Module 5: Web Security

-by
Dr. Shalu Chopra
Information Technology Department
VESIT

OWASP Top 10 Security Risks

- https://owasp.org/www-project-top-ten/
- https://www.imperva.com/learn/application-security/owasp-top-10/
- https://www.guru99.com/web-security-vulnerabilities.html

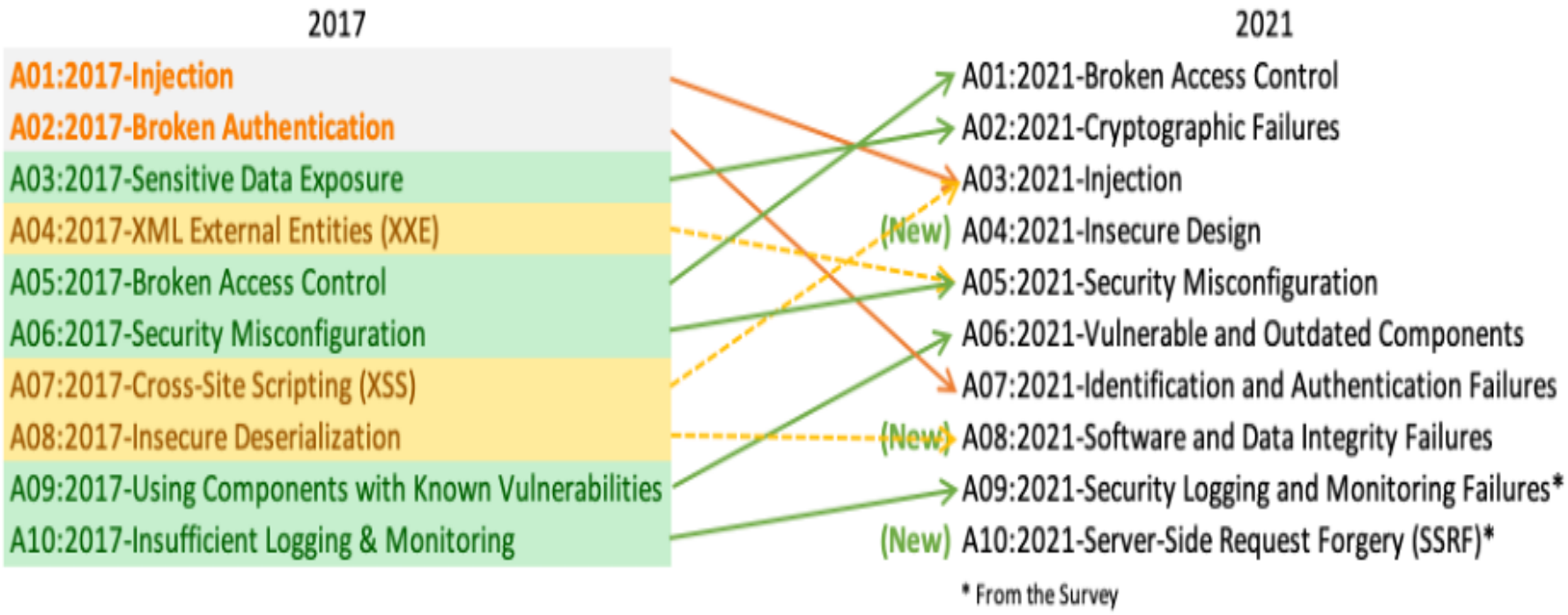
OWASP Security Risk

- The Open Web Application Security Project (OWASP) is a non-profit organization founded in 2001, with the goal of helping website owners and security experts protect web applications from cyber attacks.
- OWASP has 32,000 volunteers around the world who perform security assessments and research.
- OWASP Top 10 is a research project that offers rankings of and remediation advice for the top 10 most serious web application security dangers.
- The risks are graded according to the severity of the vulnerabilities, the frequency of isolated security defects, and the degree of their possible impacts.

OWASP Security Risk

- The OWASP Top 10 is a standard awareness document for developers and web application security.
- It represents a broad consensus about the most critical security risks to web applications.
- Companies should adopt this document and start the process of ensuring that their web applications minimize these risks.

OWASP Security Risk



- **Green arrows** are vulnerabilities that were promoted in importance
- **Orange arrows** are vulnerabilities that were demoted in importance
- **Yellow broken line arrows** are vulnerabilities removed and merged into other categories.

