CS2107 Introduction to Information Security

AY24/25 Sem 2, github.com/keithxun

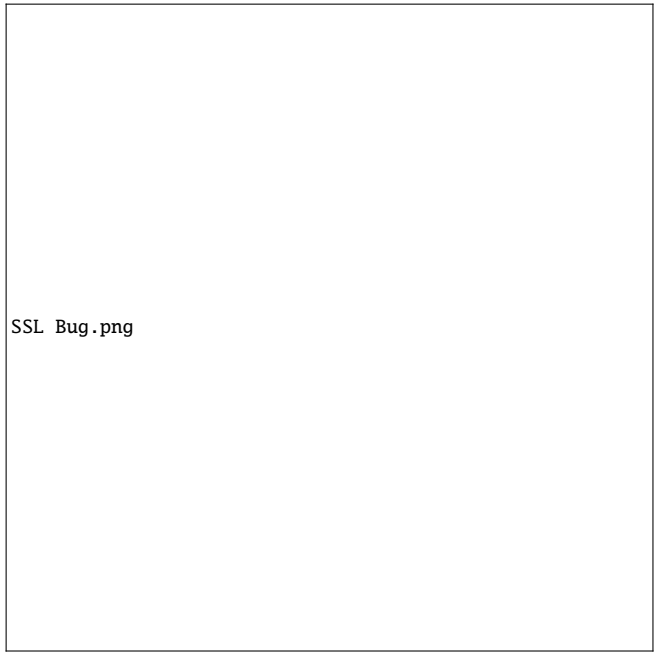
Alice Opens Browser and Visits URL

Phishing Attack: Social Engineering

Trick the user to visit a website which is a spoofed canvas login page

- Typo squatting: Attacker register domain names that are similar to legitimate domain names but with typo error.
- Subdomain Takeover: Users may be tricked into visiting these look-alike subdomains (*.attacker.com) instead of (*.com)

Phishing Attack: SSL Bug



Countermeasures

- User training: Workshops, reminders
- Blacklisting: Repositary site keeping lists of phising sites

Attacker in the Same Local Network

DNS Spoofing

DNS has no confidentiality, integrity, or authentication.

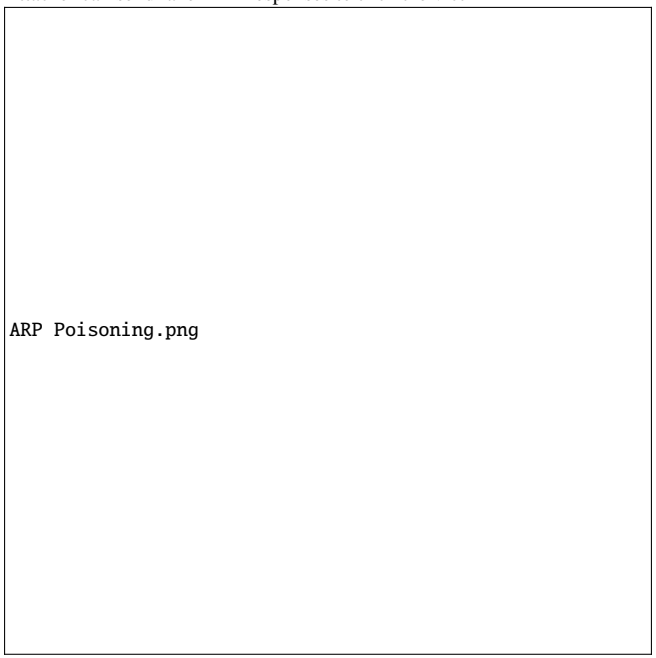
- Manipulate DNS response to redirect users to a fake website
- Same cafe (Physical layer), On-path router (IP layer)
- Capability: Attacker able to send fake response before the actual one

