Lesson 6 - Message Integrity, PKI, and TLS

Diffie Hellman - allows two entities to agree on a shared key
n is a large prime; g is a number les than n, that are made public.
Alice:
a, g, n
A = g^a mod n
Alice sends g, n, A to Bob.
Bob:
b
B = g^b mod n
Bob sends B to Alice.
Alice: K = B^a mod n
Bob: K = A^b mod n
Trudy is in the middle, and she won't be able to determine what a and b are, but Trudy can instead pretend to be Bob and Alice and form her own keys with both Alice and Bob at the same time. Alice will think she's talking to Bob, Bob will think he's talking to Alice, and both will be talking to Trudy. Diffie Hellman is susceptible to active AITM, and needs to be protected with RSA/certificates that provides authentication. That way, Alice and Bob know they are talking to each other. Diffie-Hellman cannot be used by itself.

Message Integrity
Encryption: Taking something and transforming it so it is unrecognizable without the key - it can be decrypted back to it's original form with the key.
Hash: Taking a message of any size and creating a fixed size string - message digest.
Changing any character will completely change the outcome.
Hashing functions: MD5 (broken), Sha-1 (also broken) - still sometimes used, but not for security.
Collision: When 2 different inputs produce the same hash value with the same hash function. They should always be different.

Security Levels of Crytpo Algorithms

Security Level	Work Factor	Algorithms
Weak	O(2^40)	DES, MD5
Legacy	O(2^64)	RC4, Sha1
Minimum	O(2^80)	3DES, SEAL, SKIPJACK, RSA-1024, DH-1024

Security Level	Work Factor	Algorithms
Standard	O(2^128)	AES-128, SHA-256, RSA-2048, DH-2048
High	O(2^192)	AES-192*, SHA-284
Ultra	O(2^256)	AES-256, SHA-512, RSA-4096, DH-4096

*AES-192 not used in practice

Digital Signature: Requires public key algorithms (public private key pair) - it is a hash of a file that was encrypted (the hash, not the file itself) using the private key.
To verify a digital signature, make sure that it is correct, the recipient decrypts it with the public key and validate the hash to make sure it matches the hash of the message.

"Hello this is my secret message"	Hash of message

This becomes a digital signature
The recipient takes the digital signature, decrypts it using public key, and gets a hash of the message. In email, the public key is sent with the email. In email, there is no easy and no good way to trust public keys. You need to manually confirm that the email comes from who it says it comes from.

- In a corporate environment using the same CA (e.g., everyone on Outlook at the same company), the CA issues certificates for users.
- Outside of that, when sending a digitally signed email to someone at another company, the sender’s public key is included with the email itself.
- This key contains identifying information like the sender’s email address.
- Verification usually involves manually confirming the key’s authenticity—e.g., calling the sender to check key details—before trusting it.
- There’s no universal CA lookup for these public keys in personal or cross-company email.
- The process for secure email exchange involves first sending a signed message so the recipient can get and verify your public key, and then both sides can exchange encrypted emails.
- The weakness is that the public key comes in the same email as the signature, so without independent verification, it could be spoofed.

Digital Signatures and HMAC are used for similar things, but there are some differences.

Digital Signatures	HMAC
Uses public keys for hashing	Uses symmetric keys

Digital Signatures	HMAC
Keys do not need to be shared in advance	Keys must be pre-shared
Not as fast as HMAC	Extremely fast
More used for human based discussions	Things that require extreme speed - for automated processes, computer to computer discussions

Playback Attacks:
When Alice and Bob are talking, Trudy intercepts the message Alice sends to Bob and sends it again to Bob - Bob doesn't know that the message did not come from Alice.
Solution: Nonce - number used only once. Alice initiates a conversation with Bob, Bob validates digital signature/HMAc and sends Alice a nonce to be used. Alice sends the message back to Bob with the nonce, and Bob validates that nonce was not used before. If it wasn't, then the message is good.

Public Key Infrastructure (PKI)
RSA, Diffie-Hellman, AES serve as backbone of modern cryptography - variations exist.
Public Key Algorithm: RSA
Key Sharing: Diffie-Hellman
Bulk encryption: AES

PKI is an infrastructure for numerous services that work together to enable interoperability of digital certificates
More of a framework
When you go to a website, you use TLS, which uses PKI and it enables interoperability across websites.

Reminder: > A digital signature is created by first hashing the message, and then **encrypting the hash with the sender's private key**.
Validating a digital signature: the recipient decrypts the signature using the public key and validate the hash
A public key is issued by a Certificate Authority - because the key is issued, the recipient has a way to validate the signature
Digital Signatures vs HMAC:
Digital Signatures use public keys
HMAC uses symmetric keys, and are faster
Nonce - Number used only once

Certificate Authorities
On a website, clicking on a lock logo will bring up the certificate.