A01:2021—Broken Access Control

- Broken access control means that attackers can gain access to user accounts and act as users or administrators, and that regular users can gain unintended privileged functions.
- Strong access mechanisms ensure that each role has clear and isolated privileges.

Mitigating Broken Access Control

- Deny access by default, except for public resources
- Build strong access control mechanisms and reuse them across the application
- Disable server directory listing and do not store sensitive data in root
- Rate limit API and controller access
- Validate JWT tokens after logout

A02:2021—Cryptographic Failures

- Cryptographic Failures, previously known as Sensitive Data Exposure, covers the protection of data in transit and at rest. This includes passwords, credit card numbers, health records, personal information and other sensitive information.
- It is especially important for organizations covered by standards like PCI Data Security Standards (Payment Card Industry DSS) or data privacy regulations like the EU General Data Protection Regulation (GDPR).

A02:2021—Cryptographic Failures

Mitigating Cryptographic Failures

- Identify sensitive data and apply appropriate security controls.
- Don’t store sensitive data unless absolutely needed - discard sensitive data, use tokenization or truncation.
- Encrypt all sensitive data at rest using strong encryption algorithms, protocols and keys.
- Encrypt data in transit using secure protocols like TLS and HTTP HSTS.
- Disable caching for sensitive data.
- Store passwords using strong, salted hashing functions like Argon2, scrypt and bcrypt.

A03:2021—Injection

- An injection vulnerability in a web application allows attackers to send hostile data to an interpreter, causing that data to be compiled and executed on the server. A common form of injection is SQL injection.

Preventing Injection Attacks

- Use a safe API which avoids the use of the interpreter entirely
- Use positive or “whitelist” server-side input validation
- Escape special characters
- Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.

A04:2021—Insecure Design

- Insecure Design is a category of weaknesses that originate from missing or ineffective security controls.
- Insecure default settings, incomplete configurations, or other issues lead to security vulnerabilities
- Some applications are built without security in mind. By definition, an insecure design cannot be fixed by proper implementation or configuration. This is because it is lacking basic security controls that can effectively protect against important threats.

A04:2021—Insecure Design

Preventing insecure design

- Establish a secure software development lifecycle (SSDLC)
- Leverage application security practices from early stages of software development
- Create a library of secure design patterns, and use it to build new applications
- Leverage threat modeling to design critical features like authentication and access control
- Integrate security concerns and controls into all user stories

A05:2021—Security Misconfiguration

- Security Misconfiguration is a lack of security hardening across the application stack. This can include improper configuration of cloud service permissions, enabling or installing features that are not required, and default admin accounts or passwords. This now also includes XML External Entities (XXE), previously a separate OWASP category.

A05:2021—Security Misconfiguration

Preventing security misconfiguration

- Establish a hardening process for applications, which is fast and easy to deploy
- All systems should have a minimal secure setup without unnecessary features and components
- Configurations should be regularly updated, applying patches and security advisories
- Establish an automated process to verify secure configurations in all environments

A06:2021—Vulnerable and Outdated Components

- Vulnerable and Outdated Components, previously known as “Using Components with Known Vulnerabilities,” includes vulnerabilities resulting from unsupported or outdated software. Anyone who builds or uses an application without knowing its internal components, their versions, and whether they are updated, is exposed to this category of vulnerabilities.

A06:2021—Vulnerable and Outdated Components

Preventing vulnerable and outdated components

- Remove unused dependencies, features, components, and files from applications.
- Maintain an inventory of components and their versions, both on the client side and server side, using software composition analysis (SCA) tools
- Continuously scan libraries and their dependencies for vulnerable components
- Only use components from official sources, and prefer signed packages
- Urgently remediate vulnerabilities, remove affected components, or apply a virtual patch

A07:2021—Identification and Authentication Failures

- Identification and Authentication Failures, previously known as Broken Authentication, this category now also includes security problems related to user identities. Confirming and verifying user identities, and establishing secure session management, is critical to protect against many types of exploits and attacks.

