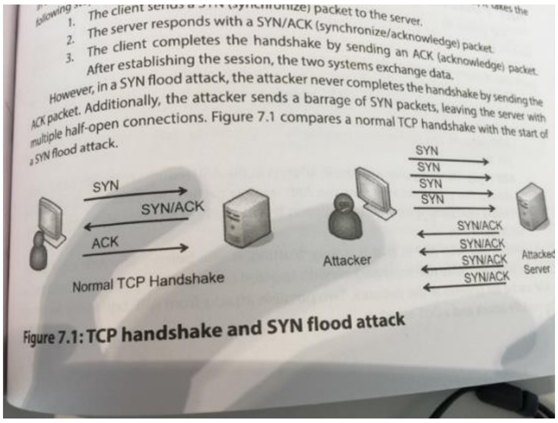
Sec+ 501 Get Certified Topic 7 – Protecting against advanced attacks

CH 7

- DoS: Denial of service attack is an attack from a single source that attempts to disrupt the services provided by another system.
- A distributed denial of service (DDoS) attack includes multiple computers attacking a single target. DDoS attacks typically include sustained, abnormally high network traffic
- Spoofing: Spoofing attacks typically change data to impersonate another system or person. MAC spoofing attacks change the source MAC address and IP spoofing attacks change the source IP address.
    - Host systems on a network have a media access control (MAC) address assigned to the network interface card (NIC). It is possible to use different software methods to associate a different MAC address to the NIC.
- Syn Flood attack: Common attack used against servers on the internet. Acts like a TCP handshake.
    - 1. Client sends a SYN (synchronise) packet to server
    - 2. Server responds with a SYN/ACK (Synchronize/acknowledge) packet
    - 3. Client completes the handshake by sending an ACK (acknowledge) packet. After establishing the session, the two systems exchange data.
    - 4. However, in SYN flood attack, attacker never completes handshake by sending the ACK packet. Attacker sends a barrage of SYN packets leaving server with multiple half-open connections



following ... takes the

1. The client sends a SYN (synchronize) packet to the server.
2. The server responds with a SYN/ACK (synchronize/acknowledge) packet.
3. The client completes the handshake by sending an ACK (acknowledge) packet. After establishing the session, the two systems exchange data.

However, in a SYN flood attack, the attacker never completes the handshake by sending the ACK packet. Additionally, the attacker sends a barrage of SYN packets, leaving the server with multiple half-open connections. Figure 7.1 compares a normal TCP handshake with the start of a SYN flood attack.

Figure 7.1: TCP handshake and SYN flood attack

    - This consumes server's resources waiting for 3rd packet, resulting in a crash and may block legitimate users. Attackers can launch SYN flood attacks from a single system in a DoS attack.
- MITM (Man in the Middle attacks) – Form of active interception or active eavesdropping.
    - Uses a separate computer that accepts traffic from each party in a conversation and forwards traffic between the 2. Two computers are unaware of the MITM computer and it can interrupt traffic and insert malicious code.
        - E.G: ARP poisoning
    - Kerberos can help prevent MITM attacks with mutual authentication.
- ARP poisoning: Attempts to mislead systems (computers/switches) about the actual MAC address of a system. Is sometimes used in MITM attacks.
    - MAC Address: Physical/Hardware address tied to the NIC [Network Interface Card]
    - ARP resolves IP addresses of systems to their MAC address and stores result in an ARP Cache within memory.
    - Arp request (broadcasts IP asking for IP) → ← Arp reply (responds with its MAC address)
- DNS attacks:
    - DNS – used to resolve host names to IP addresses
        - E.G: Just typing hostname to connect instead of IP such as gcgapermium.com as the URL in browser.
        - DNS also provides reverse lookups. Reverse Lookup: Client sends IP address to a DNS server with a request to resolve it to a name.
    - DNS poisoning: Tries to modify or corrupt DNS results.
        - E.G: Modify IP address associated with google.com and replace it with IP address of a malicious web site.
            - Use DNSSEC – DNS Security Extensions to protect DNS records and prevent DNS poisoning attacks
    - DNS pharming:

