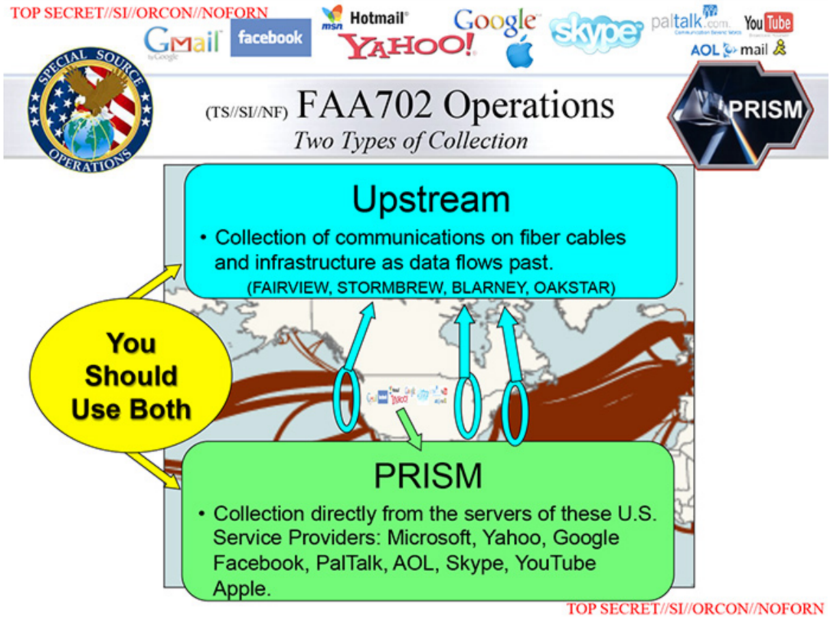
Splitter to SG3 LGX Connectivity

The Tables in this section give the splitter to SG3 LGX connectivity as shown with in the bounds of this box.

Figure 3: Fiber splitters used by the NSA to intercept network traffic

infrastructure that he included in a declaration in the legal filings for court cases that the EFF filed against the US government alleging illegal wiretapping. The cases were dismissed for lack of standing, because none of the plaintiffs could prove that they were actually surveilled by the US government using this wiretapping infrastructure.

The Edward Snowden leaks in 2013 revitalized the political discussion over NSA wiretapping of internet traffic. Figure 4 shows a classified slide that lists some examples of programs that appear to collect network traffic for intelligence purposes.



TOP SECRET//SI//ORCON//NOFORN

(TS//SI//NF) FAA702 Operations

Two Types of Collection

Upstream

- Collection of communications on fiber cables and infrastructure as data flows past. (FAIRVIEW, STORMBREW, BLARNEY, OAKSTAR)

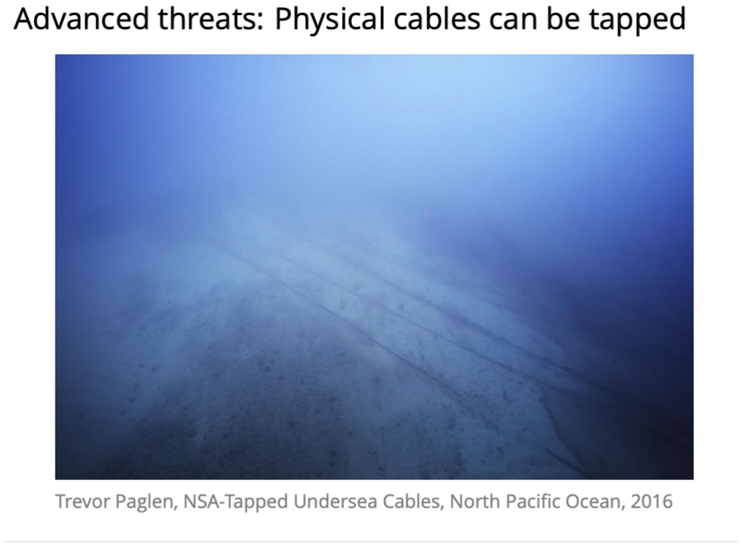
You Should Use Both

PRISM

- Collection directly from the servers of these U.S. Service Providers: Microsoft, Yahoo, Google Facebook, PalTalk, AOL, Skype, YouTube Apple.