TLS_AES_128_GCM_SHA256: GCM is mode of operation, the only one that is secure right now.
TLS v 1.3 - RSA or Diffie-Hellman, or equivalent of them, are always required. That is the reason it is not mentioned in ciphersuite.
Certificate:
Common Name: what the certificate is for (nordstrom.com for example)
Issuer: Who issued the certificate, the CA
Subject Alt Names: The names the certificate is good for: ex: nordstrom.com, *.about.nordstrom.com

A certificate can be valid for multiple domain names - if the subject alternative name for multiple domains
The purpose of a certificate is to hold the public key
Basic Constraints: says whether the certificate is a CA or not

PKI Certificate Authority:
The Root CA is typically loaded into a browser. Intermediary CA is on the webserver and cached by browser. SSL Certificate is sent with website.
Root CA is so important that it is typically offline.

Because browser trusts CA, it will trust certificates issued by CA

SSL/TLS
Transport Layer Security - a protocol built on top of PKI (used for everything)
TLS has been in use for 30 years - it has withstood the test of time.

Types of Certificates:
Domain Validation - ensures ownership of the domain name
Organization Validation - ensures ownership of a company with that domain (never really used)
Extended Validation - most secure, validate business documents. Name must match, there needs to be a call with CEO of company, etc.
These mean how much checking the CA does to make sure it is a proper domain?

Ciphersuite Examples When keys are exchanged:
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
RSA(n,e): step 3 (RSA Public keys)
RSA(n,d): never exchanged RSA Private keys
DH(A,B,g,n,a,b): A,g,n: step 4, B: step 6, a, b: never exchanged, discarded after the session
Key K: "pre-master secret" key generated on step 6 from DH values. PMS + client_random + server_random -> Master Secret, where we get the AES/HMAC keys
This has perfect forward secrecy - if the private RSA key is compromised, all past messages are still protected because nothing is encrypted with that the private key.

TLS_RSA_WITH_AES_128_CBC_SHA256
RSA(n, e): step 3
RSA(n,d): never
DH(A,B,g,n,a,b): not used in this ciphersuite
Key K: "pre-master secret": step 6, encrypted with RSA public key.
From the PMS, they generate the Master Secret, and from the master secret the AES/HMAC key is created.
This does not have perfect forward secrecy - if the RSA private keys are compromised, then all past messages are compromised - Trudy will be able to decrypt Key K.

TLS 1.2 vs TLS 1.3:
Approximately 20-25% of TLS connections use TLS v1.3.
Client Side: starts the ephemeral key exchange, when it sends over supported ciphersuites, it sends over all possible keys.
Server side chooses which ciphersuite and select which keys to use. It sends over certificates, the signature.
Client then sends over the finished message and the HTTP GET request.
Because TLS 1.3 isn't supported by appliances, TLS 1.3 is only supported by extensions.
Inspect TLS headers, it will only say it supports TLS 1.3 in extensions.

TLS is secure because it has been extensively studied over the last few decades. There are some vulnerabilities discovered -authentication issues, issues with infrastructure related to checking certificates. Key exchange issues - web browsers check to make sure keys are strong enough.
TLS vulnerabilities:
Physical attacks (private keys can be stolen), does not protect against compromised hosts, improper validation of certificate fields
Adversary-in-the-Middle Attacks:
A fake root CA can be installed on a target computer - if this happens, attackers can masquerade as any website with a TLS certificate.

Lesson 7 - Firewalls

Firewall: enforces security between networks of different security policies.
Firewalls can be configured so that every area of a network can be firewalled off from others.
Firewalls can enable microsegmentation - CIsco, Juniper, all have firewalls that can be segmented.
Prior to modern day firewalls, different projects/servers/networks would be placed on different VLANs.

Packet Filters (traditional): fast, stateless firewalls, can handle massive DDoS attacks.
Stateful Firewalls: maintain/remembers connection info, can handle more complex traffic than

packet filters
Proxies(Application Gateway): hosts that sit between servers and network, serves as an intermediary on a network - web, email, FTP
Bastion Host: host intended to withstand attacks. Don't want it to be compromised. It is a high-priority target, running very few services, and is hardened.
DMZ: Network architecture that puts outward forward-facing services in a special area of the network.

Stateful Firewalls: Kep track of connection states, may timeout entries which see no activity for set period of time. If rule table indicates that stateful table must be checked, check to see if there is already a connection in stateful table.