        - Manipulates DNS name resolution process, corrupt DNS server or DNS client. Can be used to redirect users to different websites.
            - E.G: Hosts file on Windows (C:\Windows\System32\Drivers\etc)
                - 127.0.0.1 local host & 13.207.21.200 google.com [IP addresses with host name mappings]
                - Can change IP address of google to Bing (13.207.21.200) instead for DNS pharming attack
    - DDoS DNS Attacks
        - Disrupt DNS services causing internet access problems to multiple people.
- Amplification attacks:
    - **Type of DDoS attack. Uses a method to significantly increase amount of traffic sent to or requested from a victim and can be used against a wide variety of systems including individual hosts, DNS servers and NTP servers.**
    - A ping is normally unicast -one computer to one computer through a ICMP echo request to one computer and the receiving computer responds with ICMP echo responses
    - Smurf attack sends ping out as a broadcast to all other computers in a subnet.
    - Smurf attack spoofs source IP – victim gets flooded with ICMP replies.
    - DNS amplification – Sends DNS request to DNS servers spoofing IP address of a victim. Instead of asking for one record, attack tells DNS servers to send as much zone data as possible amplifying the data. Repeat the process.
    - NTP amplification attack – monlist command. Normally would send list of last 600 hosts that connected to the NTP server. In an NTP amplification attack with monlist, attacker spoofs source IP when sending command, flooding victim with details of last 600 systems that requested time from the NTP server.
- Password attacks:

  - Brute force attacks – attempts to guess all possible character combinations
    - Online – repeatedly guessing username and password in an online system. Can be automated with tools like ncrack.
    - Offline – Attempt to discover passwords from captured database/captured packet scan.
  - Dictionary attacks – use a file of words and common passwords to guess a password. Account lockout policies help protect against brute force attacks and complex passwords thwart dictionary attacks.
- **Password Hashes/Hash attack:**
  - Many systems don't store the actually password for an account. Instead they store a hash of the password.
  - Hash attacks attack hash of a password instead of the password
    - E.G: MD5, SHA-3
  - If hash is unencrypted, attacker may be able to capture hash and use it to log on to a system through protocol analyzers.
- Pass the Hash Attack:
  - Attacker discovers hash of the user's password and uses it to log on to the system as the user.
  - Any authentication protocol that passes hash over the network in an unencrypted format is susceptible to this attack.
  - Associated with Microsoft LAN Manager (LM) and NT LAN Manager (NTLM), two old security protocols.
  - Solution: Kerberos or NTLMv2 where NTLMv2 uses a nonce (one use to prevent password reuse) on both client and authenticating server. Can do this through group policy.
- Birthday attack:
  - Technique used by an attacker to find two different messages that have the same identical hash digest
    - 99% chance of finding a matching birthday in a 57-person group
    - 50% chance of finding a matching birthday in a 23-person group

    - Hash Collision
        - Occurs when two different inputs to a hash create an identical hash digest output
        - E.G: password ‘success’ might create hash of 123 and password ‘password’ might also create same hash of 123. Attacker could use both passwords and both would work. Not desirable
    - MD5 uses 128 bits and is susceptible to birthday attacks
- Rainbow table:
    - Type of attack that attempts to discover the password from the hash. A rainbow table is a huge database of precomputed hashes.
        - 1. Application guesses a password
        - 2. Application hashes guessed password
        - 3. Application compares original password hash with guessed password hash. If they are the same, application now knows the password
        - 4. If they aren’t the same, application repeats steps 1-3 until it finds a match.
- Salting – A method to prevent rainbow table attacks
    - Salting adds additional characters to a password before hashing it adding complexity and results in a different hash than the system would create using original password.
        - E.G: PBKDF2
- **Remember this:**
    - **Passwords are typically stored as hashes. A pass the hash attack attempts to use an intercepted hash to access an account. Salting adds random text to passwords before hashing them and thwarts many password attacks, including rainbow table attacks. A hash collision occurs when the hashing algorithm creates the same hash from different passwords. Birthday attacks exploit collision in hashing algorithms.**
- Replay attack:
    - Replay attacks capture data in a session with the intent of later impersonating one of the parties in the session in

    wireless and wired networks. Attacker may capture then modify data in packet to impersonate, then replay the data until it succeeds. Timestamps and sequence numbers are effective countermeasures to this attack.
        - E.G: Kerberos with timestamped tickets.