TOP SECRET//SI//ORCON//NOFORN

Figure 4: Snowden document on program Prism, 3rd slide



Advanced threats: Physical cables can be tapped

Trevor Paglen, NSA-Tapped Undersea Cables, North Pacific Ocean, 2016

Figure 5: Underwater cables tapped by the NSA

Figure 5 is an photograph taken by artist Trevor Paglen, who has done a lot of really interesting art on government surveillance. To take this photograph, he studied leaked NSA documents to figure out which undersea cables had been tapped by the NSA, studied publicly available documents to locate these tapped cables on the sea floor, and learned to deep water dive to find the cables themselves on the sea floor.

This photograph reminds us that the Internet is a collection of physical objects. Traffic we send crosses thousands of miles of physical cable, some of which may be at the bottom of the ocean. These cables can get dragged up by currents, broken by boat anchors, or damaged in any number of ways. The physical manifestation of the internet as cables and routers and other physical network infrastructure means that we cannot entirely eliminate the possibility of physical taps or disruption.

2.2 Injection

Injection is an active form of attack that takes place on the physical and link layers. Since Ethernet packets are unauthenticated, attackers can send valid packets with any source, destination, or MAC address information they like. Using this to their advantage, a potential attacker can impersonate anybody on the network. These types of attacks violate network integrity.

2.2.1 Packet injection ARP spoofing

An example of packet injection is called ARP spoofing.

Recall that ARP is the protocol that maps IP addresses to MAC addresses on a local network.

```
$ sudo tcpdump -v -n -i eno1
tcpdump: listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:29:47.455929 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 172.16.15.1
    tell 172.16.15.151, length 46
```

Figure 6: Using tcpdump to scan network interface eno1

If we make an ARP request for the IP shown in Figure 6, we hope to receive a response from the owner with their MAC address. Since ARP requests are broadcast across the entire local network, anyone can listen for them. If an attacker wanted to impersonate somebody on a local network, they could send fake ARP responses in reply to those requests giving their own MAC address. The victim will then list the attacker's MAC address as the destination for the Ethernet frames they wish to send to the legitimate host, and the legitimate host won't see or respond to the victim's traffic because it has been misaddressed. If the attacker impersonates the local router, any packets intended for the router will list the attacker's address, allowing them to read and respond to all the traffic on the network.

2.3 Jamming

Jamming is an attack that prevents signals from reaching their destination. Jamming attacks violate the goal of availability.

Since the network must interact with the physical world to function, it can be subject to jamming. Jamming over a physical medium of the network such as an Ethernet cable can occur when the signal is overwhelmed or disrupted by some outside source. Likewise, radio communications such as WiFi can be jammed by overwhelming the frequencies it is broadcast on.

3 Network threats

3.1 Spoofing

As with Ethernet packets at the link layer, IP is vulnerable to injection at the network layer. The source and destination addresses of IP packets are completely arbitrary and can be set to anything without verification. An attacker can take advantage of this by creating a situation where a phony packet "races" a legitimate packet to respond to a request. This is called spoofing.

An example of an application protocol that is vulnerable to such spoofing is DHCP. If a client sends a packet requesting a DHCP configuration from a local server, an attacker on the same network can send a response forged to look as if it's from the DHCP server. This allows the attacker to trick the client into setting its network gateway and DNS server to hacker-controlled values. The attacker can then resolve hostnames asked for by the client to whatever they want. From here, the attacker can reroute traffic intended for legitimate hosts to the themselves.

3.2 Constructing Arbitrary Network Packets

The way IP works means that anyone can construct an IP packet with essentially any values they want for any fields in the packet header and send it into the internet. In some sense this is an intentional feature of the network layer because IP is not a connection-oriented protocol. We want users to construct network packets with the destinations of the websites or ssh servers they are trying to access. However, this also means that because a recipient isn't granting a sender permission to address traffic to them, the IP protocol doesn't give a sender any way to keep unwanted traffic from being directed towards them.

