Class Notes

September 13th, 2021

**Supply Chain Attack**:

When people attack weaknesses in a supply chain to leave vulnerabilities that can be used later. (Example: Swapping out CPU’s that have known hardware defects to exploit later)

**Threat Model:**

When we’re designing a system, we need to define the goals and capabilities of the attacker. Its not always a software level threat, it could be part of the hardware, or the supply chain.

**Apple CSAM Scanning**:

They use Perceptual Hashing to identify CSAM content. The use PhotoDNA to search content on servers.

**Perceptual Hashing**:

Take a photo, throw it into some black box, and the black box spits out a hash (fingerprint). It’s much different from a cryptographic hash, if two images seem to be the same, we want the hash to be similar as well.

**Average Hash (aHash)**:

1.) Change the image to gray scale (Only need 8 bits per pixel compared to full color)

2.) Downsample the image. Example – For 64bit hash -> Image is 8x8 pixels

3.) Compute the hash

* Compute the average value of the 64 pixels
* Go throughg pixel by pixel next and calculate each individual pixel score
* Less than average = 0, anything else is assigned a 1

**Secure Multi-Party Computation:**

A group of people computing on the same data, without anyone knowing the value of the data their computing on.

**HTTPS:**

The secured version of HTTP which encrypts your connection to web sites.

**How website certifications work:**

* HTTP Request -> GET /index.html
* HTTP Request -> POST (Username & plaintext password) All passwords are hashed server side with SALT. Both the password HASH and SALT are stored on the server.
* HTTP Responds with Token to give access to log into the website
* Token has a lifetime which reduces the amount of time someone can use it
* Don’t need to store username or password, just store the Token instead.

**Basic Authentication Scheme:**

* Sends the username and password every time you send a request to the server inside the header.
* The plaintext is Base64 encoding.

**Cookie Based Authentication:**

* Token/Session ID is stored as a Cookie
* Cookie is sent to the web server for future requests
* The server can specify the cookie details

Example: Server Reponse

HTTP/20 200 OK

Content-Type Text/HTML

Set-Cookie: yummy-cookie = chocolate (name & value)

<Web Content>

Example:

GET /index.html Http/2.0

HOST example.com

Cookie: yummy-cookie = choco; secure; HttpOnly; Path =/...

\*\*The attributes define how the cookie will be used\*\*

**Cookie Attributes:**

RFC 6265 (The Standard)

* Expires: The time to live for the token
* Max-Age: How old the token can live for
* Domain: Handles domain and subdomains
* Path: The part of the website after the .com/(HERE)
* Secure: Only sent over HTTPS (Protects against eavesdropping)
* HTTPOnly: Not accessed via JavaScript (Protects against Cross-Site Scriping)