- Known Plaintext attacks:
    - Plaintext: Human readable data.
    - Encryption algorithms scramble data creating cipher text
        - E.G: AES
    - Known plaintext attack: An attacker has samples of both plaintext and ciphertext. Attacker wants to discover encryption and decryption method so he can use same decryption method on other ciphertext.
    - Ciphertext plaintext attack: Similar, but attacker doesn't have access to plaintext.
    - Ciphertext only attack: Attacker doesn't have any information on the plaintext.
    - Usually always successful with time and resources but usually used with weak encryption algorithms.
- **Remember this:**
    - **Attackers purchase similar domain names in typo squatting (URL hijacking) attacks. Users visit typo squatting domain when they enter URL incorrectly with a common typo. In a session hijacking attack, attacker utilizes user's session ID to impersonate the user. In a domain hijacking attack, an attacker changes the registration of a domain name without permission from the owner.**
- Man-in-the-Browser:
    - Type of proxy Trojan horse infecting vulnerable web browsers. Captures browser session data and may include keyloggers along with all data sent to and from web browser.
        - E.G: Zeus – Trojan horse using man-in-the-browser techniques including keystroke logging and form grabbing. Collected logon information for a user's bank and used it to log on and transfer money to offshore accounts.

- Driver Manipulation:
    - Shimming – makes it appear older drivers are compatible.
    - Refactoring code – Rewriting internal processing of code without changing external behaviour. Usually used to correct problems within software design.
    - Attackers can run malicious code contained within a manipulated driver if they are able to fool OS to use the manipulated driver via strong programming skills.
- Zero-day exploit: Undocumented and unknown to public. Vendor might know about it but has not released a patch to fix it.

MEMORY BUFFER VULNERABILITIES

- Memory leak – bug in an application that causes application to consume more and more memory the longer it runs.
    - In rare cases the application exceeds memory of OS and crashes it
- Integer overflow – An attack that attempts to use/create a numeric value that is too big for an application to handle resulting in application giving inaccurate results.
    - E.G: Application reserves 8 bits to store a number between values 0-255. If application attempts to multiple 2 values such as 95*59 result is 5605. This causes an integer overflow error.
    - If application doesn't have error routines may cause a buffer overflow error.
- Buffer overflow
    - Buffer Overflow
        - Occurs when a process stores data outside the memory range allocated by the developer
            - E.G: GET /index.php?username=ZZZZZZZZZZZ
    - Buffer
        - A temporary storage area that a program uses to store data
        - Over 85% of data breaches were caused by a buffer overflow
- **Remember this:**
    - **Buffer overflows occur when an application receives more data than it can handle or receives unexpected data that**

exposes system memory. Buffer overflow attacks often include NOP instructions (such as x90) followed by malicious code. When successful, attack causes system to execute the malicious code. Input validation helps prevent buffer overflow attacks.

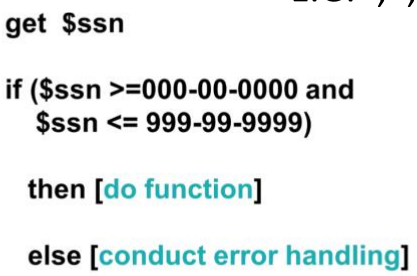
- Pointer Dereference:
  - Programming languages such as C, C++ use pointers, which store a reference to something.
    - E.G: Application has multiple modules. When customer start order application invokes Customer Data module (city, state, zip code etc.). We can pass entire array to module but this consumes memory. Instead we can use a reference to the data array which is simply a pointer to it. This is called pointer deference.
  - Pointer deference: Using pointer to access data array
- DLL Injection
  - Dynamic Link Library (DLL's) are commonly used in applications.
  - DLL Injection is an attack injecting a DLL into a system's memory and causes it to run.
  - Attacker attaches to a running process allocating memory within the running process, connects malicious DLL with allocated memory and executes malicious functions within DLL.

SECURE CODING CONCEPTS

- Compiled Code – Optimized by an application called a compiler. Checks program for errors and provides a report of items for developers to check.
- Runtime code: Code evaluated, interpreted and executed when code is run.
  - E.G: HTML web pages are interpreted at run time.
- Many languages use a cross between compiled and runtime code.
  - E.G: Python
- Input validation – practise of checking data for validity before using it. Prevents attackers from sending malicious code that an

application will use by sanitizing input to remove malicious code or rejecting the input.