3.2.1 Network Scanning

One security-relevant implication of the ability to construct arbitrary network packets is network scanning. Network Scanning means looking for services open on a network. These services may or may not be publicly advertised. In order to see if a host is accepting connections on a particular port, a scanner can send a TCP SYN packet to that IP and port number and see if it gets a SYN-ACK in response. A scanner can easily enumerate all 2^{16} port numbers on a single machine. Nmap is a popular tool for doing this.

It is also possible for network scanners to enumerate all 2^{32} possible IPv4 addresses to find un-advertised services. Zmap and Shodan are examples of tools that scan the publicly visible internet for services running on particular ports. Researchers use these tools to understand network usage and patterns. If you've ever run an ssh server, you have probably observed attackers using network scanning to find ssh servers to attempt to brute-force passwords on.

In 2018, an attacker used network scanning to cause 50,000 printers print a message telling them to "Subscribe to PewDiePie" and that their printer was "exposed to the internet" urging them to "please fix that". [1]

3.2.2 Denial of Service

Another implication of a network attacker's ability to send unwanted traffic is a Denial of Service attack. In a Denial of Service attack, the attacker's goal is to prevent the recipient from being able to open or accept new connections by overwhelming them with traffic. Historically, since the TCP protocol is a stateful protocol, an attacker could cause a victim to allocate memory by sending them a single TCP SYN packet. The victim application would then respond with a SYN-ACK packet, and allocate state to remember the current sequence numbers and so on. An attacker could then flood the victim with many SYN packets for new connections (possibly with forged source addresses) until the victim ran out of memory. (SYN cookies are a cryptographic countermeasure to this attack that we will talk about later.)

3.3 Misdirection

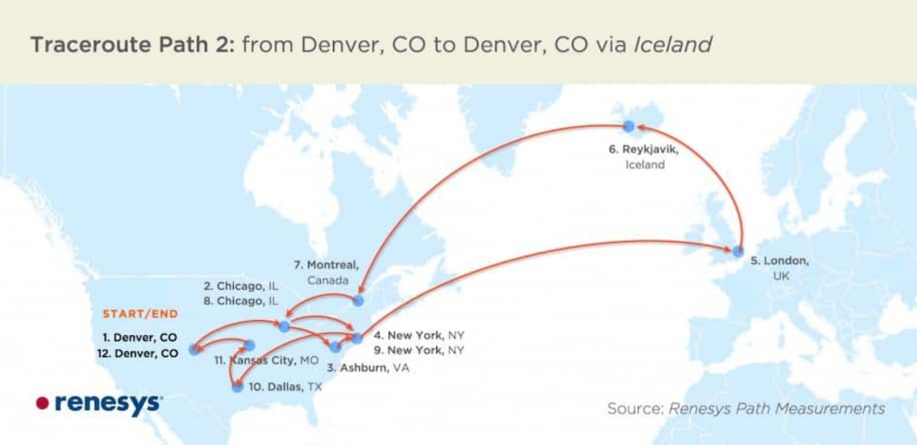
Misdirection means causing network traffic to be sent to the wrong location.

The BGP protocol is unauthenticated, and has permitted a number of such attacks on the real internet. Recall that the BGP protocol is how ASes learn routes to IP prefixes over the internet. ASes will advertise routes to prefixes that they learn about, and each AS will choose the path that is optimal for it given its constraints. Route changes propagate over the network. This is the intended functionality: we want the internet to be able to adapt to changes like parts of the network going down or new routes being added. The weakness of this system is that routes are not authenticated, allowing malicious or incorrect routes to be propagated by a bad BGP node.

A famous instance of BGP hijacking was in 2008, when the government of Pakistan required the national ISP to block a YouTube video. They responded to this request by having their BGP servers route YouTube's IPs to a location that doesn't accept network traffic (a black hole). Because they used BGP to do this, other BGP nodes outside of Pakistan began to propagate this black hole for YouTube. This resulted in all of YouTube's global traffic being routed through Pakistan, effectively crashing Pakistan's internet and making YouTube unusable until the routing information was corrected.