A07:2021—Identification and Authentication Failures

Mitigating Broken Authentication

- Implement multi-factor authentication
- Do not deploy systems with default credentials
- Check for a list of the top 10,000 worst passwords
- Use the guidelines in NIST 800-63 B section 5.1.1 for Memorized Secrets
- Harden all authentication-related processes like registration and credential recovery
- Limit or delay failed login attempts

A08:2021—Software and Data Integrity Failures

- Software and Data Integrity Failures involve code and infrastructure that are vulnerable to integrity violations. This includes software updates, modification of sensitive data, and CI/CD pipeline changes performed without validation. An insecure CI/CD pipeline can lead to unauthorized access, introduction of malware, and other severe vulnerabilities.
- There is a global concern around applications with automatic updates. In several cases, attackers broke into the supply chain and created their own malicious updates. Thousands of organizations were compromised by downloading updates and applying these malicious updates to previously trusted applications, without integrity validation.

A08:2021—Software and Data Integrity Failures

Preventing software and data integrity failures

- Use digital signatures or similar mechanisms to verify software or data is from the expected source and has not been altered.
- Ensure libraries and dependencies, such as npm or maven, are pulling from trusted repositories
- Establish a review process for code and configuration changes
- Ensure that your CI/CD pipeline has proper configuration and access controls

A09:2021—Security Logging and Monitoring Failures

- Security Logging and Monitoring Failures, previously named “Insufficient Logging and Monitoring”, involves weaknesses in an application’s ability to detect security risks and respond to them. Breaches cannot be detected without logging and monitoring. Failures in this cateogry affect visibility, alerting, and forensics.

A09:2021—Security Logging and Monitoring Failures

Preventing security logging and monitoring failures

- Ensure login, access control, and server-side input validation is logged
- Ensure logs contain enough context to identify suspicious behavior and enable in-depth forensic analysis.
- Ensure logs are in a format compatible with log management solutions
- Take measures to prevent attackers from tampering with log data
-

A10:2021—Server Side Request Forgery

- A Server-Side Request Forgery (SSRF) vulnerability occurs when a web application pulls data from a remote resource based on a user-specified URL, without validating the URL. Even servers protected by a firewall, VPN, or network access control list (ACL) can be vulnerable to this attack, if they accept unvalidated URLs as user inputs.

A10:2021—Server Side Request Forgery

Preventing Server Side Request Forgery

- Avoid accepting URLs in client inputs, and if absolutely necessary, sanitize inputs
- Isolate any remote resource access functionality in a separate network to reduce impact
- Use “deny by default” firewall policies to block unwanted Internet traffic
- Use a positive allow list with URL schema, port, and destination
- Disable HTTP redirections
- Never return raw responses to clients

WEB SERVER SECURITY AS PER OWASP

The Top 10 OWASP vulnerabilities in 2020 are:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access control
- Security misconfigurations
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Using Components with known vulnerabilities
- Insufficient logging and monitoring

WEB SECURITY

- Websites and web applications are just as prone to security breaches as physical homes, stores, and government locations.
- Unfortunately, cybercrime happens every day, and great web security measures are needed to protect websites and web applications from becoming compromised.
- That’s exactly what web security does – it is a system of protection measures and protocols that can protect your website or web application from being hacked or entered by unauthorized personnel.

USER AUTHENTICATION

- User authentication is the verification of an active human-to-machine transfer of credentials required for confirmation of a user's authenticity.
- User authentication is performed in almost all human-to-computer interactions other than guest and automatically logged in accounts.
- Authentication authorizes human-to-machine interactions on both wired and wireless networks to enable access to networked and Internet connected systems and resources.

USER AUTHENTICATION ATTACKS

- Credentials sent over HTTP
- Default Passwords
- Passwords Cracked with Brute force or Dictionary Attacks
- Abuse of Reset Forgotten Password Functionality
- Passwords being stored in Local Storage
- Authentication bypass using SQL Injection

TYPES OF AUTHENTICATION

There are three types of Authentication:

- **HTTP Basic Authentication-** In the context of an HTTP transaction, basic access authentication is a method for an HTTP user agent (e.g. a web browser) to provide a user name and password when making a request. In basic HTTP authentication, a request contains a header field in the form of {Authorization: Basic <credentials>}, where credentials is the Base64 encoding of ID and password joined by a single colon :

TYPES OF AUTHENTICATION

- **HTTP Digest Authentication-** Digest Authentication does not require the password to be transmitted. Rather, the client takes the username and password and uses the MD5 hashing algorithm to create a hash, which is then sent to the SQL Server.
- **Form Based Authentication-** HTML Form-based Authentication enables users to supply their user name and password details in an HTML form, and submit them to login to a system. The credentials are submitted by either POST or GET methods.