Countermeasures

- DNSSEC: Digital signature for DNS records for authenticity and integrity
- DoH: DNS over HTTPS for confidentiality
- DoT: DNS over TLS
- DoQ: DNS over QUIC

ARP Spoofing

ARP requests are broadcasted to all hosts in the same local network.
Attacker can send fake ARP responses to trick the victim

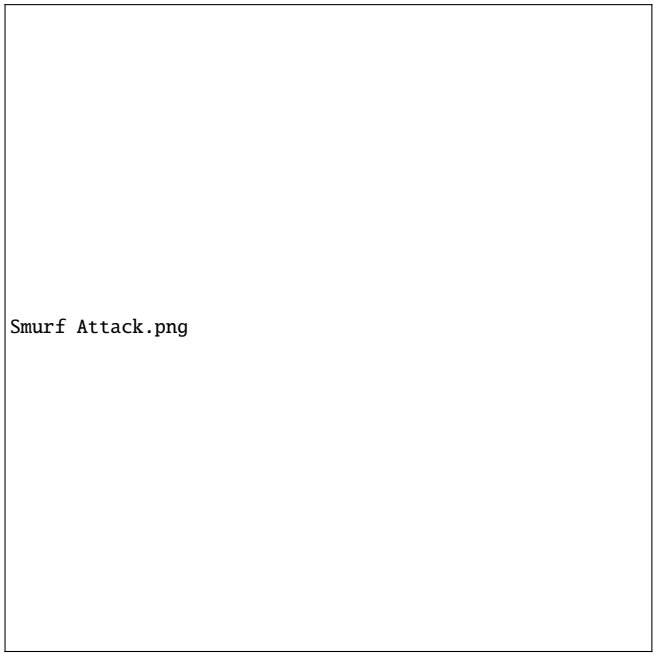


- Attacker poisons the ARP tables so as to gain MITM access
- Attack is present in the local network (Physical Layer) and able to send ARP responses

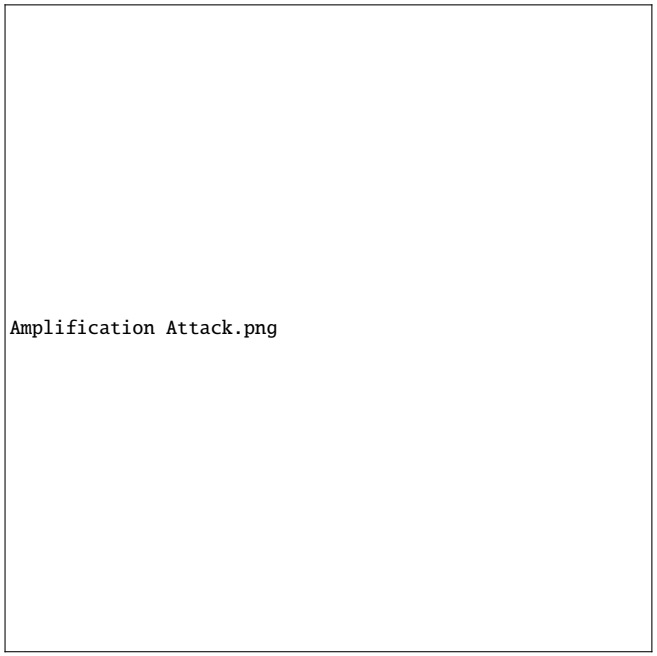
Countermeasures

- Static ARP entries: Manually configure the ARP table
- Packet filtering: Only allow ARP requests from trusted hosts, block single MAC address associated to multiple IPs

DDOS: Smurf Attack



DDOS: Amplification Attack



Countermeasures

- configure firewalls to block incoming ICMP Echo requests packets directed at broadcast addresses
- Redundant servers, rate limiting, etc

