Computer Security

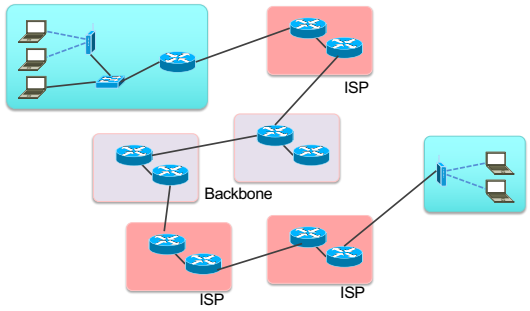
11. Network Security

Paul Krzyzanowski
Rutgers University
Fall 2019

1

The Internet

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



2

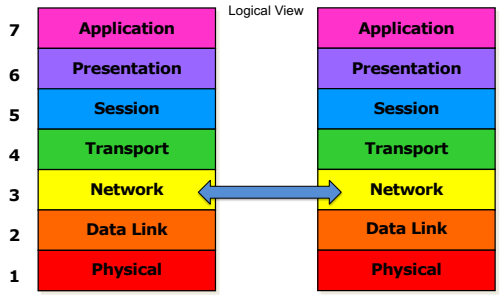
The Internet: Key Design Principles

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

3

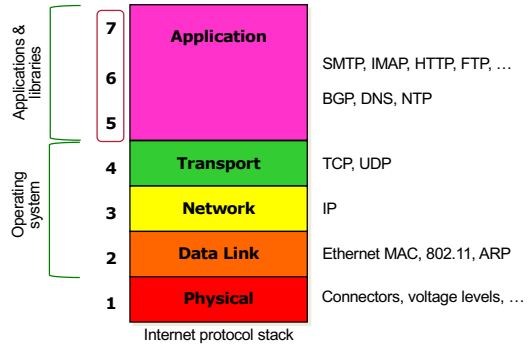
Network protocol layers

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



4

IP Protocol Stack



5

Data Link Layer

6

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

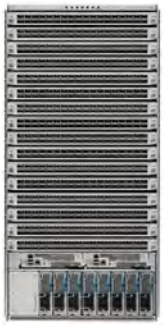
7

Link Layer: CAM overflow

Monitor all traffic on a LAN

8

Layer 2: Ethernet Switches



Cisco Nexus 9516 Switch
- 1/10/40 GbE
- 21-rack-unit chassis
- Up to 576 1/10 Gb ports

TP-Link Switch
- 8 1-GbE ports

9

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

10

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

11

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

12

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
  - *macof* tool: ~100 lines of perl
- 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

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

13

Link Layer: VLANs & VLAN hopping

Join VLANs you are not a member of

14

VLANs

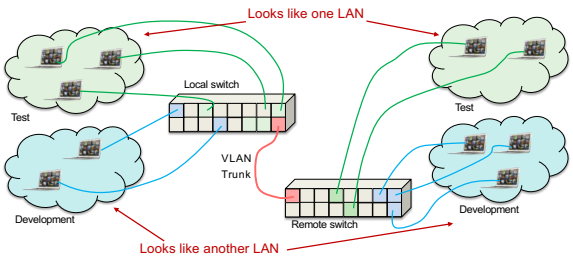
- A switch & cables creates a local area network (LAN)
- We use LANs to
  - Isolate broadcast traffic from other groups of systems
  - Isolate users into groups
  - What if users move? What if switches are inefficiently used?
- Virtual Local Area Networks (VLANs)
  - Create multiple virtual LANs over one physical switch infrastructure
  - Network manager can assign a switch's ports to a specific VLAN
  - Each VLAN is a separate broadcast domain

15

VLAN Trunking

VLANs across multiple locations/switches

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



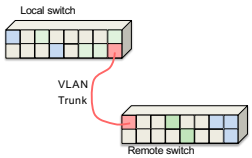
16

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



17

Avoiding VLAN Hopping

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

18

ARP Cache Poisoning (ARP Spoofing)

Intercept traffic for other IP addresses

19

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?

