# ETHICAL HACKING

(USCS607)

Smitha Renny

# UNIT - I

- **Information Security : Attacks and Vulnerabilities**
  - **Introduction to information security :** Asset, Access Control, CIA, Authentication, Authorization, Risk, Threat, Vulnerability, Attack, Attack Surface, Malware, Security-Functionality-Ease of Use Triangle
  - **Types of malware :** Worms, viruses, Trojans, Spyware, Rootkits
- **Types of vulnerabilities** : OWASP Top 10 : cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, XML External Entities, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, OWASP Mobile Top 10, CVE Database
- **Types of attacks and their common prevention mechanisms :** Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking, Clickjacking, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, IoT Attacks, BOTs and BOTNETs
- **Case-studies** : Recent attacks – Yahoo, Adult Friend Finder, eBay, Equifax, WannaCry, Target Stores, Uber, JP Morgan Chase, Bad Rabbit

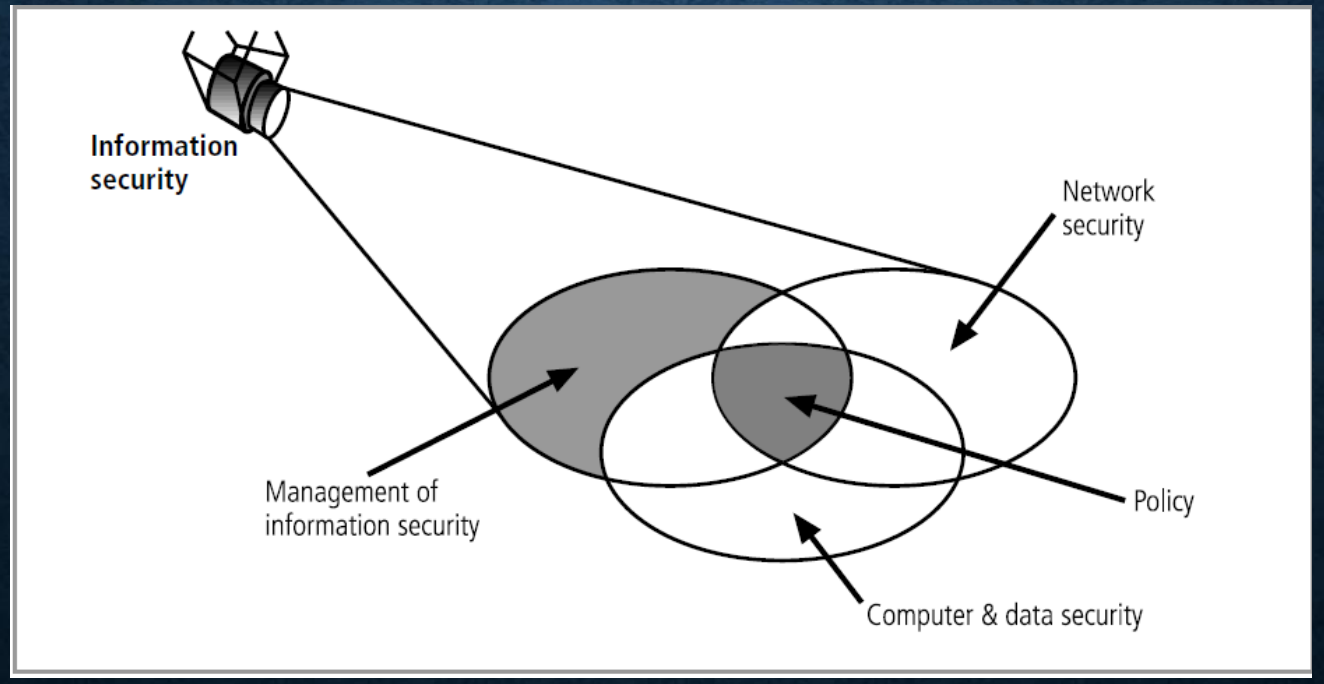
# INFORMATION SECURITY : ATTACKS AND VULNERABILITIES

## INTRODUCTION TO INFORMATION SECURITY :

### What Is Security?

- In general, **security** is "the quality or state of being secure—to be free from danger." In other words, protection against adversaries—from those who would do harm, intentionally or otherwise—is the objective.
- A successful organization should have the following multiple layers of security in place to protect its operations:
  1. **Physical security**: to protect physical items, objects, or areas from unauthorized access and misuse
  2. **Personnel security**: to protect the individual or group of individuals who are authorized to access the organization and its operations
  3. **Operations security:** to protect the details of a particular operation or series of activities
  4. **Communications security**: to protect communications media, technology, and content
  5. **Network security:** to protect networking components, connections, and contents
  6. **Information security**: to protect the confidentiality, integrity and availability of information assets, whether in storage, processing, or transmission. It is achieved via the application of policy, education, training and awareness, and technology.

- The Committee on National Security Systems (CNSS) defines information security as the protection of information and its critical elements, including the systems and hardware that use, store, and transmit that information.
- The CNSS model of information security evolved from a concept developed by the computer security industry called the C.I.A. triangle.
- The **C.I.A. triangle** has been the industry standard for computer security since the development of the mainframe. It is based on the three characteristics of information that give it value to organizations: **confidentiality, integrity, and availability**.




Components of Information Security

# ASSET

- In information security, computer security and network security, an **asset** is any data, device, or other component of the environment that supports information-related activities.
- Assets generally include hardware (e.g. servers and switches), software (e.g. mission critical applications and support systems) and confidential information.
- Assets should be protected from illicit access, use, disclosure, alteration, destruction, and/or theft, resulting in loss to the organization

# TYPES OF ASSETS

The major steps required for asset classification and controls are:

A. Identification of the assets

B. Accountability of assets

C. Preparing a schema for information classification

D. Implementing the classification schema

http://www.networkmagazineindia.com/200212/security2.shtml

# A. IDENTIFICATION OF ASSETS

We can broadly classify assets in the following categories:

1. **Information assets**
2. **Software assets**
3. **Physical assets**
4. **Services**

http://www.networkmagazineindia.com/200212/security2.shtml

# B. ACCOUNTABILITY OF ASSETS

- Establishing ownership for the information assets
- Identify the actual owners of each of the information assets
- Identifying owners of the application software
- Assets valuation

# C. PREPARING A SCHEMA FOR CLASSIFICATION

- The criteria for the classification of assets could be:

1. Confidentiality: Can the information be freely distributed? Or do we need to restrict it to certain identified individuals?
2. Value: What is the asset value? Is it a high value item, costly to replace or a low value item?
3. Time: Is the information time sensitive? Will its confidentiality status change after some time?
4. Access rights: Who will have access to the asset?
5. Destruction: How long the information will be stored? How can it be destroyed, if necessary?

# D. IMPLEMENTATION OF THE CLASSIFICATION SCHEMA

- A company's business plan is a confidential document.
- The plan will be discussed behind closed doors, known to only a few senior members.
- In the next step the final plan will be prepared and stored on the MD's computer or that of his secretary. A soft copy of this plan would be sent by email to all executives who need to refer to it. The hard disk of every computer where the plan is stored will also have a backup copy on other media.
- So the plan is now distributed across the organization, available on the hard disks of computers belonging to each secretary and each senior executive.