SESSION MANAGEMENT

- Session management refers to the process of securely handling multiple requests to a web-based application or service from a single user or entity.
- Typically, a session is started when a user authenticates their identity using a password or another authentication protocol.
- Session management involves the sharing of secrets with authenticated users, and as such, secure cryptographic network communications are essential to maintaining session management security.

SESSION MANAGEMENT

Session Management Parameters:

- **Session ID**- it is a unique number that a Web site's server assigns a specific user for the duration of that user's visit (session). The session ID can be stored as a cookie, form field, or URL.
- **Cookies-** Cookies are small pieces of information that are sent in response from the web server to the client. Cookies are the simplest technique used for storing client state on the clients machine.
- **Session Expiry**- Sessions created to be temporary objects subject to expiration

SECURE SOCKET LAYER

- **SSL** stands for Secure Sockets Layer, a global standard security technology that enables encrypted communication between a web browser and a web server.
- SSL certificates are what enable websites to move from HTTP to HTTPS, which is more secure. An SSL certificate is a data file hosted in a website's origin server.
- SSL, more commonly called TLS, is a protocol for encrypting Internet traffic and verifying server identity.
- SSL is essential for protecting your website, even if it doesn't handle sensitive information like credit cards. It provides privacy, critical security and data integrity for both your websites and your users' personal information.

SECURE SOCKET LAYER

Advantages of SSL:

- Security
- Reliability
- SEO
- Software Requirements

Disadvantages of SSL:

- Cost
- Expiry
- Performance

SECURE SOCKET LAYER

Working of SSL

- A user connects to an SSL-enabled service such as a website.
- The user's application requests the server's public key in exchange for its own public key. This public key exchange provides ways for both parties to encrypt messages that only the other party can read.
- When the user sends a message to the server, the application uses the server's public key to encrypt the message.
- The server receives the user's message and decrypts it using its private key. Messages sent back to the browser are encrypted in a similar way using a public key generated by the user's

SECURE SOCKET LAYER

TYPES OF SSL CERTIFICATES:

- **Extended Validation (EV) Certificates** – EV certificates provide the highest level of SSL certification .CAs do extensive background checks on the domain owning organization, validating its ownership, legal existence, physical location, and more.

- A website with an EV cert will turn part or all of the browser web address bar green. The padlock symbol will also be featured, as well as the organization's name.

- PROS: With an EV cert, the green bar and clearly displayed organization name will show users and customers that they should have no doubts about your site's trustworthiness and that you run a legitimate business.

- COMS: An EV SSL cert is very expensive compared to the other options. The

SECURE SOCKET LAYER

- **Organization Validated (PV) Certificates-** The background checks and verification process are more intensive here. CAs verify the individual or business that own the domain and do some minor vetting.
- In the browser address bar, an OV SSL cert is signified in much the same way as DV SSL – with the “https” prefix and a padlock. However, when you click on the padlock it will display more information about that company that owns the domain, such as name, address, and country.
- Pros: OV SSL certificates are considered trustworthy tsince users will know who is behind the website and who they are giving information to.
- Cons: OV certs take longer to issue .Verification can take several days. However, it’s more than likely worth it for your

SECURE SOCKET LAYER

- **Domain Validated (DV) Certificates**- DV SSL certificates have the lowest level of validation of the three. When issuing DV certs, CAs do not look into information about the identity of a person or company running a website.
- They simply verify that they have control over the domain that they are looking to get SSL certified. he web address will feature “https” and the padlock symbol. When you click on the padlock to view the certificate, the information about website ownership will be limited.
- Pros: They are issued more quickly than the other options due to the less rigorous verification process, which is generally online and automated. Most of the time it is issued on the same day.

HTTPS

- HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.
- The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.
- Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website.
- HTTPS is encrypted in order to increase security of data transfer.

HTTPS

- HTTPS prevents websites from having their information broadcast in a way that's easily viewed by anyone snooping on the network.
- When information is sent over regular HTTP, the information is broken into packets of data that can be easily "sniffed" using free software.
- This makes communication over the an unsecure medium, such as public Wi-Fi, highly vulnerable to interception.
- With HTTPS, traffic is encrypted such that even if the packets are sniffed or otherwise intercepted, they will come across as nonsensical characters

HTTPS

HTTP Methods:

- **GET-**The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
- **HEAD-**Same as GET, but transfers the status line and header section only.
- **POST-**A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
- **PUT-**Replaces all current representations of the target resource with the uploaded content.