- ○ 1. Verify proper characters
- ○ 2. Implement boundary or range checking
- ○ 3. Block HTML code
- ○ 4. Prevent use of certain characters
  - E.G: -,=,’ in SQL attacks

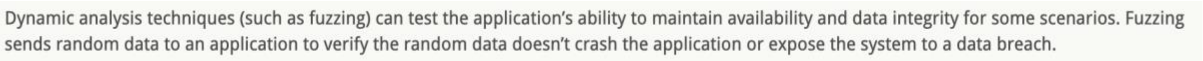


```
get  $ssn

if ($ssn >=000-00-0000 and
   $ssn <= 999-99-9999)

  then [do function]

  else [conduct error handling]
```

- **Remember this:**
  - **Lack of input validation is one of the most common security issues on web-based applications. Input validation verifies the validity of inputted data before using it, and server-side validation is more secure than client-side validation. Input validation protects against many attacks, such as buffer overflow, SQL injection, command injection, and cross-site scripting attacks**
- When 2 or more modules of an application, or two or more applications, attempt to access a resource at the same time this is called a **‘Race Condition.’**
- Error Handling:
  - Error handling ensures an application can handle an error gracefully.
    - Errors to users should be general so detailed errors can provide information that attackers can use against system.
    - Detailed information should be logged including debugging information to make it easier for developers to identify what caused the error and how to resolve.
- **Remember this:**
  - **Error and exception handling helps protect integrity of operating system and controls errors shown to users.**

Applications should show generic error messages to users but log detailed information.

- Sensitive data should be encrypted in applications before storing it, and need to decrypt data before processing it.
    - Certificates can be used to authenticate users and computers.
- Code Signing: provides digital signature for code and certificate includes hash of code. It identifies author and if there are any changes to the code via malware, hash no longer matches, alerting developer.
- Code Reuse and SDKs:
    - Developers are encouraged to reuse code when possible as code is usually gone through internal testing and survived use within application. Brand new-code also takes time to make.
    - Dead code – code that is never executed or used. Logic errors can also create dead code.
- Code obfuscation:
    - Makes code unclear or difficult to understand such as renaming variables, replacing numbers with expressions, replacing strings of characters with hexadecimal codes, removing comments etc.
    - Allows for better security of code although it is possible for someone skilled to dissect code.
- Code Quality and Testing:
    - Static Code Analysis:
        - Source code of an application is reviewed manually or with automatic tools without running the code
    - Dynamic Analysis:
        - Analysis and testing of a program occurs while it is being executed or run
            - E.G: Fuzzing



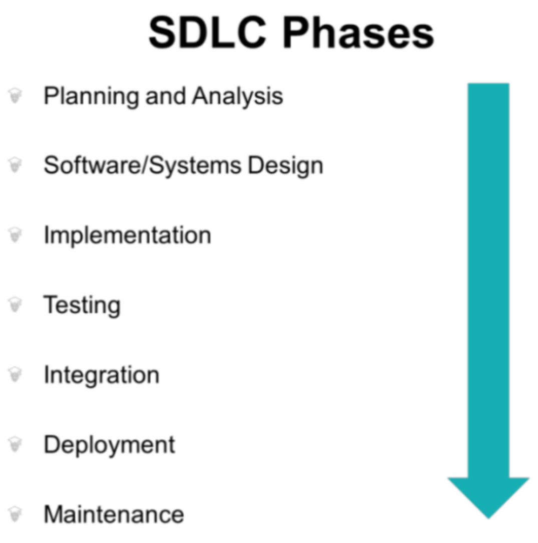
Dynamic analysis techniques (such as fuzzing) can test the application's ability to maintain availability and data integrity for some scenarios. Fuzzing sends random data to an application to verify the random data doesn't crash the application or expose the system to a data breach.

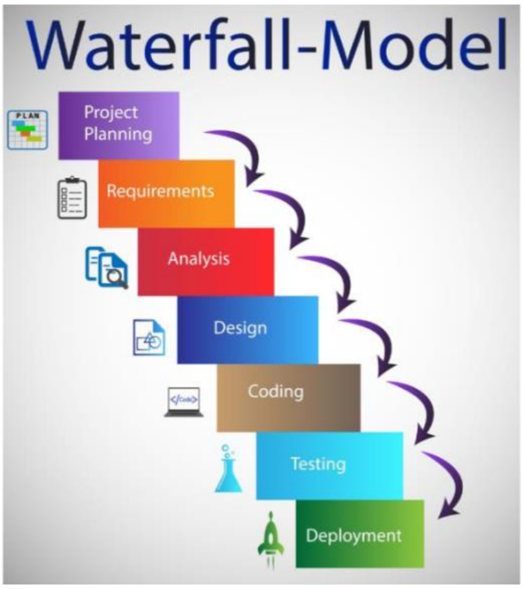
- Fuzzing
    - Injection of randomized data into a software program in an attempt to find system failures, memory leaks, error handling issues, and improper input validation

