- Privilege escalation
    - Gain higher level access to a system
        - Exploit a vulnerability
        - Might be a bug or a design flaw
    - Higher level access means moer capabilities
        - This commonly is the highest level access
        - This is obviously a concern
    - These are high priority vulnerability patches
        - You want to get these holes closed very quickly
        - Any user can be an admin
    - Horizontal privilege escalation
        - **User A can access user B resources**
- Mitigating privilege escalation
    - Patch quickly
        - Fix the vulnerability
    - Updated AV/AM
        - Block known vulnerabilities
    - Data Execution Prevention
        - Only data in executable areas can run
    - Address space layout randomization
        - Prevent a buffer overrun at a known memory address
- Elevation of privilege vulnerability
    - CVE-2020-1530
        - Windows remote access elevation of privilege vulnerability
        - Windows Remote Access
            - 7,8.1,10
    - Attacker would execute a program on a victim computer
    - Vulnerability in Remote Access would elevate privileges

---

- Cross Site Scripting
    - XSS
        - Cascading Style Sheets (CSS) are something else entirely.
    - Originally called cross site because of browser security flaws
        - Information from one site could be shared with another site
    - One of the most common web application development errors
        - Takes advantage of the trust a user has for a site
        - Complex and varied
    - Malware that uses JavaScript?
        - Do you allow scripts? Me too
- Non-persistent (reflected) XSS Attack
    - Website allows scripts to run in user input
        - Search box is a common source
    - Attacker emails a link that takes advantage of this vulnerability
        - Runs a script that sends credentials/session IDs/ cookies th the attacker

    - Script embedded in URL executes in the victim's browser
        - As if it came from the server
    - Attacker uses credentials/session IDs/cookies to steal victim's knowledge
        - Very sneaky
- Persistent (stored) XSS attack
    - Attacker posts a message to a social network
        - Includes the malicious payload
            - Its now "persistent"
            - Everyone gets the payload
        - No specific target
            - All viewers to the page
        - For social networking, this can spread quickly
            - Everyone who views the message can have it posted to their page
                - Where someone can view it and then propagate to others
- Hacking a Subaru
    - June '17, Aaron Guzman
        - Security researcher
    - When authenticating with Subaru, users get a token
    - This token never expires (bad!)
    - A valid token allowed any service request
        - Even adding your email address to someone else's account
        - Now you have full access to someone else's car
    - Web front end included an XSS vulnerability
        - A user clicks a malicious link, and you have their token.
- Protecting against XSS
    - Be careful when clicking untrusted links
        - Never blindly click in your email inbox. Never
        - Consider disabling JS
            - Or control with an extension
            - This offers limited protection
        - Keep your browser and applications up to date
        - Validate all inputs
            - Don't allow users to add their own scripts to an input field.

- Code injection
    - Code injection
        - Adding your own information into a data stream
    - Enabled because of bad programming
        - This application should properly handle input and output
    - So many different daat types:
        - HTML, SQl, XML, LDAP
- SQL Injection
    - SQL - Structured Query Language
        - The common relational database management system language

    - SQL Injection
        - Modifying SQL requests
        - Your application shouldnt really allow this.
    - Example SQL injection
        - Inputting employee name and authentication tan (Transaction address basically)
        - Inputting into both fields, Smith and 3SL99A would provide the information for that specific user but since the input box can take scripts if i add OR'1'=1 which 1 does = 1 which means return everything that's TRUE it will output the entire database.
- XML injection and LDAP injection
    - XML - Extensible Markup Language
        - A set of rules for data transfer and storage
    - XML Injection
        - Modifying XML requests, a good application will validate
    - LDAP - Lightweight Directory Access Protocol
        - Created by the telephone companies
        - Now used by almost everyone
    - LDAP injection
        - Modify LDAP requests to manipulate application results
- DLL Injection
    - Dynamic Link Library
        - A windows library containing code and data
        - Many applications can use this library
    - Inject a DLL and have an application run a program
        - Runs as part of the target process.