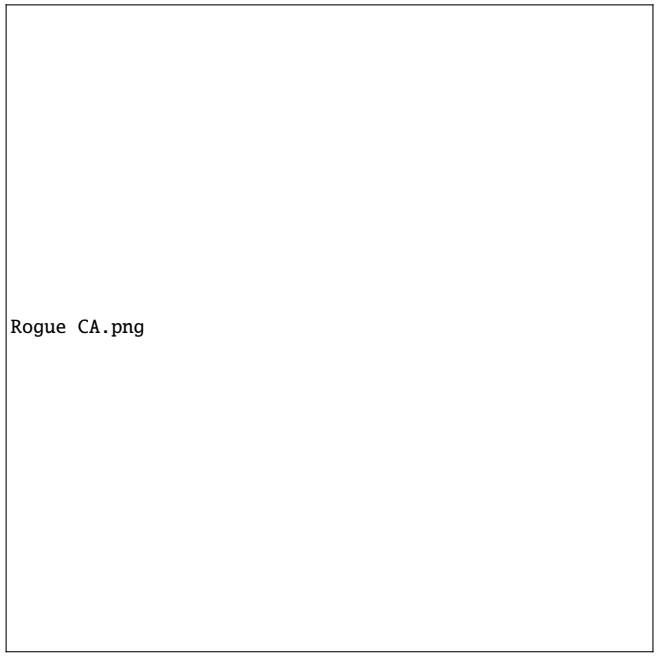
- Load balancing for amplification attack

Establishing HTTP Session

Certificates

- Certificate binds an entity to some public key (Identity, public key, validity period)
- Certificate Authority (CA) is a trusted third party that issues digital certificates
- CA signs the certificate with its private key
- CA's public key is distributed to all users
- Self-signed certificate: CA is the entity itself
- Domain Validated certificate: Least stringent validation, only verifies if applicant owns the domain
- Extended Validation certificate: More stringent validation, Includes organization details, registration number and jurisdiction etc

MITM due to Rogue CA



Countermeasures

- Certificate Revocation: To invalidate certificates (OCSP, CRL)
- Short-lived certificate: No need to revoke
- Certificate Log: Public, verifiable, and append-only log of certificates

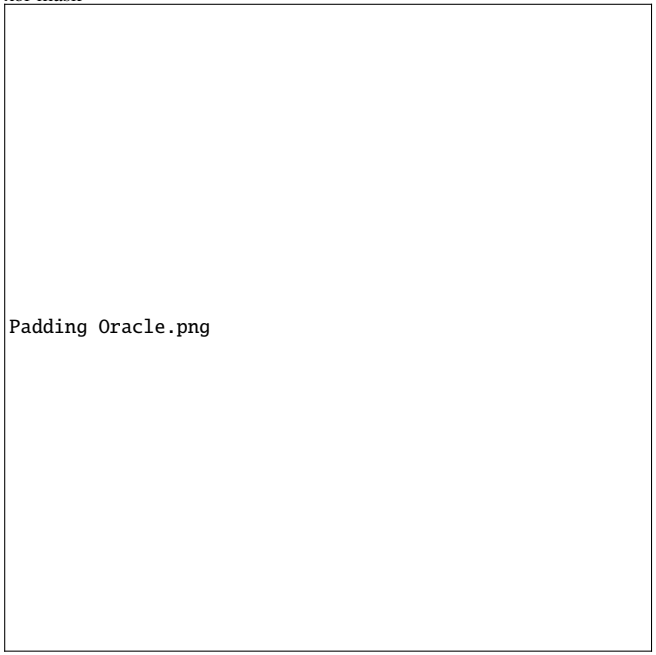
Security Principle

confidentiality, integrity, availability

- Confidentiality: Only authorized parties can access the data
- Integrity: Data is not modified by unauthorized parties
- Availability: Authorized parties can access the data when needed

Confidentiality

- Symmetric Encryption scheme: Same key for encryption and decryption
- Substitution and Permutation: Not secure - Brute force, CTO, KPA
- One-time Pad: Theoretical, secure under certain conditions
- Stream cipher: Not secure if key is reused, needs IV
- DES (not secure), AES (secure)
- MITM on DES: Ecrypt from one side, decrypt from the other side
- Padding Oracle Attack: Mask = New pad xor old pad, actual = new pad xor mask