20

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

ARP packet structure

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

21

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

22

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

23

DHCP Server Spoofing

Configure hosts with your chosen network settings

24

DHCP (Dynamic Host Configuration Protocol)

Computer joins a network – needs to be configured
- *Broadcasts* a *DHCP Discover* message

A DHCP server picks up this requests 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

25

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

26

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

27

Network Layer (IP) vulnerabilities

28

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

29

Source IP address

No source IP address authentication
- Clients are *supposed* to use their own source IP address
  - Can override with raw sockets
  - Error 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

30

Attacks on routers

- Routers are just special-purpose computers
  - People may keep default passwords or not use strong passwords
  - Router OS may not be kept updated
- 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

31

Transport Layer (UDP, TCP) vulnerabilities

32

TCP & UDP

UDP: User Datagram Protocol

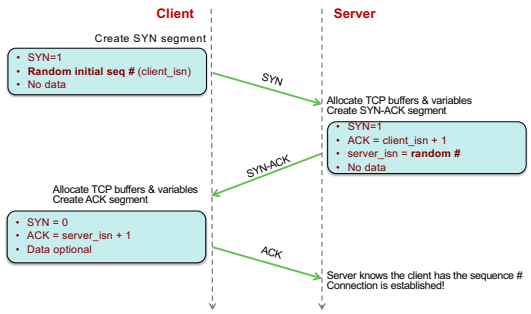
- Stateless, connectionless & unreliable
- Anyone can send forged UDP messages

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

33

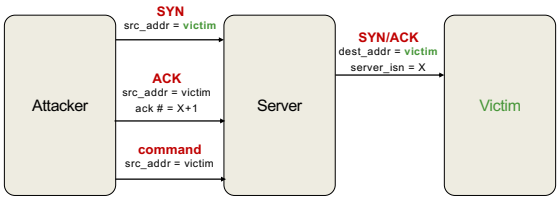
TCP connection setup: three-way handshake



34

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

35

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

36

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

37

Network Routing Protocols

38

Routing protocols

OSPF: Open Shortest Path First

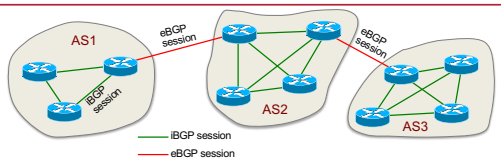
- Interior Gateway Protocol (IGP) within an autonomous system (AS)
- Uses a **link state routing algorithm** (Dijkstra's shortest path)

BGP: Border Gateway Protocol

- Exterior Gateway Protocol (EGP) between autonomous systems (AS)
- Exchanges routing and reachability information
- **Distance vector routing protocol**

39

BGP sessions maintained via TCP links



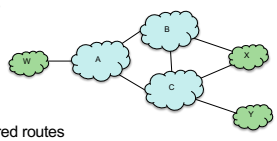
Pairs of routers exchange information via semi-permanent TCP connections

- 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

40

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

41

BGP Hijacking

- **Route advertisements are not authenticated**
  - Anyone can inject advertisements for arbitrary routes
  - Information will propagate throughout the Internet
  - Can be used for DoS or eavesdropping

(Partial) Solutions

See RFC 6480

- **RPKI** (Resource Public Key Infrastructure) framework
  - Each AS obtains an X.509 certificate from the Regional Internet Registry (RIR)
  - AS admin creates a Route Origin Authorization (ROA)
  - ROA is signed by the AS's private key
  - Advertisements without a valid, signed ROA are ignored

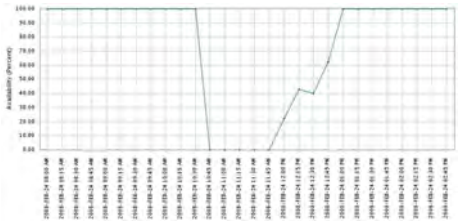
See RFC 8206

- **BGPsec**
  - Integral part of BGP protocol
  - Each hop in the AS path is protected with a signature

42

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



43

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 finally fixed after 2 hours

44

The Washington Post

Internet traffic hijack disrupt Google services

By Frank Bajak | AP November 13, 2018

An internet traffic diversion rerouted data through Russia and China and disrupted Google services on Monday, including search, cloud-hosting services and its bundle of collaboration tools for businesses.

