What are Application Layer Attacks? What to do when attacked at the application layer?

Application Layer Attacks are a type of Distributed Denial of Service (DDoS) attack that is a type of cyber attack that targets the top layer in the OSI model, where protocols like HTTP GET and HTTP POST operate.

Application Layer Attacks are attacks that target applications, especially in the web environment. These attacks often aim to disrupt application operations, steal data, or perform other malicious actions without having to penetrate the overall network system. Let's learn more with TipsMake in the following article.

What are Application Layer Attacks?

Application Layer Attacks are a type of Distributed Denial of Service (DDoS) attack that is a type of cyber attack that targets the top layer of the OSI model, where protocols like HTTP GET and HTTP POST operate. These attacks are often carried out through methods like Layer 7 (L7) DDoS, which aims to exhaust a server's resources by sending a large number of valid requests to the target application.

How do Application Layer Attacks work?

Application Layer Attacks take advantage of the disparity between the amount of resources required to execute an attack and the amount of resources required to defend against it. Application Layer Attacks cause more damage but use less bandwidth.

When a user sends a request such as logging into an online account, the amount of resources consumed by the client is very small compared to the process of the server validating the information, retrieving data, and responding. Even without the login information, the server still needs to perform a database query or API call to process the request.

When multiple devices attack the same web property via a botnet, this can overwhelm the server, resulting in the service being disabled for legitimate users. In particular, simply targeting the API with an L7 attack is enough to take the service offline.

How many types of Application Layer Attacks are there?

SQL Injection

SQL Injection is one of the most common attacks in the field of application layer security. Attackers exploit vulnerabilities in SQL queries to perform unauthorized actions such as retrieving, modifying or deleting data in the database.

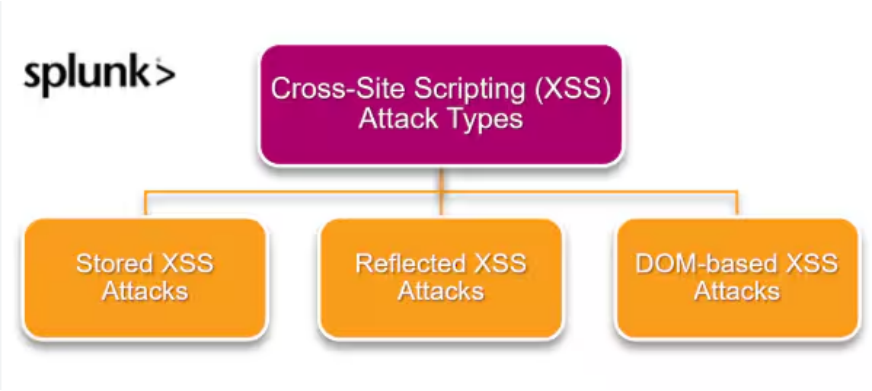
The principle of SQL Injection is simple: an attacker will insert malicious SQL code into the application's input fields. When the application executes the SQL statement without validating the input data, the attacker can completely control the query and perform unwanted actions.

The consequences of SQL Injection are extremely serious, including:

1. Steal sensitive data.
2. Create fake account.
3. Corrupt or delete data in the database.

XSS Attacks

Cross-Site Scripting (XSS) is a form of attack that allows attackers to insert malicious code into websites that users visit. This can lead to stealing user information or performing other malicious actions.



There are three main types of XSS, which are:

1. **Stored XSS:** Malicious code is stored on the server and sent to users every time they visit the website.
2. **Reflected XSS:** Malicious code is sent via URL and reflected back immediately without being stored on the server.
3. **DOM-based XSS:** Malicious code is executed on the client side without interacting with the server.

To prevent XSS, developers should:

1. Check and filter input data.

2. Use security libraries to automate protection processes.
3. Limit access to sensitive parts of the app.

CSRF Attacks

CSRF (Cross-Site Request Forgery) is a type of attack in which an attacker takes advantage of the trust of an authenticated user on a website to perform unwanted behavior.

The attacker sends a request to the application from another website that the victim does not know about. If the victim is logged into the application, the request will be made with their permissions, leading to malicious actions such as changing account information or transferring money.

To mitigate the risk of CSRF, application developers should:

1. Use authentication token for each request.
2. Check the 'Referer' parameter to confirm the origin of the request.

DDoS attacks on specific applications

A Distributed Denial of Service (DDoS) attack aims to paralyze an application by sending a large number of requests to the server. This can make the application unusable for legitimate users.