# ACCESS CONTROL

- *Access controls* are security features that control how users and systems communicate and interact with other systems and resources.
- *Access* is the flow of information between a subject and a resource.
- A *subject* is an active entity that requests access to a resource or the data within a resource. E.g.: user, program, process etc.
- A *resource* is an entity that contains the information. E.g.: Computer, Database, File, Program, Printer etc.
- Access controls give organization the ability to control, restrict, monitor, and protect resource availability, integrity and confidentiality
- Access controls can be implemented at various layers of a network and individual systems.
- The access controls can be classified into three categories based on mechanisms that can be carried out manually or automatically.
- Administrative Controls
- Physical Controls
- Technical or Logical Controls

# ACCESS CONTROL TYPES

Each of the access control categories – administrative, physical and technical work at different levels, each at a different level of granularity and perform different functionalities based on the type. The different types of access control are

- Preventative- Avoid undesirable events from occurring
- Detective- Identify undesirable events that have occurred
- Corrective- Correct undesirable events that have occurred
- Deterrent- Discourage security violations
- Recovery- Restore resources and capabilities
- Compensative- Provide alternatives to other controls

# ACCESS CONTROL THREATS

- Denial of Service(DoS/DDoS)
- Buffer Overflows
- Malicious Software
- Password Crackers
- Spoofing/Masquerading
- Emanations
- Shoulder Surfing
- Object Reuse
- Data Remanence
- Backdoor/Trapdoor
- Dictionary Attacks
- Brute-force Attacks
- Social Engineering

# AUTHENTICATION

- Authentication is the process of determining whether someone or something is, in fact, who or what it declares itself to be.
- Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server.
- Users are usually identified with a user ID, and authentication is accomplished when the user provides a credential, for example a password, that matches with that user ID

Authentication is followed by authorization

# AUTHORIZATION

- Authorization is the process of giving someone permission to do or have something.
- In multi-user computer systems, a system administrator defines for the system which users are allowed access to the system and what privileges of use (such as access to which file directories, hours of access, amount of allocated storage space, and so forth).
- Assuming that someone has logged in to a computer operating system or application, the system or application may want to identify what resources the user can be given during this session.

Logically, authorization is preceded by authentication.

# CONFIDENTIALITY, INTEGRITY, AND AVAILABILITY (CIA TRIAD)


**Confidentiality, integrity and availability**, also known as the **CIA triad**, is a model designed to guide policies for information security within an organization.

In this context, confidentiality is a set of rules that limits access to information, integrity is the assurance that the information is trustworthy and accurate, and availability is a guarantee of reliable access to the information by authorized people.

# CONFIDENTIALITY

- Confidentiality is roughly equivalent to privacy.
- Measures undertaken to ensure confidentiality are designed to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it.
- Access must be restricted to those authorized to view the data in question.
- Data can also be categorized according to the amount and type of damage that could be done if it falls into unintended hands. More or less stringent measures can then be implemented according to those categories.

# INTEGRITY

- Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle .
- Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorized people (for example, in a breach of confidentiality). These measures include file permissions and user access controls.
- Version control may be used to prevent erroneous changes or accidental deletion by authorized users becoming a problem.
- Some means must be in place to detect any changes in data that might occur as a result of non-human-caused events such as an electromagnetic pulse (EMP) or server crash.
- Some data might include checksums, even cryptographic checksums, for verification of integrity. Backups or redundancies must be available to restore the affected data to its correct state.

# AVAILABILITY

- Availability is ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed and maintaining a correctly functioning operating system environment that is free of software conflicts.
- It's also important to keep current with all necessary system upgrades. Providing adequate communication bandwidth and preventing the occurrence of bottlenecks are equally important.
- Fast and adaptive disaster recovery is essential for the worst case scenarios.
- Safeguards against data loss or interruptions in connections must include unpredictable events such as natural disasters and fire.
- To prevent data loss from such occurrences, a backup copy may be stored in a geographically-isolated location, perhaps even in a fireproof, waterproof safe. Extra security equipment or software such as firewalls and proxy servers can guard against downtime and unreachable data due to malicious actions such as denial-of-service (DoS) attacks and network intrusions.

**Any of the following break the CIA triad / (anti-CIA triad):**

- **Disclosure** is the inadvertent, accidental, or malicious revealing or allowing access of information or resources to an outside party. If you are not authorized to have access to an object, you should never have access to it.
- **Alteration** is the counter to integrity; it deals with the unauthorized modification of information. This modification can be caused by corruption, accidental access that leads to modification, or modifications that are malicious in nature.
- **Disruption** (also known as loss) means that authorized access to information or resources has been lost. Information is useless if it is not there when it is needed. Although information or other resources can never be 100 percent available, some organizations spend the time and money to ensure 99.999 percent uptime for critical systems, which averages about six minutes of downtime per year.

# THREAT

- A *threat* is any agent, circumstance, or situation that has the potential to cause harm or loss to an IT asset. Threats can take on many forms, and may not always be readily identifiable. A *threat* is an environment or situation that could lead to a potential breach of security.
- There are various threats available, system threats, Network threats, application threats, cloud threats, malicious files threats etc. Additionally bad weather, hurricane, tornado, flood, or earthquake could cause just as much damage to assets as a hacker could ever.
- Ethical hackers are, much more concerned with the virtual threat agent techniques, but security professionals designing an entire program need to be aware of as many threats as possible.
- A threat sets the stage for risk and is any agent, condition, or circumstance that could potentially cause harm, loss, or damage, or compromise an IT asset or data asset.
- From a security professional's perspective, threats can be categorized as events that can affect the **confidentiality**, **integrity**, or **availability** of the organization's assets.
- These threats can result in destruction, disclosure, modification, corruption of data, or denial of service.

Examples of the types of threats an organization can face include the following:

- **Natural disasters, weather, and catastrophic damage:** Hurricanes, such as Ockhi(which hit Lakshwadeep Islands and Tamil Nadu in 2018), storms, weather outages, fire, flood, earthquakes, and other natural events compose an ongoing threat.
- **Hacker attacks:** An insider or outsider who is unauthorized and purposely attacks an organization's components, systems, or data.
- **Cyber-attack:** Attackers who target critical national infrastructure such as water plants, electric plants, gas plants, oil refineries, gasoline refineries, nuclear power plants, waste management plants, and so on. Stuxnet is an example of one such tool designed for just such a purpose.
- **Viruses and malware:** An entire category of software tools that are malicious and are designed to damage or destroy a system or data. Cryptowall and Sality are two example of malware.
- **Disclosure of confidential information:** Anytime a disclosure of confidential information occurs, it can be a critical threat to an organization if that disclosure causes loss of revenue, causes potential liabilities, or provides a competitive advantage to an adversary.

# VULNERABILITY

- A *vulnerability* is any weakness, such as a software flaw or logic design, that could be exploited by a threat to cause damage to an asset.
- The goal of penetration testers is to discover these vulnerabilities and attempt to exploit them.
- The existence of vulnerabilities does not necessarily equate to a risk.
- For example, given physical access to any computer system, a hacker could easily (usually) successfully hack the device—so the vulnerability (physical access) exists. However, if your server is locked in an airtight room, buried in an underground silo, with multiple guards and physical security measures in place, the probability of it being exploited is reduced to near zero.

- A *vulnerability* is an existence of a software flaw, logic design, or implementation error that can lead to an unexpected and undesirable event executing bad or damaging instructions to the system.
- A vulnerability is a weakness in the system design, implementation, software, or code, or the lack of a mechanism.
- Vulnerabilities can be found in each of the following:
- **Applications:** Software and applications come with tons of functionality. Applications may be configured for usability rather than for security. Applications may be in need of a patch or update that may or may not be available. Attackers targeting applications have a target-rich environment to examine. Just think of all the applications running on your home or work computer.
- **Operating systems:** This operating system software is loaded in workstations and servers. Attacks can search for vulnerabilities in operating systems that have not been patched or updated.
- **Misconfiguration:** The configuration file and configuration setup for the device or software may be misconfigured or may be deployed in an unsecure state. This might be open ports, vulnerable services, or misconfigured network devices. Just consider wireless networking. Can you detect any wireless devices in your neighborhood that have encryption turned off?
- **Shrinkwrap software:** The application or executable file that is run on a workstation or server. When installed on a device, it can have tons of functionality or sample scripts or code available.