BGP hijacking can be less noticeable as well. Here is an example from a report by Renesys that documented cases of BGP routes sending traffic through unusual paths across the world. The speculation was that these hijacked routes were propagated by parties or governments interested in spying on traffic that they would otherwise not have had access to.

There have also been several documented cases of BGP hijacking attacks used to steal cryptocurrency.



Source: *Renesys Path Measurements*

Figure 7: Traffic from Denver, Colorado to Denver being routed through London and Reykjavik, Iceland [2]

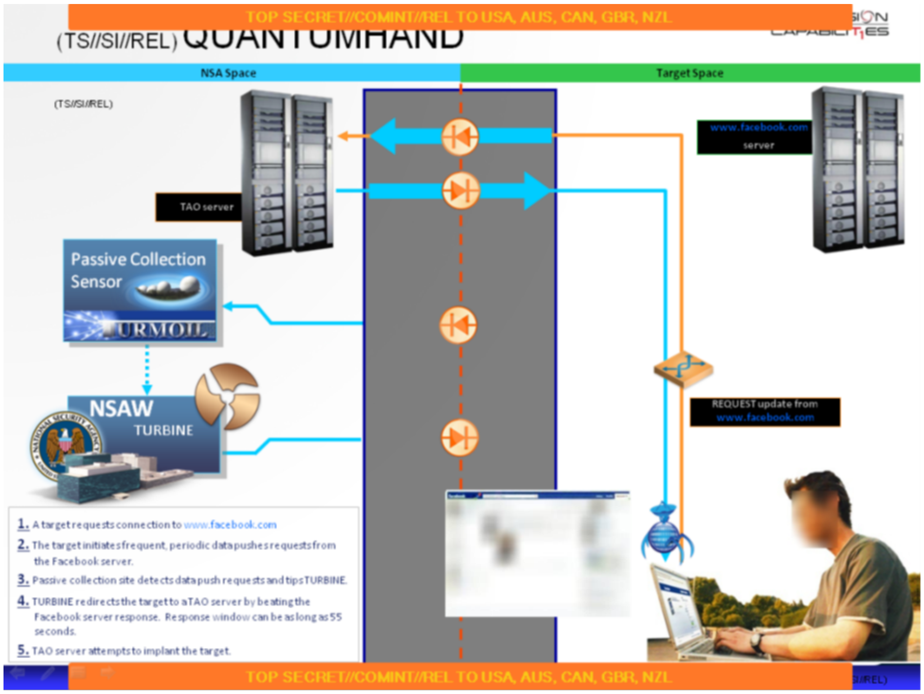
4 TCP threats

Recall that Transmission Control Protocol (TCP) sessions are identified by a source address, source port, destination address, and destination port, and that each TCP packet contains a sequence number that determines where in the stream it belongs. Attackers can exploit these pieces of information to tamper with TCP connections.

One way an attacker can tamper with a TCP connection is called **on-path injection**.

An example of an on-path injection attack is connection hijacking. Connection hijacking is an attack where an on-path attacker injects data into a TCP connection by sending packets with the same port and sequence number as legitimate packets for the TCP session. In this scenario, since the attacker is on-path, the attacker is able to see the port numbers and sequence addresses of the current stream, and can construct their own packets with malicious payloads to inject into the stream with the fully correct values.

Another example is RST injection, where the attacker injects a reset (RST) packet to close the connection. This reset packet will always be accepted by the network if the sequence number is within the acceptable window. China's Great Firewall uses RST injection to block traffic for Chinese Internet users.


TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL
(TS//SI//REL) QUANTUMHAND
NSA Space
Target Space
(TS//SI//REL)
TAO server
www.facebook.com
server
Passive Collection Sensor
TURMOIL
NSAW
TURBINE
REQUEST update from
www.facebook.com
TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

1. A target requests connection to www.facebook.com
2. The target initiates frequent, periodic datapushes requests from the Facebook server.
3. Passive collection site detects datapush requests and tips TURBINE.
4. TURBINE redirects the target to a TAO server by beating the Facebook server response. Response window can be as long as 55 seconds.
5. TAO server attempts to implant the target.