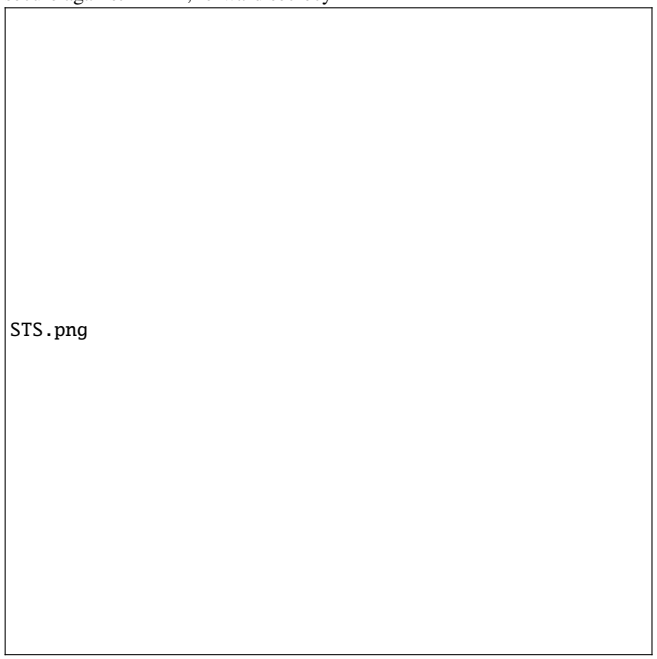
- Asymetric Encryption scheme: Different keys for encryption and decryption
- RSA: Integer factorization problem, multiplying two large primes to generate n is easy but factoring n is hard
- $n = pq$, $\phi = (p-1)(q-1)$, e = public exponent, d = private exponent such that $\gcd(e, \phi) = 1$ and $ed \bmod \phi = 1$
- $c = m^e \bmod n$, $m = c^d \bmod n$
- RSA is significantly slower than AES
- Bell-LaPadula Model: No read up, no write down

Integrity

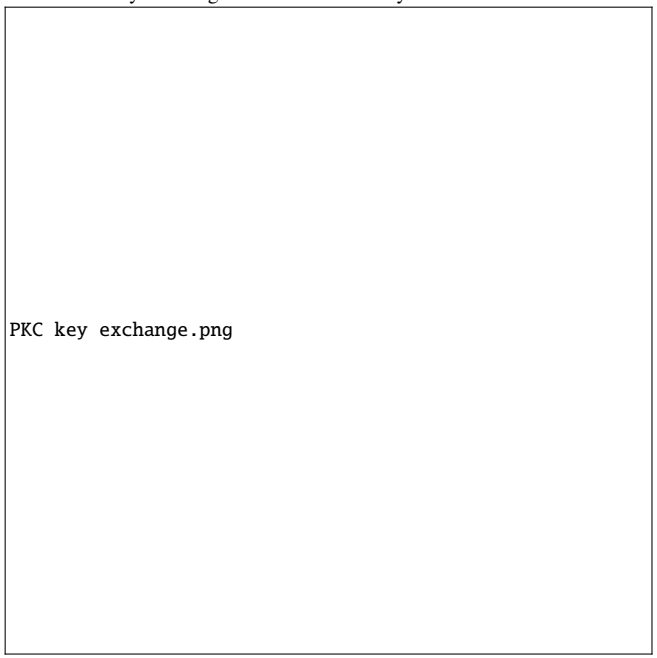
- Hash: No authentication
- MAC: Authentication and integrity, keyed, symmetric
- Signature: Authentication and integrity, asymmetric, non-repudiation
- Birthday attack: Find collision after $1.17 \times 2^{n/2}$ hashes
- Collision resistance: No 2 inputs produce the same hash
- Pre-image resistance: Cannot get the input from the hash
- Second pre-image resistance: Given an input, cannot find another input that produces the same hash
- Biba Model: No write up, no read down