HTTP 1.1

- The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.
- This protocol includes more stringent requirements than HTTP/1.0 in order to ensure reliable implementation of its features.
- In HTTP/1.0, most implementations used a new connection for each request/response exchange.
- In HTTP/1.1, a connection may be used for one or more request/response exchanges, although connections may be closed for a variety of reasons

HTTP 2

- HTTP/2 is the first upgrade to the Hypertext Transfer Protocol since 1999.
- It's goal is to improve website performance by optimizing how HTTP is expressed "on-the-wire."
- It doesn't change the semantics of HTTP, which means header fields, status codes, and cookies work exactly the same way as in HTTP/1.1.
- HTTP 2 supports features like Multiplexing, Header Compression, Stream Priority etc. all designed to improve page load times for website visitors.

SECURE SOCKET SHELL (SSH)

- **SSH(Secure Shell)** is a cryptographic network protocol that is used for transferring encrypted data over network.
- It allows you to connect to a server, or multiple servers, without having you to remember or enter your password for each system that is to login remotely from one system into another.
- It works on three-stage process for server and client authentication processes.
- It has username/password authentication system.
- It is appropriate and effective for securely executing commands across the internet.

SECURE SOCKET SHELL (SSH)

- When you connect through SSH, you will be dropped into a shell session, which is a text-based interface where you can interact with your server.
- For the duration of your SSH session, any commands that you type into your local terminal are sent through an encrypted SSH tunnel and executed on your server.
- The SSH connection is implemented using a client-server model.
- This means that for an SSH connection to be established, the remote machine must be running a piece of software called an SSH daemon.

SECURE SOCKET SHELL (SSH)

- This software listens for connections on a specific network port, authenticates connection requests, and spawns the appropriate environment if the user provides the correct credentials.
- The user's computer must have an SSH client.
- This is a piece of software that knows how to communicate using the SSH protocol and can be given information about the remote host to connect to, the username to use, and the credentials that should be passed to authenticate.
- The client can also specify certain details about the connection type they would like to establish.

PRIVACY ON THE WEB

- Privacy on the web refers to the vast range of technologies, protocols and concepts related to giving individual users or other parties more privacy protections in their use of the global Internet.
- Internet privacy and anonymity are paramount to users, especially as e-commerce continues to gain traction. For example, age and physical address alone could identify who an individual is without explicitly disclosing their name, as these two factors are unique enough to identify a specific person typically.
- Other forms may soon include GPS tracking data used by apps, as the daily commute and routine information can be enough to identify an individual.
- So consider a scenario where you are continously eyed by cookies, spams, search histories, etc. These all compromise your Privacy at

WEB BROWSER ATTACKS

- Browsers are integral to an effective working environment but they also serve as the perfect cyberattack vector.
- Web-based attacks are one of the top methods of system compromise and they are on the rise.
- Web-based threats leverage browsers and their extensions, websites, content management systems and IT components of web services and applications to harvest credentials, skim visitor payment details or infect systems with malware or ransomware .

WEB BROWSER ATTACKS

- Of particular danger to organizations are fileless attacks that take advantage of browser third-party plug-ins like JavaScript, Flash, and ActiveX, as there are no links or files for security systems to detect and behavioural monitoring always leaves some window of exposure.
- Common types of web browser attacks are Drive-by Downloads, Plugins & Extensions, Clickjacking, Man in the Browser attacks etc.

WEB BUGS

- A Web bug, also known as a Web beacon, is a file object that is placed on a Web page or in an e-mail message to monitor user behavior.
- Unlike a cookie, which can be accepted or declined by a browser user, a Web bug arrives as just another GIF or other file object. It can usually only be detected if the user looks at the source version of the page to find a tag that loads from a different Web server than the rest of the page.
- Web bugs are often used by spammers to validate e-mail addresses. When a recipient opens an email message that includes a Web bug, information returned to the sender

WEB BUGS

A Web bug can gather the following statistics:

- The IP address of the computer that fetched the Web bug.
- The URL of the page that the Web bug is located on.
- The URL of the Web bug.
- The time the Web bug was viewed.
- The type of browser that fetched the Web bug.
- A previously set cookie value.