Figure 8: NSA Example of TCP Hijacking Using Facebook

4.1 Great Firewall of China

In order to monitor all cross-border network traffic and block various international services and sites, China has implemented a collection of network policies called the "Great Firewall." The most common technique used by the Great Firewall is RST injection. The Great Firewall infrastructure examines TCP packets and will inject RST packets into connections opened to banned IPs and hosts or when a blacklisted keyword is detected upon deep packet inspection.

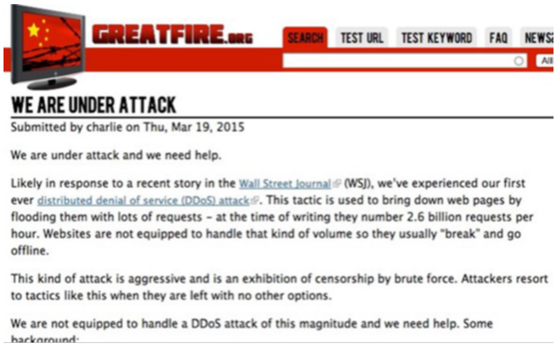
Due to the extensive lengths the Chinese government goes to monitor network traffic, circumventing the Great Firewall has become a large part of the multi-decade arms race on censorship circumvention. Some of the circumvention techniques Chinese internet users take advantage of are HTTPS, VPNs, proxies, traffic obfuscation, domain fronting, and refraction networking.

4.1.1 Example: China's Methods Against Domain Fronting

Domain fronting is a technique used to get around firewalls that takes advantage of HTTPS to connect to a different host than a packet would initially suggest upon inspection. It works by sending a packet to an IP address that accepts TLS connections for multiple domain names. The outer, unencrypted "handshake" portions of the connection appear to be requesting a connection to a non-blocked site, but then the inner encrypted HTTP request will include a GET request for a blocked site.

A famous incident of domain fronting occurred in 2015 when GreatFire.org was targeted by the Great Firewall of China for using domain fronting on Amazon CloudFront and GitHub.

China was unable to block GitHub without causing problems for their technology industry, so they wanted to force CloudFront and GitHub to take down GreatFire.org's content themselves. To do this, the Chinese government adopted a new active technique to mount a Distributed Denial of Service (DDoS) attack against GitHub and CloudFront. This was called the "Great Cannon". In this attack, the Great Firewall used TCP injection to inject a JavaScript payload into requests for Baidu, a Chinese search engine, that would cause a fraction of Baidu visitors from outside China to send unwanted traffic to GitHub. This DDoS attack continued for five days before they stopped.



WE ARE UNDER ATTACK

Submitted by charlie on Thu, Mar 19, 2015

We are under attack and we need help.

Likely in response to a recent story in the Wall Street Journal (WSJ), we've experienced our first ever distributed denial of service (DDoS) attack. This tactic is used to bring down web pages by flooding them with lots of requests – at the time of writing they number 2.6 billion requests per hour. Websites are not equipped to handle that kind of volume so they usually "break" and go offline.

This kind of attack is aggressive and is an exhibition of censorship by brute force. Attackers resort to tactics like this when they are left with no other options.

We are not equipped to handle a DDoS attack of this magnitude and we need help. Some background:

Figure 9: GreatFire.org Domain Fronting Incident

4.2 Blind Spoofing

Blind spoofing is when an attacker convinces a victim to open a TCP connection with another host. Blind spoofing is an example of an off-path attack where the attacker does not see the victim's network traffic.

The interesting challenge in this attack is how the attacker can convince the victim to open a TCP connection with the spoofed host in the first place. In order to establish the connection, the attacker sends the initial TCP handshake package with a forged IP address to the victim. Because the attacker does not have access to the victim's network traffic, they will not be able to see the SYN-ACK response with the victim's initial sequence number. If the attacker doesn't know the victim's sequence number, they won't be able to send the proper ACK response to the victim in order to finalize the connection.