# ATTACK

- The method followed by a hacker/Individual to break into the system. Denial of service attack, Misconfiguration attacks, Operating system attacks, Virus, and Worms are all example of Attacks.
- An attack is an action that is done on a system to get its access and extract sensitive data.
- An *attack* occurs when a system is compromised based on a vulnerability. Many attacks are perpetuated via an exploit.
- Ethical hackers use tools to find systems that may be vulnerable to an exploit because of the operating system, network configuration, or applications installed on the systems, and prevent an attack

Most hacking tools exploit weaknesses in one of the following four areas:

1. **Operating systems:**
   - Many systems administrators install operating systems with the default settings, resulting in potential vulnerabilities that remain unpatched.
   - Administrator accounts with no passwords, all ports left open, and guest accounts, are examples of settings the installer may forget about.
   - Additionally, operating systems are never released fully secure—they can't be, if you ever plan on releasing them within a timeframe of actual use—so the potential for an old vulnerability in newly installed operating systems is always a plus for the ethical hacker.

2. **Applications:**
   - Applications usually aren't tested for vulnerabilities when developers are writing the code, which can leave many programming flaws that a hacker can exploit.
   - These are attacks on the actual programming codes of an application.
   - Many applications on a network aren't tested for vulnerabilities as part of their creation and, have many vulnerabilities built into them. Applications on a network are a goldmine for most hackers.

3. **Shrink-wrap code:**
    - Many off-the-shelf programs come with extra features the common user isn't aware of, which can be used to exploit the system. One example is macros in Microsoft Word, which can allow a hacker to execute programs from within the application.
    - These attacks take advantage of the built-in code and scripts most off-the-shelf applications come with.
    - The old refrain "Why reinvent the wheel?" is very often used to describe this attack type. Why spend time writing code to attack something when you can buy it already "shrink wrapped"? These scripts and code pieces are designed to make installation and administration easier, but can lead to vulnerabilities if not managed appropriately.

4. **Misconfigurations:**
    - Systems can also be misconfigured or left at the lowest common security settings to increase ease of use for the user, which may result in vulnerability and an attack.
    - These attacks take advantage of systems that are not configured appropriately for security.
    - This type of attack takes advantage of the administrator who simply wants to make things as easy as possible for the users. Perhaps to do so, the admin will leave security settings at the lowest possible level, enable every service, and open all firewall ports. It's easier for the users, but creates another goldmine for the hacker.

- Attacks can be categorized as either ***passive* or *active***. Passive and active attacks are used on both network security infrastructures and on hosts.
- **Active attacks** actually alter the system or network they're attacking, whereas passive attacks attempt to gain information from the system.
- Active attacks affect the availability, integrity, and authenticity of data;
- **Passive attacks** are breaches of confidentiality.
- In addition to the active and passive categories, attacks are categorized as either ***inside* or *outside*** attacks.
- An attack originating from within the security perimeter of an organization is an inside attack and usually is caused by an "insider" who gains access to more resources than expected.
- An outside attack originates from a source outside the security perimeter, such as the Internet or a remote access connection.

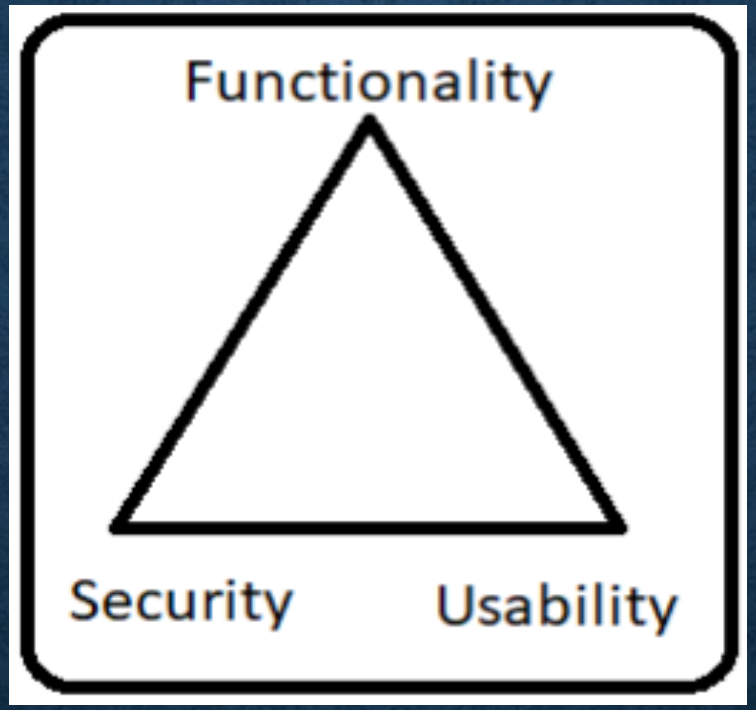
# ATTACK SURFACE

- An attack surface is the total sum of the vulnerabilities in a given computing device or network that are accessible to a hacker.
- Anyone trying to break into a system generally starts by scanning the target's attack surface for possible attack vectors, whether for an active attack or passive attack, ethical hacking or a hacking competition.
- Attack surfaces can be divided in to a few categories:
    1. The network attack surface.
    2. The software attack surface.
    3. The physical attack surface

- Every point of network interaction is a potential part of the **network attack surface**. A network attack surface can be reduced by closing unnecessarily open ports and limiting the resources that are available to untrusted users and to the Internet in general, through methods like MAC address filtering.
- As all running code has the possibility of having exploitable vulnerabilities, one of the first and simplest ways to limit **software attack surface** is to reduce the amount of running code. The more a piece of malware can use various exploits, the more chance it can get in via a hole in a target system's attack surface.
- **Physical access** also constitutes an **attack surface**. This surface is exploitable by inside vectors such as rogue employees or hired workers. External risks include password retrieval from carelessly discarded hardware or from password sticky notes.

- Best practices for physical attack surface remediation include enforcing strong authentication, destroying hard drives before throwing them out and refraining from leaving hard copy access data -- like sticky note passwords – in proximity to a computer.
- Knowledge of all elements of an organization's attack surface is crucial to proper setup of breach detection systems (BDS), firewalls, intrusion prevention systems, data policies and other security measures.

# SECURITY, FUNCTIONALITY AND USABILITY TRIANGLE




There is an inter dependency between these three attributes. When security goes up, usability and functionality come down. Any organization should balance between these three qualities to arrive at a balanced information system.

# TYPES OF MALWARE

- Malware is short for ***mal***icious soft***ware***, meaning software that can be used to compromise computer functions, steal data, bypass access controls, or otherwise cause harm to the host computer.
- Malware is a broad term that refers to a variety of malicious programs. There are several common types of malware; rootkits, spyware, Trojan horses, viruses, and worms etc.