Service interruptions lasted for nearly one and a half hours and ended about 5:30 p.m. EST., network service companies said. In addition to Russian and Chinese telecommunications companies, a Nigerian internet provider was also involved.

The diversion "at a minimum caused a massive denial of service to G Suite (business collaboration tools) and Google Search" and "put valuable Google traffic in the hands of ISPs in (internet service providers) in countries with a long history of Internet surveillance," the network-intelligence company ThousandEyes said in a blog post.

45

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/

46

BORDER GATEWAY PROTOCOL —

ars TECHNICA

Strange snafu misroutes domestic US Internet traffic through China Telecom

Telecom with ties to China's government misdirected traffic for two and a half years.

DAN GOODIN - 11/6/2018, 9:05 AM

China Telecom, the large international communications carrier with close ties to the Chinese government, misdirected big chunks of Internet traffic through a roundabout path that threatened the security and integrity of data passing between various providers' backbones for two and a half years, a security expert said Monday. It remained unclear if the highly circuitous paths were intentional hijackings of the Internet's Border Gateway Protocol or were caused by accidental mishandling.

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

47

Domain Name System (DNS) Vulnerabilities

48

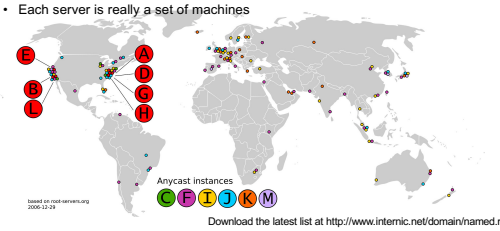
Domain Name System

- 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 regsirty:** Maintains the top-level domain database
- *So how do you find the right DNS server?*
  - Start at the root

49

Root name servers

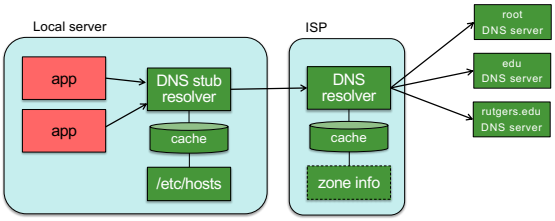
- The **root name servers** provide lists of authoritative name servers for top-level domains
- 13 root name servers
  - `A.ROOT-SERVERS.NET`, `B.ROOT-SERVERS.NET`, …
  - Each has redundancy (via *anycast* routing or load balancing)
    - Each server is really a set of machines



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

50

DNS Resolvers in action



Local stub resolver:
- check local cache
- check local hosts file
- send request to external resolver

External resolver:
- Running at ISP, Cloudflare, Google Public DNS, OpenDNS, etc.

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

51

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

52

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

53

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 does 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 numbers were sequential and not hard to guess
  - Fix by using random Query IDs

54

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)

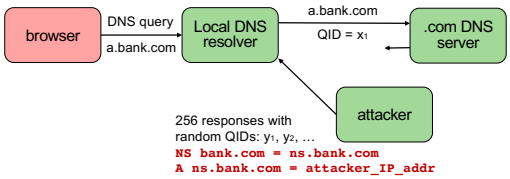
If the attacker is successful, one of the responses matches up

- 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!

55

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.

56

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)

57

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 provide authenticated requests & 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

58

DNS Rebinding

59

DNS Rebinding

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

- What is the **same-origin policy**?
  - The core web application security model
  - Client web browser scripts can access data from other web pages ***only*** if they have the same **origin**
  - Origin = same { protocol, host name, port number }
- The policy relies on **comparing domain names**
- If we can change the underlying address:
  - We can send messages to an attacker's system while the software 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, …

60

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 domain
    - Permitted under **same-origin policy**
      - A browser permits scripts in one page to access data in another only if both pages have the same origin & protocol
    - The script causes the browser to issue a new DNS request
    - Attacker replies with a new IP address (e.g., a target somewhere in the victim's LAN)
    - The script can continue to access content at the same domain
      - But it really isn't in the domain!