- Buffer overflows
    - Overwriting a buffer of memory
        - Spills over into other memory areas
    - Developers need to perform bounds checking
        - The attackers spend a lot of time looking for openings
    - Not a simple exploit
        - Takes time to avoid crashing things
        - Takes time to make it do what you want
    - A really useful buffer overflow is repeatable
        - Which means that a system can be compromised

- Replay Attack
    - Useful information is transmitted over the network
        - A crafty hacker will take advantage of this.
    - Need access to the raw network data
        - Network tap, ARP poisoning
        - Malware on the victim computer

        - The gathered information may help the attacker (Replay Attack)
            - Replay the data to appear as someone else
        - This is not an on path attack
            - The actual replay doesn't require the original workstation
            - Pass the hash example
            - Avoid this type of replay attack with a salt or encryption
                - **SSL or TLS is encrypted and can't find any hash info**
            - Use a session ID with the password hash to create a unique auth hash each time
- Browser cookies and session IDs
    - Cookies
        - Information stored on your computer by the browser
    - Used for tracking personalization, session management
        - Not executable, not generally a security risk
            - Unless someone gets access to them
        - Could be considered be a privacy risk
            - Lots of personal information in there
        - Session IDs are often stored in the cookie
            - Maintains sessions across multiple browser sessions
- Header manipulation
    - Information gathering
        - Wireshark, Kismet
    - Exploits
        - Cross site scripting
    - Modify headers
        - Tamper, Firesheep, Scapy
    - Modify cookies
        - Cookies Manager+
            - Firefox add on
- How to prevent
    - Encrypt end to end
        - They can't capture your session ID if they cant see it
        - Additional load on the web server HTTPS
        - Firefox: HTTPS Everywhere, force TLS
        - Many sites are now HTTPS only
    - Encrypt end to somewhere
        - At Least avoid capture over a local wireless network
        - Still in the clear for part of the journey
        - Personal VPN (OpenVPN, VyprVPN, etc)

- Cross Site requests
    - Cross site requests are common and legitimate
        - Loads professor messer server
        - Loads a video from YT

        - Loads pictures from IG
    - HTML on ProfessorMesser.com directs requests from your browser
        - This is normal and expected
        - Most of these are unauthenticated requests
- The client and the server
    - Website pages consist of client side code and server-side code
        - Many moving parts
    - Client side
        - Renders the page on the screen
        - HTML, JS
    - Server side
        - Performs requests from the client
        - HTML, PHP
        - Transfer money from one account to another
        - Post a video on YT
- Cross site request forgery
    - One click attack, session riding
        - **XSRF, CSRF (sea surf)**
        - Takes advantage of the trust that a web application has for the user
            - The website trusts your browser
            - Requests are made without your consent or your knowledge
            - Attacker posts a FB status on your account
    - Significant web application development oversight
        - The application should have anti forgery techniques added
        - Usually a cryptographic token to prevent a forgery
- Server side request forgery (SSRF)
    - Attacker finds a vulnerable web application
        - Sends requests to a webs server
        - Web server performs the request on behalf of the attacker
    - Caused by bad programming
        - Never trust the user input
        - Server should validate the input and the responses
        - These are rare, but can be critical vulnerabilities
- Capital One SSRF breach
    - Attacker is able to execute commands on the Capital One website
        - This is normally stopped by a WAF (Web Application Firewall)
        - The WAF was misconfigured
        - Attacker obtained security credentials for the WAF role
        - WAF role account listed the buckets on Amazon S3
        - Attacker retrieved the data from the Amazon buckets

---

- Malware hide and go seek
    - Traditional antivirus is very good at identifying known attacks
        - Checks the signature