# VIRUS

- A virus represents the oldest form of malware and is by far the best known to the public. A virus is a form of malware that is capable of copying itself and spreading to other computers. A virus is a self-replicating application that attaches itself to other executable programs. Many viruses affect the host as soon as they are executed; others lie in wait, dormant, until a predetermined event or time, before carrying out their instructions.
- Many potential actions can take place, like, altering data, infecting other programs, replicating, encrypting itself, transforming itself into another form, altering configuration settings, destroying data, corrupting or destroying hardware
- Viruses can be used to steal information, harm host computers and networks, create botnets, steal money, render advertisements, and more.

- The process of developing a virus is very methodical. It occurs in six steps:

1. *Design* —The author envisions and creates the virus. The author may choose to create the virus completely from scratch or use one of the many construction kits that are available to create the virus of their choice.
2. R*eplication* —Once deployed, the new virus spreads through replication: multiplying and then ultimately spreading to different systems. How this process takes place depends on the author's original intent, but the process can be very rapid, with new systems becoming infected in short order.
3. *Launch* —The virus starts to do its work by carrying out the task for which it was created (such as destroying data or changing a system's settings). Once the virus activates through a user action or other predetermined action, the infection begins.
4. *Detection* —The virus is recognized as such after infecting systems for some period of time. During this phase, the nature of the infection is typically reported to antivirus makers, who begin their initial research into how the software works and how to eradicate it.

5. *Incorporation* —The antivirus makers determine a way to identify the virus and incorporate the process into their products through updates. Typically, the newly identified malware is incorporated into signature files, which are downloaded and installed by the antivirus application.

6. *Elimination* —Users of the antivirus products incorporate the updates into their systems and eliminate the virus.

**It is important to realize that this process is not linear: It is a loop or cycle. When step 6 is reached, the whole process starts over at step 1 with another round of virus development.**

# KINDS OF VIRUSES

Modern viruses come in many varieties:

- A *system* or *boot sector virus* is designed to infect and place its own code into the master boot record (MBR) of a system.
- *Macro viruses* debuted in force around 2000. They take advantage of embedded languages such as Visual Basic for Applications (VBA).
- *Cluster viruses* are another variation of the family tree that carries out its dirty work in yet another original way. This virus alters the file-allocation tables on a storage device, causing file entries to point to the virus instead of the real file.
- A *stealth* or *tunneling virus* is designed to employ various mechanisms to evade detection systems.
- *Encryption viruses* are a newcomer to the scene. They can scramble themselves to avoid detection.
- A *logic bomb* is designed to lie in wait until a predetermined event or action occurs. When this event occurs, the bomb or payload detonates and carries out its intended or designed action.

# WORM

Computer worms are among the most common types of malware. Unlike viruses, which require some sort of action to occur in order to trigger their mischief, worms are entirely self-replicating. Worms effectively use the power of networks, malware, and speed to spread very dangerous and effective pieces of malware. They spread over computer networks by exploiting operating system vulnerabilities

- Worms typically cause harm to their host networks by consuming bandwidth and overloading web servers. Computer worms can also contain "payloads" that damage host computers. Payloads are pieces of code written to perform actions on affected computers beyond simply spreading the worm.
- Payloads are commonly designed to steal data, delete files, or create botnets. Computer worms can be classified as a type of computer virus, but there are several characteristics that distinguish computer worms from regular viruses. A major difference is that computer worms have the ability to self-replicate and spread independently while viruses rely on human activity to spread (running a program, opening a file, etc). Worms often spread by sending mass emails with infected attachments to users' contacts.

# THE FUNCTIONING OF COMPUTER WORMS

Worms are an advanced form of malware, compared to viruses, and have different goals in many cases. One of the main characteristics of worms is their inherent ability to replicate and spread across networks extremely quickly, as the previous Slammer example demonstrated. Most worms share certain features that help define how they work and what they can do:

- Do not require a host application to perform their activities.
- Do not necessarily require any user interaction, direct or otherwise, to function.
- Replicate extremely rapidly across networks and hosts.
- Consume bandwidth and resources.
- Other functions worms perform:

Worms can also perform some other functions:

- Transmit information from a victim system back to another location specified by the designer.
- Carry a payload, such as a virus, and drop off this payload on multiple systems rapidly.

With these abilities in mind, it is important to distinguish worms from viruses by considering a couple of key points:

- A worm can be considered a special type of malware that can replicate and consume memory, but at the same time it does not typically attach itself to other applications or software.
- A worm spreads through infected networks automatically and requires only that a host is vulnerable. A virus does not have this ability.

# TROJAN HORSE

- One of the older and potentially widely misunderstood forms of malware is the Trojan.
- A *Trojan* is a software application that is designed to provide covert access to a victim's system. The malicious code is packaged in such a way that it appears harmless and thus gets around both the scrutiny of the user and the antivirus or other applications that are looking for malware. Once on a system, its goals are similar to those of a virus or worm: to get and maintain control of the system or perform some other task.

A Trojan infection may be indicated by some of the following behaviors:

- The CD drawer of a computer opens and closes.
- The computer screen changes, either flipping or inverting.
- Screen settings change by themselves.
- Documents print with no explanation.
- The browser is redirected to a strange or unknown web page.
- The Windows color settings change.
- Screen saver settings change.
- The right and left mouse buttons reverse their functions.
- The mouse pointer disappears.
- The mouse pointer moves in unexplained ways.
- The Start button disappears.
- Chat boxes appear.
- The Internet service provider (ISP) reports that the victim's computer is running port scans.
- People chatting with you appear to know detailed personal information about you.
- The system shuts down by itself.
- The taskbar disappears.
- Account passwords are changed.
- Legitimate accounts are accessed without authorization.
- Unknown purchase statements appear on credit card bills.

A Trojan relies on these items:

- An *overt channel* is a communication path or channel that is used to send information or perform other actions. HTTP and TCP/IP are examples of communication mechanisms that can and do send information legitimately.
- A *covert channel* is a path that is used to transmit or convey information but does so in a way that is illegitimate or supposed to be impossible but is able to circumvent security. The covert channel violates security policy on a system.

- Why would an attacker wish to use a Trojan instead of a virus?
- The reason is because a Trojan is more stealthy, coupled with the fact that it opens a covert channel that can be used to transmit information. The data transmitted can be a number of items, including identity information.

# SPYWARE

- Spyware is a type of malware that functions by spying on user activity without their knowledge. These spying capabilities can include activity monitoring, collecting keystrokes, data harvesting (account information, logins, financial data), and more.
- Spyware often has additional capabilities as well, ranging from modifying security settings of software or browsers to interfering with network connections.
- Spyware spreads by exploiting software vulnerabilities, bundling itself with legitimate software, or in Trojans.

# ROOTKIT

- A rootkit is a type of malicious software designed to remotely access or control a computer without being detected by users or security programs.
- Once a rootkit has been installed it is possible for the malicious party behind the rootkit to remotely execute files, access/steal information, modify system configurations, alter software (especially any security software that could detect the rootkit), install concealed malware, or control the computer as part of a botnet.
- Rootkit prevention, detection, and removal can be difficult due to their stealthy operation. Since a rootkit continually hides its presence, typical security products are not effective in detecting and removing rootkits.
- As a result, rootkit detection relies on manual methods such as monitoring computer behavior for irregular activity, signature scanning, and storage dump analysis.
- Organizations and users can protect themselves from rootkits by regularly patching vulnerabilities in software, applications, and operating systems, updating virus definitions, avoiding suspicious downloads, and performing static analysis scans.