CLICKJACKING

- Clickjacking, also known as a “UI redress attack”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page.
- Thus, the attacker is “hijacking” clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.
- Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

CROSS SITE SCRIPTING FORGERY

- Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform.
- It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other.
- In a successful CSRF attack, the attacker causes the victim user to carry out an action unintentionally.
- For example, this might be to change the email address on their account, to change their password, or to make a funds transfer. Depending on the nature of the action, the attacker might be able to gain full control over the user's account.

CROSS SITE SCRIPTING FORGERY

For a CSRF attack to be possible, three key conditions must be in place:

- **A relevant action-**There is an action within the application that the attacker has a reason to induce. This might be a privileged action or any action on user-specific data.
- **Cookie-based session handling-** Performing the action involves issuing one or more HTTP requests, and the application relies solely on session cookies to identify the user who has made the requests. There is no other mechanism in place for tracking sessions or validating user requests.
- **No unpredictable request parameters**- The requests that perform the action do not contain any parameters whose values the attacker cannot determine or guess.

SESSION HIJACKING

- Session hijacking is defined as taking over an active TCP/IP communication session without the user's permission.
- When implemented successfully, attackers assume the identity of the compromised user, enjoying the same access to resources as the compromised user.
- Identity theft, Information theft, stealing sensitive data are some of the common impacts of session hijacking.
- Encryption, Use of a long random number or string as the session key, Regenerating the session id after a successful login could help prevent session hijacking

DNS ATTACKS

- A DNS Attack is any attack targeting the availability or stability of a network's DNS service.
- Attacks that leverage DNS as its mechanism as part of its overall attack strategy, such as cache poisoning, are also considered DNS attacks.
- DNS attacks are any type of attack that involves the domain name system (DNS).
- There are many different ways that attackers can take advantage of weaknesses in the DNS.
- Most of these attacks are focused on abusing the DNS to stop internet users from being able to access certain websites

DNS ATTACKS

- The DNS system is complicated, and attackers can take advantage of it in a range of different ways, with various end goals in mind. Many of these attacks aim to make websites, networks or machines unreachable. Different types of DNS attacks are:
- DNS Flood
- DNS Cache Poisoning
- DNS Poisoning/ DNS Hijacking
- DNS Spoofing

SECURE ELECTRONIC TRANSFER

- Secure electronic transaction (SET) was an early communications protocol used by e-commerce websites to secure electronic debit and credit card payments.
- Secure electronic transaction was used to facilitate the secure transmission of consumer card information via electronic portals on the Internet.
- Secure electronic transaction protocols were responsible for blocking out the personal details of card information, thus preventing merchants, hackers, and electronic thieves from accessing consumer information.

SECURE ELECTRONIC TRANSFER

Requirements in SET : SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is intended user or not and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.
- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of best security mechanisms.

EMAIL ATTACKS

- Email is frequently an intruder's gateway into an organization.
- An email attack occurs when email is used as an attempt to cause damage or harm to either an individual or an organization.
- Although the mechanisms of email-based attacks vary, the objective is almost always the same: steal money or data.
- Email Security refers to the security measures that an organization takes in order to secure various aspects of its email system such as identity, content, media

EMAIL ATTACKS

- **Phishing:** Phishing is a fraudulent attempt to obtain sensitive information such as usernames, passwords and credit card details by pretending to be a trusted entity. Phishing attacks are sent in high volume, and the legitimate look of the email can trick users into accidentally opening an attachment or clicking on a malicious link.
- **Spear Phishing:** Spear phishing. Spear phishing is an advanced phishing attack that is targeted at one or a few individuals. Before the attack is launched, the attacker spends time researching their target to gain information

EMAIL ATTACKS

- **Vishing:** It is a social engineering attack that attempts to trick victims into giving up sensitive information over the phone. In most cases, the attacker strategically manipulates human emotions, such as fear, sympathy, and greed in order to accomplish their goals.
- **Smishing**: It is a cyberattack that uses misleading text messages to deceive victims. The goal is to trick you into believing that a message has arrived from a trusted person or organization, and then convincing you to take action that gives the attacker exploitable information.

