OWASP TOP API Attack 2023 - Server Side Request Forgery by APISecurityEngine

Introduction

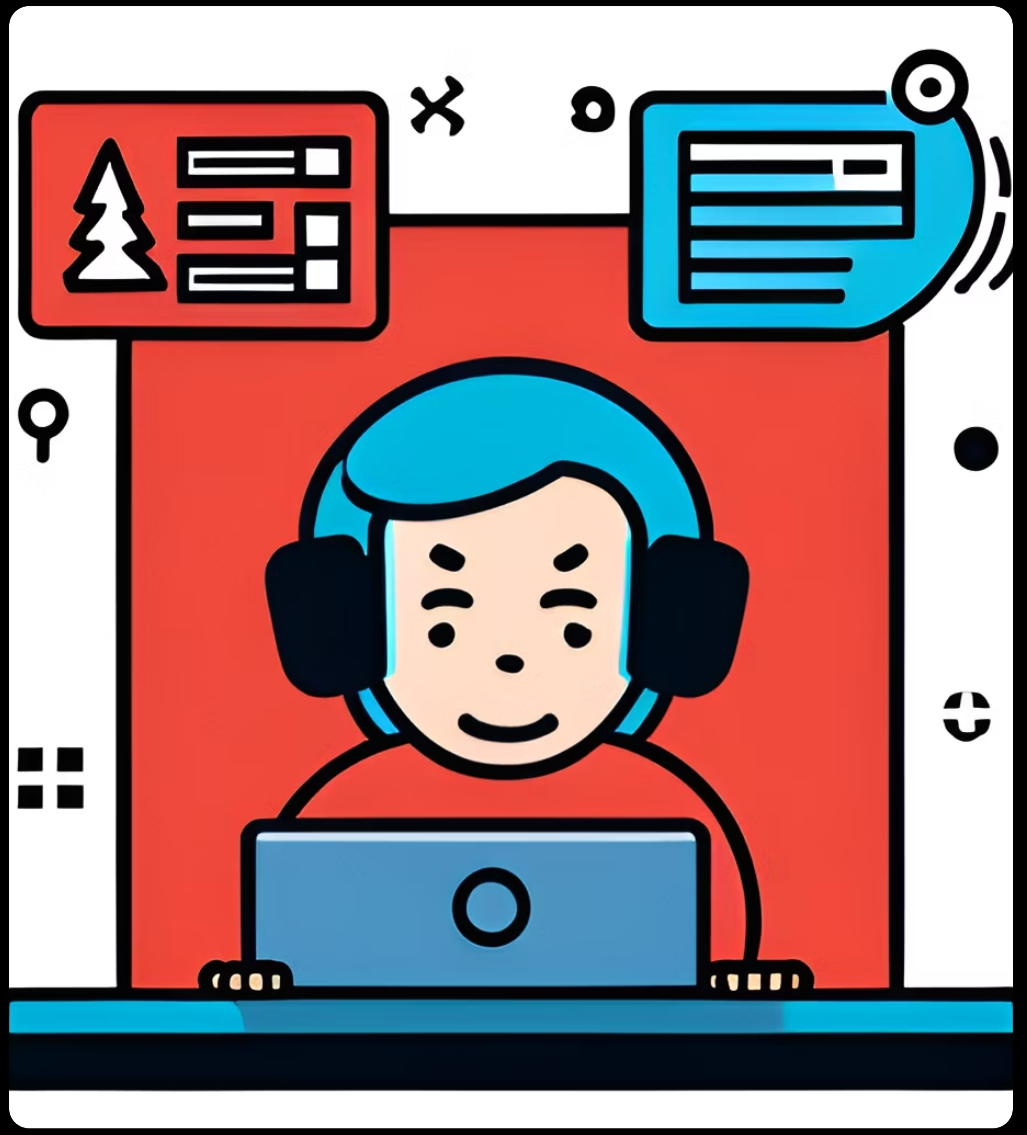
Welcome to our presentation on the OWASP TOP API Attack 2023 - Server Side Request Forgery (SSRF). Today, we'll be discussing a serious topic that affects many organizations and their APIs.