Smitha Renny

# MALWARE PREVENTION AND REMOVAL

- There are several general best practices that organizations and individual users should follow to prevent malware infections. Some malware cases require special prevention and treatment methods, but following these recommendations will greatly increase a user's protection from a wide range of malware:
- Install and run anti-malware and firewall software. When selecting software, choose a program that offers tools for detecting, quarantining, and removing multiple types of malware. At the minimum, anti-malware software should protect against viruses, spyware, adware, Trojans, and worms. The combination of anti-malware software and a firewall will ensure that all incoming and existing data gets scanned for malware and that malware can be safely removed once detected.
- Keep software and operating systems up to date with current vulnerability patches. These patches are often released to patch bugs or other security flaws that could be exploited by attackers.
- Be vigilant when downloading files, programs, attachments, etc. Downloads that seem strange or are from an unfamiliar source often contain malware.

# CH 3
# TYPES OF ATTACKS AND THEIR COMMON PREVENTION MECHANISMS

- Networks are subject to attacks from malicious sources.
- Attacks can be from two categories: "Passive" when a network intruder intercepts data traveling through the network, and "Active" in which an intruder initiates commands to disrupt the network's normal operation or to conduct reconnaissance and lateral movement to find and gain access to assets available via the network

Smitha Renny

# KEYSTROKE LOGGING

- **Keystroke logging**, or **keylogging** or **keyboard capturing**, is the action of recording (logging) the keys struck on a keyboard, generally covertly, so that the person using the keyboard is unaware that their actions are being monitored.
- A powerful way of extracting information from a victim's system is to use a piece of technology known as a *keylogger*. Software in this category is designed to capture and report activity in the form of keyboard usage on a target system.
- When placed on a system, it gives the attacker the ability to monitor all activity on a system and reports back to the attacker.
- Under the right conditions, this software can capture passwords, confidential information, and other data.
- Data can then be retrieved by the person operating the logging program.
- A **keylogger** can be either software or hardware.

Ref: https://en.wikipedia.org/wiki/Keystroke_logging#Software-based_keyloggers

# SOFTWARE KEYLOGGER

- Software-based keyloggers are computer programs designed to work on the target computer's software.

There are several categories:

- **Hypervisor-based**: The keylogger can theoretically reside in a malware hypervisor running underneath the operating system
- **Kernel-based**: A program on the machine obtains root access to hide itself in the OS and intercepts keystrokes that pass through the kernel.

# HARDWARE-BASED KEYLOGGERS

Hardware-based keyloggers do not depend upon any software being installed as they exist at a hardware level in a computer system.

- **Firmware-based**: BIOS-level firmware that handles keyboard events can be modified to record these events as they are processed.
- **Keyboard hardware**: Hardware keyloggers are used for keystroke logging by means of a hardware circuit that is attached somewhere in between the computer keyboard and the computer, typically inline with the keyboard's cable connector.

- **Wireless keyboard and mouse sniffers**: These passive sniffers collect packets of data being transferred from a wireless keyboard and its receiver.

# COUNTERMEASURES

- **Anti-keyloggers:** An anti-keylogger is a piece of software specifically designed to detect keyloggers on a computer, typically comparing all files in the computer against a database of keyloggers looking for similarities which might signal the presence of a hidden keylogger.
- **Anti-spyware / Anti-virus programs**
- **Automatic form filler programs:** Automatic form-filling programs may prevent keylogging by removing the requirement for a user to type personal details and passwords using the keyboard.
- **On-screen keyboards:**
- **Keystroke interference software:** These programs attempt to trick keyloggers by introducing random keystrokes, although this simply results in the keylogger recording more information than it needs to

# DENIAL-OF-SERVICE ATTACK

- **Denial-of-service attack** (**DoS attack**) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet.
- Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.
- In a **distributed denial-of-service attack** (**DDoS attack**), the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source

# COUNTERMEASURES

- **Application front end hardware: This** intelligent hardware placed on the network before traffic reaches the servers. It can be used on networks along with routers and switches. Application front end hardware analyzes data packets as they enter the system, and then identifies them as priority, regular, or dangerous
- **IPS based prevention:**
- **DDS based defense**
- **Firewalls**
- **Routers**

# PHISHING

- Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords and credit card details by disguising as a trustworthy entity in an electronic communication.
- It is carried out by email spoofing or instant messaging, it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site.
- Phishing is an example of social engineering techniques being used to deceive users.

# COUNTERMEASURES

1. **Anti-phishing**

- There are anti-phishing websites which publish exact messages that have been recently circulating the internet, such as FraudWatch International and Millersmiles. Such sites provide specific details about the particular messages.

2. **User training**

- People can be trained to recognize phishing attempts, and to deal with them through a variety of approaches. Such education can be effective, especially where training emphasises conceptual knowledge and provides direct feedback.

3. **Browsers alerting users to fraudulent websites**

- Another popular approach to fighting phishing is to maintain a list of known phishing sites and to check websites against the list. One such service is the Safe Browsing service. Web browsers such as Google Chrome, Internet Explorer 7, Mozilla Firefox 2.0, Safari 3.2, and Opera all contain this type of anti-phishing measure.

4. **Filtering out phishing mail**

- Specialized spam filters can reduce the number of phishing emails that reach their addressees' inboxes, or provide post-delivery remediation, analyzing and removing spear phishing attacks upon delivery through email provider-level integration.

5. **Monitoring and takedown**

- Several companies offer banks and other organizations likely to suffer from phishing scams round-the-clock services to monitor, analyze and assist in shutting down phishing websites. Individuals can contribute by reporting phishing to both volunteer and industry groups, such as cyscon or PhishTank.

# SESSION HIJACKING

- **Session hijacking**, also known as **cookie hijacking** is the exploitation of a valid computer session—sometimes also called a *session key*—to gain unauthorized access to information or services in a computer system.
- In particular, it is used to refer to the theft of a magic cookie used to authenticate a user to a remote server.
- It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many web sites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim's comp

# COUNTERMEASURES

1. Encryption of the data traffic passed between the parties by using SSL/TLS; in particular the session key. This technique is widely relied-upon by web-based banks and other e-commerce services, because it completely prevents sniffing-style attacks.
2. Use of a long random number or string as the session key. This reduces the risk that an attacker could simply guess a valid session key through trial and error or brute force attacks.
3. Regenerate the session id after a successful login. This prevents session fixation because the attacker does not know the session id of the user after s/he has logged in.

4. Some services make secondary checks against the identity of the user. For instance, a web server could check with each request made that the IP address of the user matched the one last used during that session. This does not prevent attacks by somebody who shares the same IP address
5. Some services change the value of the cookie with each and every request. This reduces the window in which an attacker can operate and makes it easy to identify whether an attack has taken place.
6. Users may also wish to log out of websites whenever they are finished using them.

# CLICKJACKING

- **Clickjacking** is a malicious technique of tricking a user into clicking on something different from what the user perceives , thus potentially revealing confidential information or allowing others to take control of their computer while clicking on seemingly innocuous objects, like web pages.
- In Web browsers, clickjacking is a browser security issue that is a vulnerability across a variety of browsers and platforms. Clickjacking can also take place outside of web browsers, including applications.

# COUNTERMEASURES

- ***NoScript***
- Protection against clickjacking can be added to Mozilla Firefox desktop and mobile versions by installing the NoScript add-on: its ClearClick feature, released on 8 October 2008, prevents users from clicking on invisible or "redressed" page elements of embedded documents or applets.
- ***GuardedID***
- GuardedID (a commercial product) includes client-side clickjack protection for users of Internet Explorer and Firefox without interfering with the operation of legitimate iFrames. GuardedID clickjack protection forces all frames to become visible.
- ***Gazelle***
- Gazelle is a Microsoft Research project secure web browser based on IE, that uses an OS-like security model, and has its own limited defenses against clickjacking. In Gazelle, a window of different origin may only draw dynamic content over another window's screen space if the content it draws is opaque.