Protocol	State	source addr	dst addr	src port	dst pot
tcp	SYN sent	222.22.1.7	37.96.87.123	126999	80

Application Gateways: something in between the user and the server, to inspect traffic and improve usage/performance.
Example: When going to amazon.com, the web proxy is in the middle and takes the connection, recreates it, and goes to the real server. It can perform malware scans, caching, heuristic analysis, DLP policies.
For TLS, and the TLS the proxy is also a CA - a proxy CA.
Chaining Proxies: not usually done due to performance issues, can be done for email due to latency not really being a big concern.

Some protocols for the intended purpose of carrying another protocol.
SOCKS Proxy Protocol: Generic proxy protocol, doesn't have to redo all the code when proxifying an application.
Can be used by HTTP, FTP, telnet, SSL
Independent of application layer protocol
Includes authentication

DMZ: a network architecture philosophy/style in which some important forward facing servers are placed in special area of network that is neither internal nor external.

Internal Network -> Firewall -> router -> Firewall
|
V
DMZ
Typically, each server gets its own area (Web, FTP, Email)

iptables - specific firewall, interface to the Linux firewall netfilter. Can do a lot - filter - allow, deny, drop, reject, etc. Can also do NAT and PAT (Network and Port Address Translation), manipulate packet.
Filter Table has 3 chains: INPUT, OUTPUT, FORWARD
INPUT: Is it coming to the firewall? The firewall is the destination
OUTPUT: Is it coming out of the firewall? The firewall is the source
FORWARD: Is it coming through the firewall? The firewall is neither source nor destination
Input and Output are for host based - FORWARD is for network based firewalls.

iptables cheat sheet
-A, --append
-i, --in-interface: interface in which packet is received
-j ACCEPT, DENY
iptables -L -v: see the rules
-F: flush tables
-I INPUT 1: insert rule into Input table position 1
-p protocol (tcp, udp, icmp)
--tcp-flags SYN
-s: source IP (can also be in CIDR notation)
--sport: source port
-d: dest IP
--dport: destination port

Default Policy: when iptables goes through rules and does not find a matching rule, it goes to the default policy.
For this class: should drop everything (IMPORTANT, REMEMBER)
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

Reject vs Drop:
Reject: "Nice" method: sends an ICMP unreachable
DROP: "Conventional": doesn't respond, discards packet

Connection tracking
-m conntrack --ctstate
NEW,ESTABLISHED,RELATED (for protocols that use multiple ports like FTP)

Lesson 8 - Layer 2 Security

Layer 2 refers to Ethernet and WiFi. It pertains to how you connect to the network - plug in, connect to WiFi.

In OSI or TCP/IP, different layers work together without knowledge of each other. An attack on Layer 2 is an attack on other layers as well.

CAM Table vs ARP Table
ARP Table: Maps IP address to MAC addresses, Layer 3 and Layer 2.
CAM Table: physical ports to MAC, mapping between Layer 2 and Layer 1. Which MAC address relates to which physical port it comes from. CAM Tables are only on physical devices. TCAM memory is extremely fast and expensive physical memory - a switch is able to route extremely fast, so it needs a CAM table to route very quickly.
A CAM Table has limited space because it is physical memory.

MAC	Port
A	1
C	3

MAC A sends a message to port 3 (Switch) destined for MAC B. Port 3 doesn't know where B is, so it sends out a broadcast message. B sends back the response, and it updates the CAM table. After that, traffic between A and B will be private.

MAC	Port
A	1
B	2
C	3

Let's say C is an attacker. A CAM overflow attack (or table flooding) is sending out a lot of messages to the switch. The messages will have bogus MAC addresses and will cause the switch to dump the valid addresses it has in the CAM table to make room for the bogus information. After it happens, the switch will default to broadcast private messages, including to the attacker. This attack does not work on virtual routers or virtual switches - that memory can be unlimited. It only works for physical features.
Countermeasures: Port Security
Limits the Amount of MACs on an interface. Port security limits MAC flooding and locks down port and sends an SNMP trap. It can be configured so that only one MAC address can come from a specific port. If it sees a second MAC address, it will shut down the port.