Authenticated Key Exchange

- Station to Station Protocol: Deffie Hellman key exchange protocol, not secure against MITM, forward secrecy



- MITM attack: Attacker intercepts the public keys and sends his own public key to both parties
- PKC-based Key Exchange: No forward secrecy



- Perfect forward secrecy: Even if the private key is compromised, past

sessions are still secure

Authenticated Encryption

- Encrypt-then-MAC: Encrypt the message first, then MAC it, secure and can verify MAC before decrypting
- MAC-then-encrypt: MAC the message first, then encrypt it, not secure, no ciphertext integrity
- Encrypt-and-MAC: Encrypt and MAC at the same time, not secure, no ciphertext integrity
- SSL/TLS protects data in transport layer
- IPsec protects data in network layer
- WPA2 protects data over Wi-Fi between link and physical layer
- VPN tunnel at layer 3 (IP layer) to further improve security, hides everything but src mac and dest mac

Passwords

- Bootstrapping: Establishing common password
- Entropy: $H =$ Password Length $\times \log_2$(Symbol Set Size)
- 2FA: Something you know, something you have, something you are. Different from multi-step verification
- Attack on Password reset: If reset link is constructed by taking the domain name from the host header of http request without validation, attacler post request to host and gets victim's otp
- Online attack (rate limit) vs Offline attack (Add salt and hash)

Cookies

Choice of token should include mac computed using server secret key Same origin matches protocol, hostname and port number. Same site matches protocol and last part of hostname only (String comparison www. matters)

Properties

- Domain: Domain that can access the cookie
- Path: Path that can access the cookie
- Secure: Only sent over encrypted HTTPS
- HttpOnly: Not accessible via JavaScript
- SameSite (Strict or lax): Prevent CSRF attacks
- Expiration date: When the cookie expires

CSRF

- Victim has logged into the site and has a valid session cookie
- Victim clicks a malicious link to attacker's site
- Malicious site issue a stealthy request to the target site

Countermeasures

- SameSite cookie: Prevents CSRF attacks by not sending cookies with cross-site requests
- CSRF token: Unique token for each request, sent as a hidden field in the form
- Referer header: Check if the request is coming from the same site

XSS

- Reflected XSS: Attacker injects malicious script into the URL, victim clicks the link and the script is executed
- Stored XSS: Attacker injects malicious script into the website, victim visits the website and the script is executed
- Attacker tricks user to click on malicious link, which contains the target website and a malicious script
- Server constructs a response html that contains the script and browser then runs the script

Countermeasures

- Input validation: Validate all user inputs
- HttpOnly cookie: Prevents JavaScript from accessing the cookie

SQL Injection

- Attacker injects malicious SQL code into the input field
- Server constructs a SQL query that contains the malicious code
- Attacker can manipulate the database
- Example: Attacker enters "bob' OR 1=1; –" in the input field
- Server constructs a SQL query that looks like this: "SELECT * FROM User WHERE name = ' " + userinput1 + " ' AND password = ' " + userinput2 + " '";
- The query returns all users in the database

Countermeasures

Parameterized queries: Use prepared statements to prevent SQL injection

Access Control

Exploit software vulnerabilities

- Stack overflow: Attacker injects malicious code into the stack (Carnaries, memory randomization)
- Integer overflow: Leads to unintended behaviour
- Format string attack: Attacker injects malicious format string into the input field
- Unsafe functions such as gets, scanf, sprintf

Time of Check to Time of Use

Attacker can exploit the time gap between checking the access control and using the resource by deleting the original file and creating a link with the same name

Countermeasures

- Avoid separate system calls for checking and using the resource
- Set effective UID to the appropriate user