# COOKIE THEFT

- An HTTP cookie, is a small piece of data sent from a website and stored in the user's web browser while the user is browsing it.
- Every time the user loads the website, the browser sends the cookie back to the server to notify the user's previous activity.
- Cookies are basically just text files, stored on your computer, used by the browser to save useful information about actions you take.
- At times when information worth power, even large, established and well-secured companies find themselves under continuous attempts of cookie theft attacks. Hackers will do everything they can in order to access private and sensitive information and gain control over private accounts.

# COUNTERMEASURES

1. Require the use of SSL/TLS connection on all your website pages. Doing so increases the CPU load time so at least consider using SSL/TLS for login pages and pages which transmit cookies.
2. Set a session timeout to make sure the session is deleted after a fixed amount of time – provide a "remember me" button to make it easy for users to stay logged in after the session has timed out.
3. Mark the cookie as an HttpOnly cookie (using the attribute) to ensure that this cookie cannot be transmitted via scripts.
4. Restrict the cookie domain to a minimum.
5. Do not use consecutive numbers for session ID values, as they are easy to guess.
6. Invalidate the session before authenticating a new user.

# URL OBFUSCATION

- Also called a hyperlink trick, an obfuscated URL is a type of attack where the real URL that a user is directed to is obfuscated - or concealed - to encourage the user to click-through to the spoof Web site.
- For example, the attacker may use a cleverly misspelled domain name (e.g. PayPals.com instead of PayPal.com), or hide the actual URL in friendly text, such as "click here to verify your account now". Obfuscated URLs are commonly used in phishing attacks and other spam e-mails.

# COUNTERMEASURES

1. Desktop protection technologies
2. Local Anti-Virus protection of Personal Firewall
3. Personal IDS or Personal Anti-Spam
4. Spyware Detection  Utilization of appropriate less sophisticated
5. communication settings.  User application-level monitoring solutions.
6. Locking-down browser capabilities;
7. Digital signing and validation of email;
8. To help prevent many Phishing attack vectors, web browser users should:
9. Disable all window pop-up functionality
10. Disable Java runtime support or Disable ActiveX support or Disable all multimedia and auto-play/auto-execute extensions
11. Prevent the storage of non-secure cookies to Ensure that any downloads cannot be automatically run from the browser, and must instead be downloaded into a directory for anti-virus inspection.

# FAKE WAP

- Fake WAP (Wireless Access Point) is a type of hacking attack in which the hacker sets up a wireless router with a convincingly legitimate name in a public spot where people might connect to it. Eg a spoofed WiFi hotspot
- Once they do, the hacker can monitor and even change internet connections to steal sensitive data or force the user to download malware onto their device.
- Eg  a hotel, cafe, or airport that has one or more separate guest wifi networks. It may not be connected to a secured router owned by the establishment you are visiting.

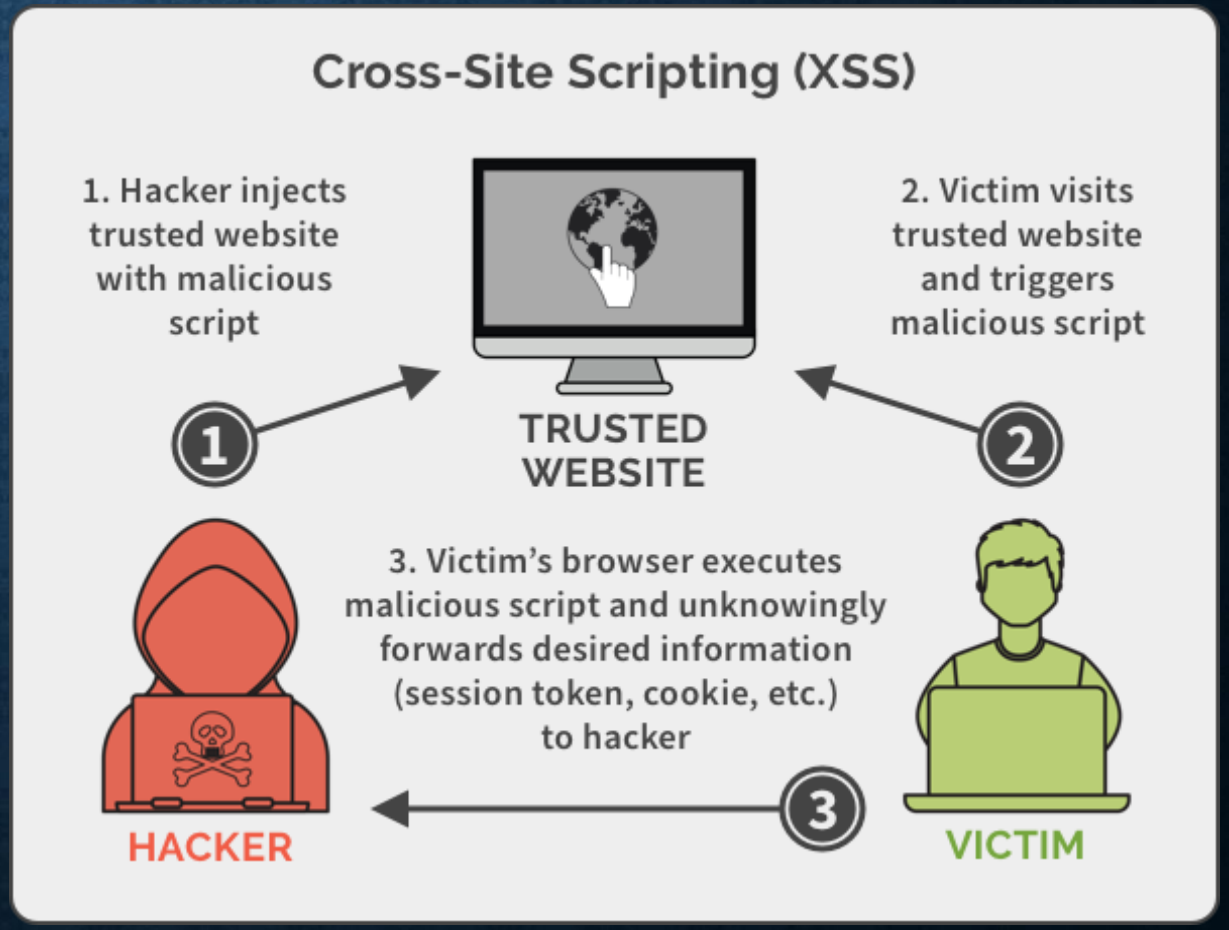
# COUNTERMEASURES

- Avoid the use of free Wi-Fi hotspots
- Avoid automatic connections
- Ignore unexpected communications
- Avoid using apps from untrusted sources

# CH 2 - TYPES OF VULNERABILITIES

## CROSS-SITE SCRIPTING (XSS)

- Cross-site scripting (XSS) is a type of injection security attack in which an attacker injects data, such as a malicious script, into content from otherwise trusted websites.
- Typically, this data is included as part of dynamic content that has not gone through validation checks.



- Cross-site scripting attacks happen when an untrusted source is allowed to inject its own code into a web application, and that malicious code is included with dynamic content delivered to a victim's browser.
- Cross-site scripting allows an attacker to execute malicious scripts in another user's browser.
- However, the attacker doesn't attack the victim directly; rather, the attacker exploits a vulnerability in a website the victim visits and gets the website to deliver the malicious script for the attacker.
- XSS can be used in a number of ways to cause serious problems.
- There are two main forms of this attack:
  - ❖Stored XSS attacks
  - ❖reflected XSS attacks.