        - Block anything that matches
    - There are still ways to infect and hide
        - Its a constant war
        - Zero day attacks, new attack types, etc.
- Your drivers
    - The interaction between the hardware and your operating system
        - They are often trusted
        - Great opportunity for security issues
    - May 2016 - HP audio drivers
        - Conexant audio chips
        - Driver installation includes a audio control software
        - Debugging feature enables a keylogger
        - Hardware interactions contain sensitive information
            - Video keyboard mouse
- Shimming
    - Filling in the space between two objects
        - A middleman
    - Windows includes its own shim
        - Backwards compatibility with previous Windows versions
        - Application compatibility Shim Cache
    - Malware authors write their own shims
        - Get around security (like UAC)
- Refactoring
    - **Metamorphic Malware**'
        - A different program each time its downloaded
    - Make it appear diff each time
        - Add NOP instructions
        - Loops, pointless code strings
    - Can intelligently redesign itself
        - Reorder functions
        - Modify flow
        - Reorder code and insert unused data types
    - Difficulty to match with signature-based detection
        - Used a layered approaches

- SSL Stripping / HTTP downgrade
    - Combines an on-path attack with a downgrade attack
    - Difficult to implement, but big returns for the attacker.
- Attacker must sit in the middle of the conversation
    - Must modify data between the victim and web server
        - Proxy server, ARP spoofing, rogue WIFI hotspot
    - VIctim does not see any significant problem
        - Except the browser isnt encrypted
        - Strips the A away from HTTPS

    - This is a client and server problem
        - Works on SSL and TLS.
    - SSL (Secure Sockets Layer) 2.0
        - Deprecated in 2011
    - SSL 3.0
        - Vulnerable to the POODLE attack
            - June 2015
    - TLS Transport Layer Security (1.0)
        - Upgrade to SSL 3.0
        - Can downgrade to SSL 3.0
        - TLS 1.1
            - Deprecated in 2020
        - Tls 1.2 and 1.3 the latest standards
    - Web site visitor —> attacker ←– web server
        - Intercepts the HTTPS and converts to HTTPS and sends it back to web site visitor. Sends post for login and user info and the attacker can grab it.

- Race condition
    - A programming conundrum
        - Sometimes, things happen at the same time
        - This can be bad if you've not planned for it
    - Time-of-check to time-of-use attack (TOCTOU)
        - Check the system
        - When do you use the results of your last check?

- Memory Vulnerability
    - Manipulating memory can be advantageous
        - Relatively difficult to accomplish
    - Memory leak
        - Unused memory is not properly released.
        - Begins to slowly grow in size
        - Eventually uses all available memory
        - System crashes
    - NULL Pointer dereference (pointing to nothing in memory)
        - Programming technique that references a portion of memory
        - Application crashes, DoS, debug information
    - Integer overflow
        - Large numbers into a smaller sized space.
        - Where does the extra number go
        - You shouldnt be able to manipulate memory this way.
    - Directory traversal attack
        - Directory traversal / path traversal
        - Read files from a web server that are outside of the website's file directory
        - Users shouldnt be able to browse the Windows folder.

        - Web server software vulnerability
            - Wont stop suers from browsing past the web server root
        - Web application code vulnerability
            - Take advantage of badly written code
    - Improper error handling
        - Errors happen, and you should probably know about it
        - Messages should be informational enough
            - Avoid too much detail
            - Network info
            - Memory dump
            - Stack traces
            - Database dumps
            - Easy to fix
    - Improper input handling
        - Many applications accept user input
            - We put data in, we get data back
        - All input should be considered malicious
            - Check everything
        - Allowing invalid input can be devastating
            - SQL injections, buffer overflow, DoS
        - It takes a lot of work to find input that can be used maliciously
    - API attacks
        - API - Application Programming interface
        - Attackers look for vulnerabilities in this new communication path
            - Exposing sensitive data, dos, privileged access
    - Resource exhaustion
        - A specialized dos attack
            - Zip bomb
            - Uncompressed to 4.5 petabytes from 42 kb compressed file.
            - Antivirus will identify these vulnerabilities
        - DHCP starvation
        - Attack floods a network with Ip address requests
        - MAC address changes each time
        - DHCP server eventually runs out of addresses
        - Can limit DHCP requests