EMAIL ATTACKS

- **Whaling:** Whaling attacks use fraudulent emails that appear to be from trusted sources to try to trick victims into divulging sensitive data over email or visiting a spoofed website that mimics that of a legitimate business and asks for sensitive information such as payment or account details. Whaling emails and websites are highly personalized towards their targets and often include targets' names, job titles, and basic details to make the communications look as legitimate as possible.
- **Spam:** Spam is known as a high volume commercial messaging sent over email.Despite several tools to filter out unwanted email, spam remains a significant challenge for organizations large and small.

PHISHING

- Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. Different techniques for Phishing are:
  - Spear Phishing
  - Session Hijacking
  - Email Spam
  - Content Injection
  - Web Based Delivery
  - Phishing through Search Engines
  - Vishing and Smishing
  - Link Manipulation

PHARMING

- Pharming is a type of cyberattack involving the redirection of web traffic from a legitimate site to a fake site for the purpose of stealing usernames, passwords, financial data, and other personal information.
- Pharming is a sophisticated kind of phishing attack and it can affect anyone on any platform.
- Pharming can be conducted either by changing the hosts file on a victim's computer or by exploitation of a vulnerability in DNS server software.
- The term "pharming" is a neologism based on the words "farming" and "phishing".

PHARMING

Two types of Pharming attacks are:

- **Pharming malware** aka DNS changers/hijackers infect a victim's computer and stealthily make changes to the victim's hosts file.
- **DNS poisoning**- also know as DNS spoofing takes advantage of exploits in the software that controls DNS servers in order to hijack the servers and reroute web traffic. Typically, DNS poisoning goes after the companies that run and maintain the DNS servers that translate human-friendly domain names into computer ready IP

WEB SERVICE SECURITY

Web services security includes several aspects:

- **Authentication**—Verifying that the user is who she claims to be. A user's identity is verified based on the credentials presented by that user, such as:
  - Something one has, for example, credentials issued by a trusted authority such as a passport (real world) or a smart card (IT world).
  - Something one knows, for example, a shared secret such as a password.
  - Something one is, for example, biometric information.
- **Authorization (or Access Control)**-Granting access to specific resources based on an authenticated user's entitlements.

WEB SERVICE SECURITY

- **Confidentiality, privacy**—Keeping information secret. Accesses a message, for example a Web service request or an email, as well as the identity of the sending and receiving parties in a confidential manner. Confidentiality and privacy can be achieved by encrypting the content of a message and obfuscating the sending and receiving parties' identities.
- **Integrity, non repudiation**—Making sure that a message remains unaltered during transit by having the sender digitally sign the message. A digital signature is used to validate the signature and provides non-repudiation. The timestamp in the signature prevents anyone from replaying this message after the expiration

WEB SERVICE SECURITY

Web services security requirements are supported by industry standards both at the transport level and at the application level relying on XML frameworks.

- **Transport-level Security-** Secure Socket Layer (SSL), otherwise known as Transport Layer Security (TLS), the Internet Engineering Task Force (IETF) officially standardized version of SSL, is the most widely used transport-level data-communication protocol providing:

- Authentication (the communication is established between two trusted parties).
- Confidentiality (the data exchanged is encrypted).
- Message integrity (the data is checked for possible corruption).
- Secure key exchange between client and server.
- SSL provides a secure communication channel, however, when the data is not "in transit," the data is not protected. This makes the environment vulnerable to attacks in multi-step transactions. (SSL provides point-to-point security, as

PENETRATION TESING

- Penetration testing (or pen testing) is a security exercise where a cyber-security expert attempts to find and exploit vulnerabilities in a computer system.
- The purpose of this simulated attack is to identify any weak spots in a system's defenses which attackers could take advantage of.
- This is like a bank hiring someone to dress as a burglar and try to break into their building and gain access to the vault. If the 'burglar' succeeds and gets into the bank or the vault, the bank will gain valuable information on how they need to tighten their security measures.

PENETRATION TESING

Penetration Testing Stages:

- Planning and Reconnaissance
- Scanning
- Gaining Access
- Analysis

PENETRATION TESING

Methods of Penetration Testing:

- External Testing
- Internal Testing
- Blind Testing
- Double Testing
- Targeted Testing

PENETRATION TESING

Categories of Penetration Testing:

- Black Box Testing
- White Box Testing
- Grey Box Testing

Frameworks used for Penetration Testing:

- OSSTMM
- NIST

THANK YOU