# STORED XSS ATTACKS

- XSS attacks that fall into this category tend to be the most dangerous type.
- The attack is enabled by any web application allows a visitor to store data when they visit the site.
- Web applications gather input from a visitor and store the input within a data store for later retrieval and use.
- When a malicious visitor visits the site and their malicious input is stored in the data store their data is also part of the site.
- When a subsequent visitor comes to the site, they inadvertently run the same data. Since the code runs locally, it will run with the security privileges of the client application.

- Depending on how the data is crafted, the attack can carry out a number of tasks, including these:
  - Hijacking another user's browser
  - Capturing sensitive information viewed by application users
  - Pseudo defacement of the application
  - Port scanning of internal hosts (internal in relation to the users of the web application)
  - Directed delivery of browser-based exploits

# REFLECTED XSS ATTACK

- Also known as non-persistent attacks
- Occurs when a malicious script is reflected off a web application to the victim's browser
- The script is activated through a link which sends a request to a website with a vulnerability that enables execution of malicious scripts

- Stored XSS is more damaging of the two. It occurs when a malicious script is injected directly into a vulnerable web application.

# CROSS SITE REQUEST FORGERY (CSRF/XSRF)

- Cross-site request forgery (XSRF or CSRF) is a method of attacking a Web site in which an intruder masquerades as a legitimate and trusted user.
- An XSRF attack can be used to modify firewall settings, post unauthorized data on a forum or conduct fraudulent financial transactions.
- An XSRF attack can be executed by stealing the identity of an existing user and then hacking into a Web server using that identity.

# DIFFERENCE BETWEEN XSS AND CSRF

- XSS and CSRF are two types of computer security vulnerabilities.
- XSS stands for Cross-Site Scripting. CSRF stands for Cross-Site Request Forgery.
- In XSS, the hacker takes advantage of the trust that a user has for a certain website. On the other hand, in CSRF the hacker takes advantage of a website's trust for a certain user's browser.

| | 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. |

# SQL INJECTION

- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the placement of malicious code in SQL statements, via web page input.
- SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
- It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access.
- In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behaviour.

*https://www.hacksplaining.com/exercises/sql-injection#/start*

# INPUT PARAMETER MANIPULATION

- The Web Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc.
- Usually, this information is stored in cookies, hidden form fields, or URL Query Strings, and is used to increase application functionality and control.
- This attack can be performed by a malicious user who wants to exploit the application for their own benefit, or an attacker who wishes to attack a third-person using a Man-in-the-middle attack.

# BROKEN AUTHENTICATION

- These types of weaknesses can allow an attacker to either capture or bypass the authentication methods that are used by a web application.
  1. User authentication credentials aren't protected when stored using hashing or encryption.
  2. Credentials can be guessed or overwritten through weak account management functions (e.g., account creation, change password, recover password, weak session IDs).
  3. Session IDs are exposed in the URL (e.g., URL rewriting).
  4. Session IDs are vulnerable to session fixation attacks.
  5. Session IDs don't timeout, or user sessions or authentication tokens, particularly single sign-on (SSO) tokens, aren't properly invalidated during logout.
  6. Session IDs aren't rotated after successful login.
  7. Passwords, session IDs, and other credentials are sent over unencrypted connections.

# SENSITIVE INFORMATION DISCLOSURE

- Sensitive Data Exposure occurs when an application does not adequately protect sensitive information. The data can vary and anything from passwords, session tokens, credit card data to private health data and more can be exposed.
- The first thing you have to determine is which data is sensitive enough to require extra protection. For example, passwords, credit card numbers, health records, and personal information should be protected. For all such data:
    1. Is any of this data stored in clear text long term, including backups of this data?
    2. Is any of this data transmitted in clear text, internally or externally? Internet traffic is especially dangerous.
    3. Are any old / weak cryptographic algorithms used?
    4. Are weak crypto keys generated, or is proper key management or rotation missing?
    5. Are any browser security directives or headers missing when sensitive data is provided by / sent to the browser?

# XML EXTERNAL ENTITIES

- XML input containing a reference to an external entity is processed by a weakly configured XML parser.
- This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.
- An XML External Entity attack is a type of attack against an application that parses XML input.
- This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.

# BROKEN ACCESS CONTROL

- Broken Authentication involves all kinds of flaws that are caused by error in implementations of authentication and/or session management.
- The category includes everything from login lacking timeout, meaning that users who forget to logout on a public computer can get hijacked, to more technical vulnerabilities such as session fixation

# SECURITY MISCONFIGURATION

- Security misconfiguration vulnerabilities could occur if a component is susceptible to attack due to an insecure configuration option.
- These vulnerabilities often occur due to insecure default configuration, poorly documented default configuration, or poorly documented side-effects of optional configuration.
- This could range from failing to set a useful security header on a web server, to forgetting to disable default platform functionality that could grant administrative access to an attacker.

# USING COMPONENTS WITH KNOWN VULNERABILITIES

- This kind of threat occurs when the components such as libraries and frameworks used within the app almost always execute with full privileges.

- If a vulnerable component is exploited, it makes the hacker's job easier to cause a serious data loss or server takeover.

# INSUFFICIENT LOGGING AND MONITORING

- Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident.
  Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.
- Insufficient logging, detection, monitoring and active response occurs any time:
- Auditable events, such as logins, failed logins, and high-value transactions are not logged.
- Warnings and errors generate no, inadequate, or unclear log messages.
- Logs of applications and APIs are not monitored for suspicious activity.
- Logs are only stored locally.
- Appropriate alerting thresholds and response escalation processes are not in place or effective.
- Penetration testing and scans by DAST tools (such as OWASP ZAP) do not trigger alerts.
- The application is unable to detect, escalate, or alert for active attacks in real time or near real time.

# OWASP MOBILE TOP 10 (OPEN WEB APPLICATION SECURITY PROJECT®)

- This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, or some other security control that is part of the mobile operating system.
- The defining characteristic of risks in this category is that the platform (iOS, Android, Windows Phone, etc.) provides a feature or a capability that is documented and well understood.
- The app fails to use that capability or uses it incorrectly. This differs from other mobile top ten risks because the design and implementation is not strictly the app developer's issue.

- There are several ways that mobile apps can experience this risk.
    1. Violation of published guidelines. All platforms have development guidelines for security (c.f., ((Android)), ((iOS)), ((Windows Phone))). If an app contradicts the best practices recommended by the manufacturer, it will be exposed to this risk. For example, there are guidelines on how to use the iOS Keychain or how to secure exported services on Android. Apps that do not follow these guidelines will experience this risk.
    2. Violation of convention or common practice. Not all best practices are codified in manufacturer guidance. In some instances, there are de facto best practices that are common in mobile apps.
    3. Unintentional Misuse. Some apps intend to do the right thing, but actually get some part of the implementation wrong. This could be a simple bug, like setting the wrong flag on an API call, or it could be a misunderstanding of how the protections work.

# CVE DATABASE (COMMON VULNERABILITIES & EXPOSURES)

- Common Vulnerabilities and Exposures (CVE®) is a list of common identifiers for publicly known cybersecurity vulnerabilities.
- It is a program launched in 1999, that operates research and development centers sponsored by the federal government, to identify and catalog vulnerabilities in software or firmware into a free "dictionary" for organizations to improve their security.
- The dictionary's main purpose is to **standardize the way each known vulnerability or exposure is identified.** Standard IDs allow security administrators to access technical information about a specific threat across multiple CVE-compatible information sources.