61

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

62

Network Layer Conversation Isolation: Virtual Private Networks (VPNs)

63

Fundamental Layer 2 & 3 Problems

- IP relies on store-and-forward networking
  - Network data passes through untrusted hosts
  - Routes may be altered to pass data through malicious hosts
- Packets can be sniffed (and new forged packets injected)
- Ethernet, IP, TCP & UDP
  - All designed with no authentication or integrity mechanisms
  - No source authentication on IP packets
  - TCP session state can be examined or guessed …
    … and TCP sessions can be hijacked
- ARP, DHCP, DNS protocols
  - Can be spoofed to redirect traffic to malicious hosts
- Internet route advertisement protocols are not secure
  - Can redirect traffic to malicious routers/hosts

64

Solution: Use private networks

Connect multiple geographically-separated private subnetworks together

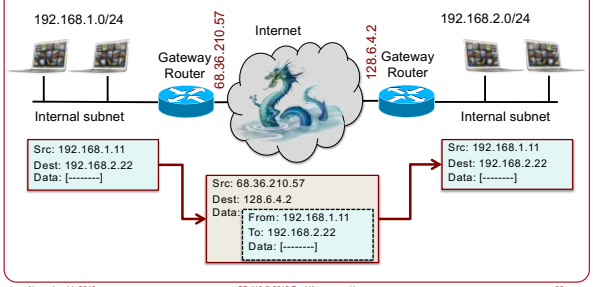
But this is expensive … and not feasible in many cases (e.g., cloud servers)

65

What's a tunnel?

Tunnel = Packet encapsulation

Treat an entire IP datagram as payload on the public network



66

Tunnel mode vs. transport mode

- **Tunnel mode**
  - Communication between gateways: *network-to-network*
  - Or *host-to-network*
  - Entire datagram is encapsulated
- **Transport mode**
  - Communication between hosts
  - IP header is not modified

67

Virtual Private Networks

Take the concept of tunneling

… and safeguard the encapsulated data

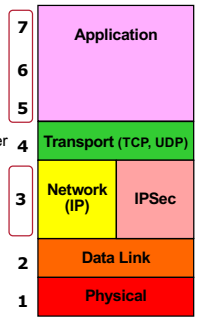
- **Add a MAC**
  - Ensure that outsiders don't modify the data
- **Encrypt the contents**
  - Ensure that outsiders can't read the data

68

IPsec

Internet Protocol Security

- End-to-end solution at the IP layer
- Two protocols:
  - **IP Authentication Header** Protocol (AH)
    - Authentication & integrity of payload and header
    - ***Provides integrity***
  - **Encapsulating Security Payload** (ESP)
    - AH + Confidentiality of payload
    - ***Adds content encryption***

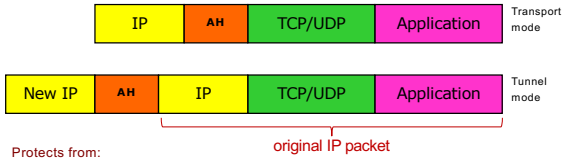


69

IPsec Authentication Header (AH)

Guarantees integrity & authenticity of IP packets

- MAC for the contents of the entire IP packet
- Over unchangeable IP datagram fields (e.g., not TTL or fragmentation fields)



Protects from:

- Tampering
- Forging addresses
- Replay attacks (signed sequence number in AH)

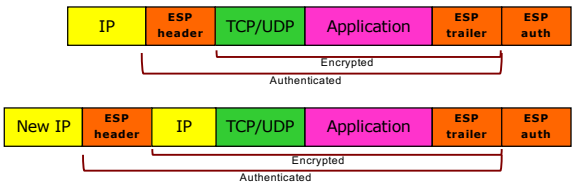
Layered directly on top of IP (protocol 51) - not UDP or TCP

70

IPsec Encapsulating Security Payload (ESP)

Encrypts entire payload

- Plus authentication of payload + IP header (everything AH does) (may be optionally disabled – but you don't want to)



Directly on top of IP (protocol 51) - not UDP or TCP

71

IPSec algorithms

- **Authentication**
  - Certificates, or pre-shared key authentication
- **Key exchange**
  - Diffie-Hellman to exchange keying material for key generation
  - Key lifetimes determine when new keys are regenerated
- **Confidentiality**
  - 3DES-CBC
  - AES-CBC
- **Integrity protection & authenticity**
  - HMAC-SHA1
  - HMAC-SHA2