VLAN Hopping Attacks
VLAN: A way of logically separating/segmenting a physical network into much smaller, isolated networks. Although devices are connected to the same switch, devices on different VLANS

cannot talk to each other unless they go through a router.
Basic Trunk Ports: An ethernet connection that can handle multiple VLANs. Trunk ports have access to all VLANs by default, used to route traffic for multiple VLANs across the same physical link. Encapsulation can be 802.1q or ISL.
VLANs can span multiple different switches.
Auto-trunking: switches automatically negotiate and configure trunk ports between themselves. Eliminates the need for manual configuration of trunk links.
Switch Spoofing: The attacker comes in and pretends to be a switch. Years ago, auto-trunking is on, allowing this attack. To mitigate, turn auto-trunking, don't use VLAN 1 for user traffic, explicitly configure trunking on infrastructure ports.
Double 802.1q Encapsulation Attack: a computer puts 2 different VLAN tags on their headers. When it comes to the first switch, it removes the first tag and accidentally sends it to a second VLAN. The attacker needs to come from the native VLAN - most enterprise networks have VLANs enabled, but there can be legacy part of the network that doesn't have VLANs enabled. Attack only works one way.

DHCP Attacks
DHCP: The protocol that gives IP addresses to anyone connecting to a network. The person coming into the network does a **Discover** as a broadcast message to get to the DHCP server. The DHCP server responds with an **Offer** that offers an IP address to that client. The client then sends a **Request** with the IP address it wants, broadcast. The server then responds with an **Ack** to the client. There are sometimes more than one DHCP servers for redundancy.
If the client gets more than one DHCP offer, it selects the one that comes first. When DHCP server gives IP address, it gives out based on MAC address. If a machine connects again, DHCP will give out same IP, keeping track of MAC addresses. It doesn't look at layer 2 MAC address, it looks at the MAC address inside the DHCP header (called the Client Hardware Address, but it is the same as the MAC address).
DHCP Starvation Attack: When an attacker comes in and takes all the IP addresses. This is done with Gobbler (tool that does it). Let's say DHCP server has 155 IP addresses to give out. Gobbler will make 155 DHCP discoveries, get 155 offers, make 155 requests, and get 155 acks. The attacker needs to change MAC address in the DHCP header, not necessarily the Ethernet Header. Can be easily done in python - scapy.
Mitigation: Port Security, limits number of src MAC addresses on port. Port Security only limits on Ethernet Header - does not limit the Ethernet Header. If port security is only mitigation, it is easy to bypass this type of attack.
Rogue DHCP Server: Attacker pretends to be a DHCP server. Client sends the broadcast discovery, the attacker acts as a DHCP server alongside the actual DHCP server - the client will take the first one that comes, so if the attacker is faster, this attack will work. A DHCP offer has an IP address, the subnet mask, the default routers, the DNS servers, the lease time, and if it's windows, it will also have the proxy servers. They can then manipulate any of that information - for example, route the client to a malicious DNS server.

Countermeasures: On mid-level devices, there is something called DHCP snooping. It has a set of 3 features: it sets each interface as trusted or untrusted: client side as untrusted, server as trusted. The untrusted side can only send stuff the client is able to send - DHCP discovery or request in this example. Cannot send offers or acks. The second feature is keeping a binding table that records all information/messages. Third feature is to match the DHCP MAC address to the Ethernet Header.
DHCP Rogue server can stop by blocking DHCP dport on the switch side if you do not have a more expensive switch. This won't block starvation attacks
Discovery/Request: sport 68, dport 67
Offer/Acks/Naks: sport 67 dport 68

ARP Attacks
Before a station can talk to another, ARP is used to map the IP address to MAC address. Done through an ARP request, broadcast. The reply is sent from that host (I am IP X with MAC Y) and the requester updates the ARP cache.
In ARP header, the MAC address is on it multiple times (as part of the Ethernet Header). These don't have to be the same.
Attacker can poison the ARP table to point the IP addresses to the attacker IP address. There needs to be a poison on both sides of the conversation - poison router and target, otherwise it will be fixed quickly.
Countermeasures:
Dynamic ARP Inspection - only works on more expensive devices, first it requires DHCP snooping to be turned on. It checks the DHCP snooping bind table

Spoofing Attacks
Attacker sends packets with incorrect source MAC address
IPSG: IP source guard, countermeasure. Requires DHCP binding table, reads info and then from that information it looks at IP packet and compares it to table. If the mapping (MAC to IP) in the binding table is correct, it lets it through.
This feature is very expensive ($).

Spanning Tree Protocol
Protocol on enterprise networks that stops loops in networks. In network architecture, you do want loops for redundancy, but you also want the topology to resemble a tree.
STP looks at network and creates the best tree-like architecture by sending Bridge Protocol Data Units. It helps avoid loops and ensure broadcast traffic does not become storms.
An attacker sends BPDU to become root bridge - where all messages are sent through this bridge, and can see messages.

BPDU Guard makes sure that only specific devices can send BPDU messages, switches do not accept BPDU messages from random hosts, just other switches.