    - Stress testing: Attempts to simulate live environment and determine how effective or efficient an application operates with a load.
        - E.G: Simulate a DDoS attack and determine impact on web application
    - Sandboxing: Run an application within an isolated environment to test it
    - Model Verification: Testing helps identify and remove bugs.
- **Remember this:**
    - **Static code analysis examines code without running it and dynamic analysis checks code while it is running. Fuzzing techniques send random strings of data to applications to look for vulnerabilities. Stress testing verifies an application can handle a load. Sandboxing runs an application within an isolated environment to test it. Model verification ensures application meets all specifications and fulfils its intended purpose.**

DEVELOPMENT LIFECYCLE [SDLC]

- SDLC – Software development life cycle models attempt to give structure to software development projects
    - Waterfall and Agile





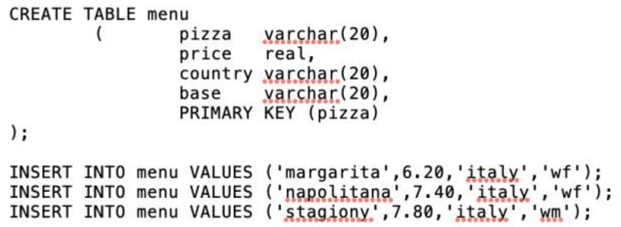
- Waterfall:
    - Multiple stages from top → bottom
    - Each stage feeds on next stage.

        - You cannot go back to a stage after finishing it.
    - Requirements: requirements of what application should do
    - Design: Design of the software architecture. Focuses on structure of project
    - Implementation: Write code based on requirements/design
    - Verification: Verify code is good to go.
    - Maintenance: Maintain changes and update application once required.
    - **Lacks flexibility as it is difficult to revise anything from previous stages. Usually used in big corporations. Agile for smaller.**
- Agile:
    - Set of principles by cross functional teams.
    - Collaborate with customers, respond to change, stress interaction, create working application
    - Uses iterative, time-boxed/ small increment cycles to allow adaptivity to change
    - Testers verify product works with current features then move on to next cycle to develop additional features.
    - Emphasizes interaction between customers, developers and testers during each cycle unlike waterfall model which only does via requirement stage.
- Secure DevOps – Secure Software Development Operations
    - Security Automation – use automated tests to check code. Ensure code doesn't introduce more software bugs/flaws.
    - Continuous integration – Merge code into a central repository. Software is built and tested from the central repository. Includes version control system and supports rolling back code changes when there is a problem.
    - Baselining – Apply changes to baseline code and build code from these changes.
    - Immutable systems cannot be changed – test/create systems in a controlled environment and then deploy it in an immutable system to ensure it stays secure
    - Infrastructure as a code – Manage and provision data centres with code that defines VM's.
- **Remember this:**

    - **SDLC models provide structure for development projects. Waterfall uses multiple stages going from top to bottom, with each stage feeding next stage. Agile is a flexible model that emphasizes interaction with all players in a project. Secure DevOps is an agile-aligned methodology that stresses security throughout lifetime of project.**
- Change Management – Ensures developers don't make unauthorized changes. Several people examine change to ensure there are non-intended consequences. Document changes via accounting structure through version control document.
- Version control – Tracks version of software when updated.
- Provision and Deprovisioning – User accounts.
    - Provisioning: Providing access to an account to various resources.
        - E.G: Provision app to various devices to run on such as on iPhone only. App needs to be provisioned with appropriate code for each device/service.
    - Deprovisioning: Disabling/removing access to resources such as disabling account. Deprovisioning an app means removing it from a device.
- Web servers – Host web sites accessible to internet and can be placed through a DMZ.
    - Apache – Most popular web server
    - IIS (internet information services) – Microsoft web server

DATABASE + SQL

- SQL – Structured Query Language
    - Language used to communicate with databases
- Database – Structured set of data



```
CREATE TABLE menu
        (       pizza   varchar(20),
                price   real,
                country varchar(20),
                base    varchar(20),
                PRIMARY KEY (pizza)
);

INSERT INTO menu VALUES ('margarita',6.20,'italy','wf');
INSERT INTO menu VALUES ('napolitana',7.40,'italy','wf');
INSERT INTO menu VALUES ('stagiony',7.80,'italy','wm');
```