Attackers often use a network of bots to launch attacks simultaneously from many different IP addresses. As a result, the server cannot handle all the requests and becomes unavailable.

To prevent DDoS attacks, businesses can:

1. Use DDoS protection services.
2. Set a limit on the number of requests from an IP address within a given period of time.

Buffer Overflow

Buffer Overflow is a security vulnerability that occurs when an attacker overwrites an application's memory, allowing them to execute malicious code. When an application fails to adequately check the size of input data, an attacker can pass data that is larger than the allowed memory size. This can result in overwriting memory that the application does not have access to, allowing malicious code to be executed.

There are 2 ways to prevent Buffer Overflow, which are:

1. Use safer programming languages.
2. Check the input data size carefully.

Brute Force Attacks

Brute Force Attacks are a type of attack where an attacker attempts to guess a password by trying all possible options. The attacker uses automated tools to repeatedly try passwords until the correct one is found. This often takes a long time but can be successful if the password is weak.

To prevent Brute Force Attacks, businesses should apply the following 2 methods:

1. Use strong, complex passwords.
2. Limit the number of password attempts before locking the account.

Using Malware and Viruses

Attackers can use malware or viruses to take control of a system or steal sensitive information. Malware can be delivered via email, downloaded from a malicious website, or through untrusted applications. Once on a system, malware can do a variety of things, such as steal information or track user activity.

To prevent malware, install reliable anti-virus software and update your operating system and software regularly.

Directory Traversal

Directory Traversal is a form of attack that aims to gain unauthorized access to files on a server. Attackers exploit vulnerabilities in applications to gain access to sensitive files in the server's file system, allowing them to steal information or perform other malicious actions.

To prevent Directory Traversal, developers should check and validate file paths. Also, use security libraries to isolate application directories.

API Attacks

APIs (Application Programming Interfaces) are one of the most important components of modern application development. However, APIs can also become targets for attacks.

Attackers can exploit vulnerabilities in the API to perform attacks such as:

1. Steal data through sending malicious requests.
2. Activating invalid functions, resulting in data corruption.
3. To protect the API from attacks:
4. Use authentication and authorization for requests.
5. Monitor and analyze API activity to detect anomalous behavior.

What are the dangerous consequences of Application Layer Attacks?

Application Layer Attacks have serious consequences for the organization or individual attacked. These consequences not only affect data but can also impact the organization's reputation and finances.

1. **Data Loss:** One of the most serious consequences of Application Layer Attacks is data loss. Attackers can steal or delete important data, leading to serious financial losses for the organization as well as the risk of litigation from customers.
2. **Reputation Impact:** When an organization is attacked, their reputation can be severely affected. Consumers tend to lose trust in organizations that do not ensure information security, and this can lead to a loss of customers.
3. **Financial Damage:** The financial damage from these attacks can be significant, including the cost of remediation, customer compensation, and even fines from regulatory agencies. Organizations may also have to pay for additional security services to prevent future attacks.

4. **Legal Risk:** If sensitive customer data is stolen, the organization may face significant litigation and legal risks. This not only affects finances but can also lead to loss of business.

What to do when attacked by Application Layer Attacks?

Layer 7 DDoS attacks typically target the application layer, using malicious requests to overload servers and result in denial of service. Here are some measures organizations can take to minimize the risk of Application Layer Attacks:

1. **Increase web server connection limits:** This measure reduces vulnerability to connection-based attacks like Slowloris, cutting down on the number of open connections needed for an attacker to maintain control of a resource.
2. **Implement rate limiting:** Limiting the number of requests from any IP address helps prevent DDoS attacks. Organizations need to regularly monitor traffic to detect anomalies and take timely action.
3. **Use load balancers and web application firewalls (WAFs):** WAFs protect against DDoS attacks by identifying and blocking malicious traffic before it reaches the network. They also provide logs to help organizations detect potential threats. Load balancers and reverse proxies can buffer connections and apply management techniques to protect applications and web servers from incomplete requests.
4. **Use cloud-based DDoS protection services:** These solutions provide rapid detection of suspicious activity and timely response.
5. **Apply security best practices:** Update software and patch regularly to minimize the risk of attacks, including techniques like the Slowloris DDoS.

Conclude

Application Layer Attacks are a serious threat to any organization operating in an online environment. Understanding the types of attacks, how they work, and the possible consequences is an important step in protecting your data and reputation. At the same time, preparing a timely response plan will help your organization minimize damage and regain customer trust after unexpected incidents.