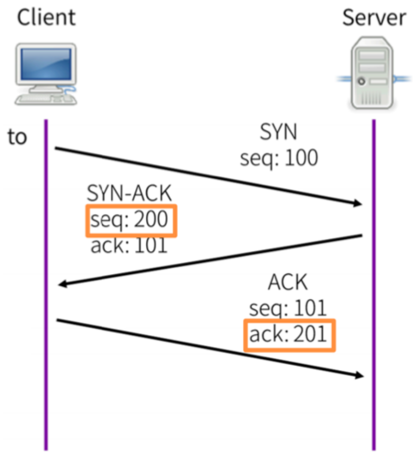


Figure 10: Process of opening a TCP connection

In the original TCP specification, the sequence number could be generated from the current time. The attacker then had a reasonable chance of being able to guess the victim's sequence number.

The mitigation for this is to randomly generate an initial sequence number (ISN). This makes it harder for an attacker to guess the sequence number: with a 32-bit value, a random guess would have probability 2^{-32} of success.

(Off-path TCP injection for an existing connection can have a higher chance of success because implementations will accept packets with a sequence number within a window of the current sequence number.)

5 Application Layer Threats

5.1 DNS Spoofing

Recall that DNS is used to map domain names to IP addresses. To do their job quickly, DNS responses are cached to improve query times. DNS threat models include malicious DNS servers, local/on-path attackers, and off-path attackers.

5.1.1 Malicious DNS Server

Any DNS server in the query chain can lie about responses. This can happen if the attacker managed to appear as a valid DNS server to local networks using DHCP spoofing or ARP spoofing at the data link layer. If the attacker is successful, they can map the victim's request to any IP address they want.

5.1.2 Local/On-Path Attacker

Recall that an on-path attacker can see and add packets, but cannot block packets. The original DNS specification is not authenticated by default. This gives on-path attackers an opportunity to impersonate a DNS server and send fake responses.

5.1.3 Off-Path Attacker

An attacker can try to forge response to a DNS server by matching the 16-bit query ID. If successful, this would allow an off-path attacker to have DNS servers resolve hostnames to whatever IP they wanted.

Originally this was easy because the query ID increments with each request. This means the attacker could simply look at the previous query and predict the next query ID. In response to this, the specification was changed so that the query ID is generated in a more random way. Even with random query IDs though, DNS servers are still vulnerable as Dan Kaminsky has demonstrated with what is called the Kaminsky Attack.