- INT – Integer, VARCHAR – Varial Number of alphanumeric Numerals, TEXT – paragraphs, Decimal – Monetary values
- Normalization:

  - Organizing tables and columns to reduce redundant data and improve overall database performance.
- 1NF (First Normal Form)
  - Each row within a table is unique and identified with a primary key
    - E.G: Author table has primary key of authorID. Book table has primary key BookID. BookAuthor has composite primary key via Book_BookID and Author_AuthorID.
  - Related data is contained in a separate table.
  - None of the columns include repeating groups
    - E.G: Author table has FirstName and LastName. Combining these in a column violates this rule
- 2NF (Second Normal Form)
  - Applies to tables that have a composite primary key where 2 or more columns make up full primary key.
  - It is in 1NF
  - Non-primary key attributes are completely dependent on composite primary key. If any column is dependent on only one column of the composite key, it is not in 2NF.
- 3NF (Third normal form)
  - Helps eliminate unnecessary redundancies within a database.
  - Is in 2NF and 1NF
  - All columns that aren't primary key are only dependent on primary key. None of the columns in the table are dependent on non-primary key attributes.
- **Remember this:**
  - **Normalization is a process used to optimise databases. While there are several normal forms available, a database is considered normalized when it conforms to the first 3 normal forms.**
- Select * FROM Books WHERE Author = 'Darril Gibson'
  - * is wildcard and returns all columns in a table.
- SQL injection:
  - Attack consisting of the insertion or injection of an SQL query via input data from the client to a web application. Attackers

enter additional data into web page form to generate SQL statements.

- Injection Attack:
  - Insertion of additional information or code through data input from a client to an application
    - **SQL**
    - HTML
    - XML
    - LDAP
      - Most common type is an SQL injection
  - E.G:
    - ../../etc/passwd or /etc/passwd to read file in a full directory or rm-rf to remove directory. Use input validation to prevent these attacks.

- **Remember this:**
  - **Attacker use SQL injection attacks to pass queries to back end databases through web servers. Many SQL injection attacks use the phrase ' or '1'='1' to trick the database server into providing information. Input validation and stored procedures reduce risk of SQL injection attacks.**
- Stored procedure – Group of SQL statements that execute as a whole like a program.
  - Parameterized store procedure – accepts data as an input called a parameter. Input is passed to stored procedure as a parameter.

	XSS	CSRF
Full Form	Cross-Site Scripting	Cross-Site Request Forgery
Definition	In XSS, a hacker injects a malicious client side script in a website. This script is added to cause some form of vulnerability to a victim.	It takes advantage of the targeted website's trust in a user. A malicious attack is designed in such a way that a user sends malicious requests to the target website without having knowledge of the attack.
Dependency	Injection of arbitrary data by data that is not validated	On the functionality and features of the browser to retrieve and execute the attack bundle
Requirement of JavaScript	Yes	No
Condition	Acceptance of the malicious code by the sites	Malicious code is located on third party sites
Vulnerability	A site that is vulnerable to XSS attacks is also vulnerable to CSRF attacks	A site that is completely protected from XSS types of attacks is still most likely vulnerable to CSRF attacks.

- **Remember this:**
    - **Cross Site scripting (XSS) attacks allow attackers to capture user information such as cookies. Input validation techniques at server help prevent XSS attacks.**
    - **Cross site request forgery (XSRF) scripting causes users to perform actions on websites such as making purchased without their knowledge. In some cases allows attacker to steal cookies and harvest passwords.**
- Framework – structure used to provide a foundation.
    - Regulatory – Based on relevant laws and regulations
        - HIPAA – Health Insurance Portability and Accountability Act

  - ONC – Office of national coordinator for Health Information Technology
  - OCR – HHS Office for Civil Rights
  - SRA – HIPAA Security Risk Assessment tool
- Non Regulatory – Not required by any law. Identifies common standards and best practices that organizations can follow
  - COBIT – Control Objectives for Information and Related Technologies is used to ensure business goals and IT goals are linked together
- National versus International
  - Some frameworks are used within a single country (national) while others are used internationally
  - NIST – Cybersecurity framework for US
  - ISO and IEC – International Organization for Standardization and International Electrotechnical Commission create/publish international standards
    - ISO/IEC 27002 for international IT security
- Industry specific – Apply to certain industries
  - PCI DSS – Payment Card Industry Data Security Standard for organizations that handle credit cards.