72

Transport Layer Conversation Isolation: Transport Layer Security (TLS)

73

Transport Layer Security

- Goal: provide a *transport layer* security protocol
- After setup, applications feel like they are using TCP sockets

SSL: Secure Socket Layer

- Created with HTTP in mind
  - Web sessions should be secure
  - Mutual authentication is usually not needed
    - Client needs to identify the server but the server won't know all clients
    - Rely on password authentication after the secure channel is set up

75

TLS vs. SSL – versions

SSL evolved to TLS (Transport Layer Security)

SSL 3.0 was the last version of SSL
… and is considered insecure

We now use TLS (but is often still called SSL)

- TLS 1.0 = SSL 3.1, TLS 1.1 = SSL 3.2, TLS 1.2 = SSL 3.3
- Latest version = TLS 1.3 = SSL 3.4

- Retired versions
  - TLS 1.0/SSL 3 are not considered strong anymore and their use is not recommended
  - As of 2019, Google Chrome deprecates support for TLS 1.1

76

TLS Protocol

Goal:

Provide authentication (usually one-way), privacy, & data integrity between two applications

- Principles
  - **Data encryption**
    - Use symmetric cryptography to encrypt data
    - **Key exchange**: keys generated uniquely at the start of each session
  - **Data integrity**
    - Include a MAC with transmitted data to ensure message integrity
  - **Authentication**
    - Use public key cryptography & X.509 certificates for authentication
    - Optional – can authenticate 0, 1, or both parties
  - **Interoperability & evolution**
    - Support many different key exchange, encryption, integrity, & authentication protocols – negotiate what to use at the start of a session

78

TLS Protocol & Ciphers

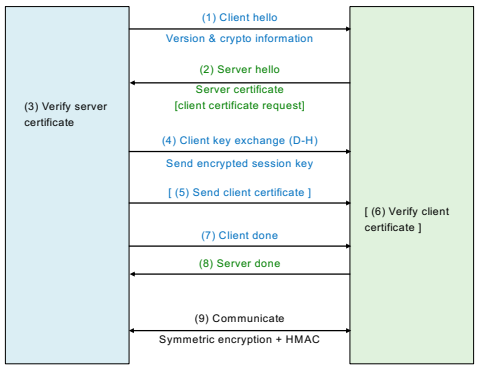
Two sub-protocols

1. Authenticate & establish keys
2. Communicate
   - HMAC used for message authentication

- Authentication
  - Public keys (X.509 certificates and – usually – RSA cryptography)
- Key exchange options
  - Ephemeral Diffie-Hellman keys (generated for each session)
  - Pre-shared key
- Data encryption options
  - AES GCM, AES CBC, ARIA (GCM/CBC), ChaCha20-Poly1305, …
- Data integrity options
  - HMAC-SHA1, HMAC-SHA256/384, …

79

TLS Protocol



80

Benefits of TLS

- Protects integrity of communications
- Protects the privacy of communications
- Validates the authenticity of the server (if you trust the CA)

81

Some attacks on TLS

- Man-in-the-middle: BEAST attack in TLS 1.0
  - Attacker was able to see Initialization Vector (IV) for CBC and deduce plaintext (because of known HTML headers & cookies)
  - An IV doesn't have to be secret – but it turned out this wasn't a good idea
  - Attacker was able to send chosen plaintext & get it encrypted with a known IV
  - Fixed by using fresh IVs for each new 16K block
- Man-in-the-middle: crypto renegotiation
  - Attacker can renegotiate the handshake protocol during the session to disable encryption
  - Proposed fix: have client & server verify info about previous handshakes
- THC-SSL-DoS attack
  - Attacker initiates a TLS handshake & requests a renegotiation of the encryption key – repeat over & over, using up server resources

82

TLS: Client Authentication Problem

Client authentication is almost never used

- Generating keys & obtaining certificates is not an easy process for users
- Any site can request the certificate
  - User will be unaware their anonymity is lost
- Moving private keys around can be difficult
  - What about public computers?

We usually rely on other authentication mechanisms

- Usually user name and password
- But no danger of eavesdropping since the session is encrypted
- May use one-time passwords or two-factor authentication if worried about eavesdroppers at physical premises

83

The end

84