Report 01

URL: us.coca-cola.com

BREACH Attack Detected

HIGH

The BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext) attack vulnerability refers to a specific type of security flaw that targets systems or applications using certain compression techniques, such as GZIP, in combination with predictable data patterns. This vulnerability allows an attacker to deduce sensitive information by exploiting the behavior of the compression algorithms. The attack takes advantage of the fact that compression algorithms work by finding repeated patterns in data and encoding them more efficiently. When predictable patterns, such as user tokens or session identifiers, are present in the data being compressed, they create a distinguishable pattern in the compressed output. The attacker injects specially crafted malicious data into a user-input field, typically within a web form. By modifying the injected data and repeatedly sending requests to the vulnerable system, the attacker can observe the size of the compressed responses. Through careful analysis of the variations in response sizes, the attacker can infer whether specific patterns or information are present in the compressed responses. This allows them to deduce sensitive information that should remain confidential, such as user credentials or other sensitive data.

Affected URL: https://us.coca-cola.com/store/retail/atlanta

Affected Components

- Web Applications or Systems: The vulnerability can impact the overall web application or system that employs compression algorithms, such as GZIP, for reducing network

traffic. If the system processes user-supplied data that contains predictable patterns, it becomes susceptible to a BREACH attack.

- ❖ Web Servers: The web server component may play a role in the vulnerability if it is responsible for handling compression and decompression of data. If compression is applied to user-provided input without proper mitigation measures, the server can inadvertently expose the vulnerability.

- ❖ Network Communication: The vulnerability primarily arises during the transmission of data over the network. If the system or application sends compressed responses that contain predictable patterns, an attacker can exploit the differences in response sizes to deduce sensitive information.

- ❖ User-Input Fields: User-input fields within web forms or other interactive elements of the application can serve as entry points for the BREACH attack. Crafted malicious data is injected into these fields, triggering multiple requests and allowing the attacker to analyze the variations in response sizes.

Impact Assessment

The impact of the BREACH attack vulnerability can be significant and pose various risks to the affected system or application.

- Disclosure of Sensitive Information: The primary consequence of a successful BREACH attack is the disclosure of sensitive information. Attackers can deduce confidential data, such as user credentials, session tokens, or other sensitive information embedded in the compressed responses. This can lead to unauthorized access, identity theft, or compromise of sensitive data.

- Account Takeover: With access to user credentials or session tokens obtained through the BREACH attack, attackers can potentially hijack user accounts. They can

  impersonate legitimate users, gain unauthorized access to their accounts, and perform malicious activities on their behalf, including unauthorized transactions, data manipulation, or spreading malware.

- Privacy Breach: The disclosure of sensitive information through a BREACH attack can result in a significant privacy breach. Personally identifiable information (PII), financial data, or other confidential information that is exposed can be exploited for various malicious purposes, including identity theft, blackmail, or unauthorized profiling.

- Compliance Violations: Depending on the nature of the system and the type of data involved, a successful BREACH attack can lead to compliance violations. Industries with strict data protection regulations, such as healthcare (HIPAA), finance (PCI-DSS), or personal data protection laws (such as GDPR), may face legal and financial consequences if confidential data is compromised.

- Reputational Damage: Data breaches and privacy incidents can severely damage the reputation of an organization. If users or customers lose trust in the security of the affected system, it can result in reputational harm, loss of business opportunities, and diminished customer loyalty.

- Financial Losses: BREACH attacks can have financial implications for both individuals and organizations. Financial losses may occur due to fraudulent transactions, legal fines and penalties resulting from non-compliance, costs associated with mitigating the attack, and potential legal actions taken by affected parties.

Steps to Reproduce

- Identify a target: First, you need to identify a site that uses HTTP compression, uses HTTP responses with secrets in the body, and has a stable reflected parameter.

- Establish a Man-in-the-Middle (MITM) position: For a BREACH attack to be successful, the attacker must be able to read the user's network traffic. This is often done by manipulating the network so that the attacker is in a MITM position between the user and the web server. This could be achieved by infecting the user's machine with malware, or by setting up a rogue access point, etc.

- Inject code into the client's browser: This can be accomplished through various methods such as XSS attacks, malicious advertisements, or social engineering attacks. The code should be capable of making multiple requests to the target site on behalf of the user and analyzing the size of the responses.

- Execute the attack: The malicious code makes a large number of requests to the target site, each time guessing a different part of the secret (e.g., CSRF token). By analyzing the size of the compressed responses, the attacker can determine if their guess was correct.

- Analyze the response size: The length of the HTTP response reveals whether the attacker's guess was correct. If the size of the response decreases, it's likely that the guessed character is correct because of the way compression algorithms work. The attacker can then move on to the next character.

- Repeat the process: The attacker repeats step 4 and 5 until they have correctly guessed the entire secret.

Proposed Mitigation or Fix

- ✓ Disable HTTP-Level Compression: If feasible, consider disabling compression at the HTTP level. This helps eliminate the vulnerability altogether, although it may impact network bandwidth and performance.

- ✓ Separate Sensitive Information: Ensure that sensitive information, such as user credentials or session tokens, is not mixed with user input data. Keep them separate to minimize the potential for predictable patterns in the compressed responses.

- ✓ Protect Vulnerable Pages with CSRF Tokens: Implement Cross-Site Request Forgery (CSRF) protection mechanisms, such as CSRF tokens. These tokens help ensure the integrity of user requests and prevent attacks that exploit predictable response patterns.

- ✓ Utilize Same Site Cookie Attribute: Configure cookies with the Same Site attribute to mitigate the issue further. By setting the Same Site attribute, cookies belonging to the target website won't be sent along with requests that lack top-level navigation, reducing the risk of BREACH attacks initiated through invisible frames.

- ✓ Hide Traffic Length: Add a random number of bytes to the responses, effectively hiding the actual length of the traffic. This can make it more challenging for an attacker to infer information based on response size differences.

- ✓ Implement Rate Limiting: Apply rate limiting measures to restrict the number of requests that can be made to vulnerable pages within a specific time frame. For example, you can set a maximum limit of five requests per minute for a particular page, which helps mitigate the impact of repeated requests used in the BREACH attack.