- CVE isn't just another vulnerability database.

- It is designed to allow vulnerability databases and other capabilities to be linked together, and to facilitate the comparison of security tools and services.

- A CVE listing only contains the standard identifier number with status indicator, a brief description and references to related vulnerability reports and advisories. It does not include risk, impact, fix or detailed technical information.

- The CVE helps computer security tool vendors identify vulnerabilities and exposures. The **key objective** of CVE is to help share data across different vulnerable databases and security tools
    - ❖ Lists all publicly known security problems
    - ❖ Assigns unique identifier to each problem
    - ❖ Remains independent of multiple perspectives
    - ❖ Is publicly open and shareable
    - ❖ Community-wide effort via the CVE editorial Board.

# FIELDS OF CVE ENTRY

## 1. CVE ID

- CVE Identifiers (also called "CVE names," "CVE numbers," "CVE-IDs," and "CVEs") are unique, common identifiers for publicly known information security vulnerabilities.
- Each CVE Identifier includes the following:
  - CVE identifier number (i.e., "CVE-1999-0067").
  - Indication of "entry" or "candidate" status.
  - Brief description of the security vulnerability or exposure.
  - Any pertinent references (i.e., vulnerability reports and advisories or OVAL-ID).
  - CVE Identifiers are used by information security product/service vendors and researchers as a standard method for identifying vulnerabilities and for cross-linking with other repositories that also use CVE Identifiers It is a number with four or more digits in the sequence number portion of the ID (e.g., "CVE-1999-0067", "CVE-2014-12345", "CVE-2016-7654321").
- The new CVE-ID syntax is variable length and includes: CVE prefix + Year + Arbitrary Digits

## 2. Description

- This is a standardized text description of the issue(s).

## 3. References : -

- Each reference used in CVE has the following structure:

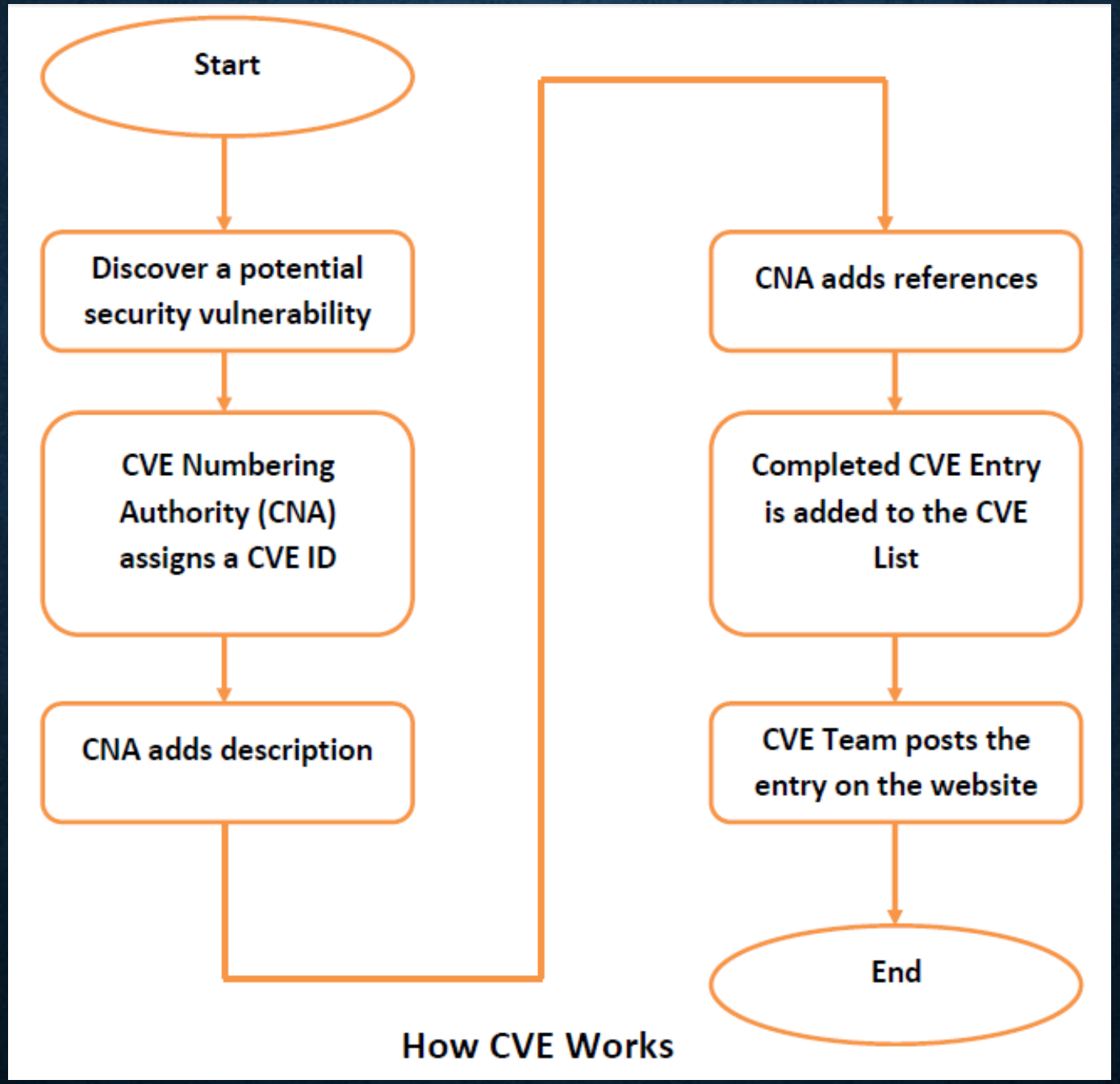
  **SOURCE: NAME**

- Where SOURCE is an alphanumeric keyword. (Examples: "BUGTRAQ", "OVAL", etc.)
- NAME is a single line of ASCII text and can include colons and spaces. (Examples: "BUGTRAQ: Posting to Bugtraq mailing list"; "OVAL: Open Vulnerability and Assessment Language (OVAL) vulnerability definition"; etc.)
- Where possible, the NAME is selected to facilitate searches on a SOURCE's website.
- For references that do not have a well-defined identifier, a release date and/or subject header may be included.

4. **Assigning CNA:** CVE Numbering Authority is the organization authorized to assign CVE ID to vulnerabilities for inclusion in first-time public announcements of new vulnerabilities.
5. **Date entry created :-**
    - This is the date the entry was created.
    - For CVEs assigned directly by Mitre, this is the date Mitre created the CVE entry.
    - For CVEs assigned by CNAs (e.g. Microsoft, Oracle, HP, Red Hat, etc.) this is also the date the entry was create by Mitre, not by the CNA.
    - So in the case where a CNA requests a block of CVE numbers in advance (e.g. Red Hat currently requests CVEs in blocks of 500) the entry date would be when that CVE is assigned to the CNA.
    - The CVE itself may not be used for days, weeks, months or even possibly years (e.g. Red Hat maintains blocks of CVEs for older security issues in open-source software that were not assigned CVEs yet).

# HOW CVE WORKS

- The process of creating a CVE Entry begins with the discovery of a potential security vulnerability.
- The information is then assigned a CVE ID by a CVE Numbering Authority (CNA), the CNA writes the Description and adds References, and then the completed CVE Entry is added to the CVE List and posted on the CVE website by the CVE Team.




How CVE Works

# END OF UNIT I