Did you know that according to a recent report, SSRF attacks accounted for over 30% of all API vulnerabilities? This is a significant number and highlights the importance of understanding and preventing these types of attacks. So let's dive in and learn more about what SSRF is and why it's so important.

What is SSRF?

Server Side Request Forgery (SSRF) is a type of attack that targets web applications by exploiting their trust in other systems. It works by tricking the application into sending requests to a different server than intended, often resulting in sensitive information being exposed or even allowing the attacker to take control of the system.

One common example of an SSRF attack is when an application allows users to upload files to the server, but fails to properly validate the input. An attacker can then upload a file containing malicious code that will execute on the server, giving them access to sensitive data or allowing them to launch further attacks.



Impact of SSRF Attacks

SSRF attacks can have devastating consequences for organizations, as they can be used to steal sensitive data, compromise systems, and launch further attacks. In fact, according to a recent report by Akamai, SSRF attacks accounted for over 30% of all web application attacks in 2020.

One of the most high-profile examples of an SSRF attack occurred in 2019, when Capital One suffered a major data breach that exposed the personal information of over 100 million customers. The attacker was able to exploit an SSRF vulnerability in the company's AWS infrastructure to access sensitive data stored in the cloud.

Another example is the 2017 Equifax breach, which was caused by an unpatched vulnerability in the company's web application framework that allowed attackers to perform SSRF attacks. This resulted in the theft of personal information for over 143 million people, including names, birthdates, social security numbers, and more.

Clearly, the impact of SSRF attacks can be severe and far-reaching. Organizations must take steps to prevent and detect these attacks to protect themselves and their customers.

Preventing SSRF Attacks

One practical step for preventing SSRF attacks is input validation. This involves checking user input to ensure that it is in the expected format and within acceptable ranges. For example, if an application expects a URL as input, it should validate that the input starts with 'http://' or 'https://', and does not contain any characters that could indicate an SSRF attack, such as 'localhost' or '127.0.0.1'.

Another step is whitelisting. This involves maintaining a list of trusted sources and only allowing requests from those sources. For example, an application might only allow requests from known IP addresses or domains. Whitelisting can help prevent SSRF attacks by limiting the potential targets for attackers.

Using secure coding practices is also important for preventing SSRF attacks. This includes avoiding the use of user input directly in API calls, and using safe alternatives such as parameterized queries or prepared statements. Additionally, developers should be trained on how to recognize and avoid common security vulnerabilities, including SSRF.

By implementing these practical steps, organizations can greatly reduce their risk of falling victim to an SSRF attack.

Detecting SSRF Attacks

Detecting SSRF attacks can be challenging, but there are several techniques that can help. One approach is to monitor network traffic for unusual patterns, such as requests to internal IP addresses or unexpected ports. This can be done using tools like Wireshark or tcpdump, which capture and analyze network packets.

Another technique is to use a web application firewall (WAF) that can detect and block SSRF attacks. WAFs can be configured to inspect incoming requests and block those that match known attack patterns. Additionally, some cloud providers offer built-in protections against SSRF attacks, such as AWS's VPC endpoint policies.

Conclusion

In conclusion, SSRF attacks are a serious threat to organizations that use APIs. These attacks can lead to data breaches, financial losses, and damage to reputation. It is important to take steps to prevent and detect these attacks before they cause harm.

To prevent SSRF attacks, organizations should implement input validation, whitelisting, and secure coding practices. They should also monitor network traffic for unusual patterns and use tools such as firewalls and intrusion detection systems.

As we have seen, the impact of SSRF attacks can be devastating. In 2019, Capital One suffered a massive data breach due to an SSRF attack. This breach affected over 100 million customers and cost the company over $100 million in damages. We cannot afford to ignore this threat any longer.