**Module 1 Threats, Attacks & Vulnerabilities**

**1.22 Application/Service Attacks**

**Application Attacks**

* Buffer Overflow
* Injection
* Cross-site scripting (XSS)
* Cross-site request forgery (CSRF/XSRF)
* Privilege escalation

**OWASP Top 10 Application Security Risks (2017)**

1. Injection
2. Broken Authentication
3. Sensitive data exposure
4. XML external entities (XXE)
5. Broken Access control
6. Security misconfiguration
7. Cross-site scripting (XSS)
8. Insecure deserialization
9. Using components with known vulnerabilities
10. Insufficient logging & monitoring

**Buffer Overflow**

* When more data written to buffer than it can hold
* An anomaly where a program when writing data to a buffer, overruns buffer’s boundary & overwrites adjacent memory locations

**Injection**

* Occur when untrusted data sent to interpreter as part of a command/query
* Most commonly fall into these categories:

1. Escape characters not filtered correctly
2. Type handling not properly done
3. Conditional errors
4. Time delays

* Way to defend is to always filter input
* Eg. SQL/OS/LDAP/XML Injection

**Cross-site Scripting & Cross-site Request Forgery**

* Cross-site scripting (XSS) – occur whenever app includes untrusted data in new web page w/o proper validation/escaping, or updates an existing webpage with user-supplied data using browser API that can create HTML/Javascript

1. Eg. Ron<SCRIPT>alert(‘hello’)</SCRIPT>Woerner

* Cross-site request forgery (CSRF/XSRF) – attack that forces end user to execute unwanted actions on web app. AKA session riding/one-click attack

**Privilege Escalation**

* Act of exploiting bug, design flaw, or configuration oversight in OS/software app to gain elevated access to resources normally protected from app/user

**Application Attacks (Prevention & Response)**

* Good coding practices – see OWASP
* Filter & validate any user input
* Use Web Application Firewall (WAF)
* Build security into Software Development Life Cycle (SDLC)
* Have incident response plan in place

**Zero-Day (0 Day) Exploits**

* Attack that exploits previously unknown security vulnerability
* Takes advantage of security vulnerability on same day that vulnerability becomes generally known
* Eg. Stuxnet
* Prevention

1. Defence in depth
2. Patch
3. Keep antivirus up-to-date

**Impersonation/Masquerading/Replay Attacks**

* Act of pretending to be someone/something to gain unauthorised access to system
* Capturing network traffic via eavesdropping, then re-establishing communications session by replaying captured traffic using spoofed authentication credentials
* Prevention – token authentication (Kerboros), MFA/TFA, Encryption, Sequenced session identification

**Driver Manipulation**

* Driver – program that controls a device (Eg. Printers, media, keyboards etc)
* Shimming – creating a library/modifying an existing library to bypass driver & perform function other than the one for which API was created
* Refactoring – set of techniques used to identify flow & then modify internal structure of code w/o changing code’s visible behaviour

**Cryptographic Attacks**

* Birthday – attack on cryptographic hash that looks for hash collisions, exploiting 1-to-1 nature of hashing functions
* Known plain text/cipher text – attacker attempts to derive cryptographic key by using pairs of known plain text along with corresponding cipher text
* Frequency analysis – looking at blocks of encrypted message to determine if any common patterns exists
* Dictionary – systematically entering each word in dictionary as password
* Brute force – systematically attempting all possible combinations of letters, numbers & symbols. Usually automated
* Rainbow tables – all possible password hashes computed in advance & those hash values compared with password database
* Pass the hash – attacker attempts to authenticate to remote server/service by intercepting password hashes on network