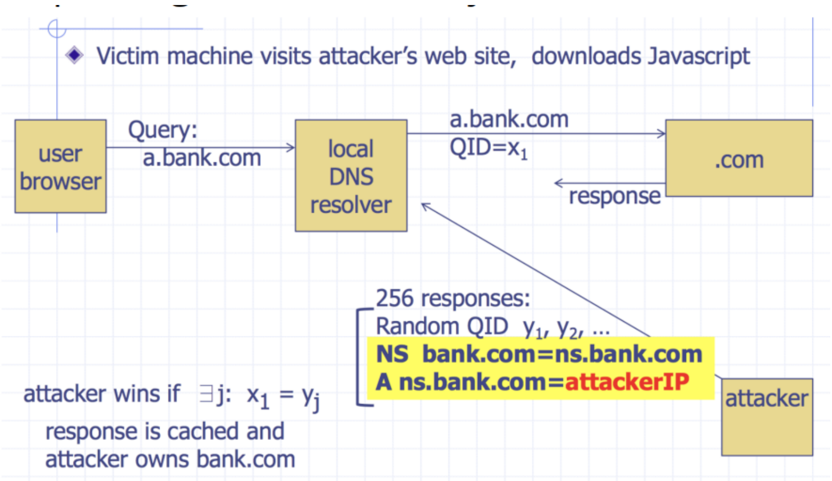
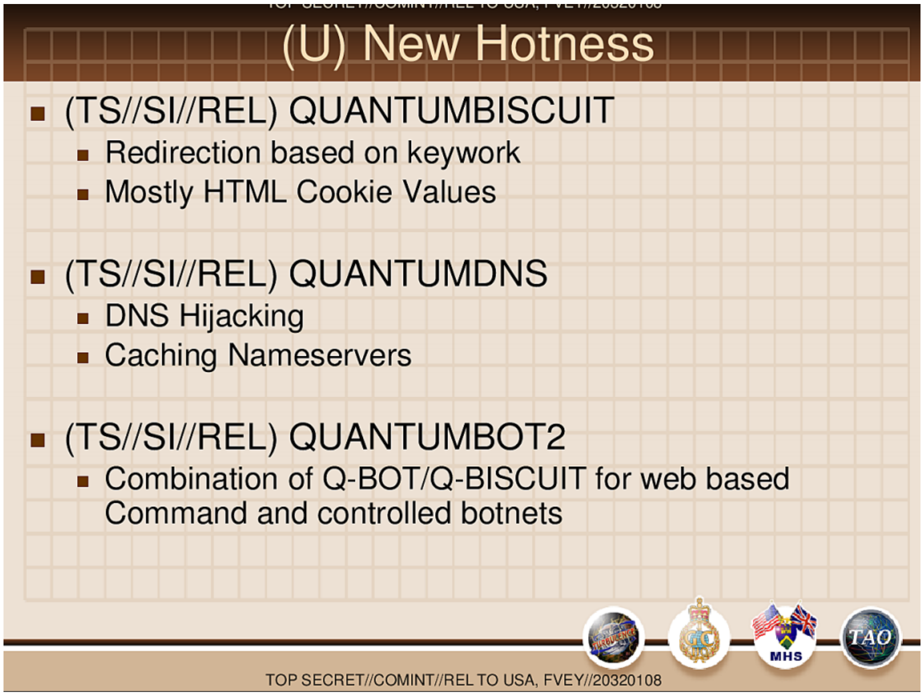


Figure 11: Example of DNS spoofing: 2008 Kaminsky attack

The victim first visits the attacker's website where they download a malicious JavaScript script. The script causes the victim to make a bunch of DNS look up queries for various subdomains of the bank's website. If the user browser makes at least 256 queries and the attacker simultaneous makes at least 256 responses with a random query ID to the local DNS server, there is a good chance that one of the query IDs will match. This works because if the victim is relying on $2^{16} = 65536$ possible random values and an attacker is sending values at the rate of the $\sqrt{2^{16}} = 2^8 = 256$ the attacker has a 50 percent chance of success by the Birthday bound.

One way to mitigate this is to add another 16 bits of randomness by randomizing the source port of the DNS query from the user.



TOP SECRET//COMINT//REL TO USA, FVEY//20320108

(U) New Hotness

- (TS//SI//REL) QUANTUMBISCUIT
  - Redirection based on keywork
  - Mostly HTML Cookie Values
- (TS//SI//REL) QUANTUMDNS
  - DNS Hijacking
  - Caching Nameservers
- (TS//SI//REL) QUANTUMBOT2
  - Combination of Q-BOT/Q-BISCUIT for web based Command and controlled botnets

TOP SECRET//COMINT//REL TO USA, FVEY//20320108

Figure 12: DNS hijacking with the Kaminsky attack

Figure 12 outlines the process of DNS hijacking with the Kaminsky attack. Defenses against this attack include doing DNS queries inside of encrypted protocols (TLS or HTTPS) to provide authentication and privacy for queries.

6 Conclusion

The Internet was build built on top of protocols that assumed trustworthy network operators. This has allowed a number of clever attacks abusing the protocols in ways that were never expected by their designers, and which have caused problems for decades. There are countermeasures in place against many of the attacks we have discussed. The most effective ones use cryptography.

References

[1] Patricia Hernandez. Someone hacked printers worldwide, urging people to subscribe to pewdiepie. *The Verge*, Nov 2018.

[2] Andrea Peterson. Researchers say u.s. internet traffic was re-routed through belarus. that's a problem. *The Washinton Post*, Nov 2013.