Generated on 02 August 2023
Valid until 31 May 2024

Vulnerability Assessment Report

Prepared for Acme Co

Table of Contents

Introduction

Acme Co contracted GlitchSecure to conduct a Continuous Vulnerability Assessment of 1 externally facing web assets over the period of 1 year. The evaluation, which formally began on 01 June 2023, and is scheduled to conclude on 31 May 2024, aimed to thoroughly assess the security posture of the organisation's online presence.

Please note that the following report has been specifically prepared for demonstration and illustrative purposes. Consequently, it may not provide an exhaustive account of the technical particulars typically found within an authentic security assessment report.

Executive Summary

The primary objective of the evaluation was to identify potential security vulnerabilities and issues within a specific subset of the Acme Co infrastructure, with the aim of safeguarding the security and privacy of its users and overall system. To achieve this, a black box penetration test was performed, simulating the actions and strategies of a real-world adversary. By adopting this approach, the GlitchSecure team sought to gain comprehensive insights into defence mechanisms and identify any exploitable weaknesses.

All assessment activities were conducted in a manner that simulated an external malicious actor engaged in a targeted attack. The ultimate goal was to identify and exploit any existing security weaknesses that could allow a remote attacker to gain unauthorised access to organisational and customer data and systems. This assessment adhered to the recommendations and industry best practices outlined by The Open Web Application Security Project (OWASP). OWASP's framework covers various aspects, including but not limited to: input validation, session management, encryption, error handling, and secure coding practices.

Attack Narrative

Reconnaissance & Information Gathering

As with typical black box assessments, Acme Co provided minimal information regarding the existing infrastructure and technologies employed. This approach aimed to closely replicate a real-world attack where external actors lack internal knowledge. Information gathered about the targets came from a variety of sources and focused on identifying the software utilised, discovering open ports and services, and conducting file and directory enumeration.

Identifying Technologies Used

Fingerprinting was performed using a combination of manual source code review and Open Source tools.

Port Scanning & Service Identification

Port scanning was performed on all hosts within the scope with scanning covering a port range of 1-65535 across TCP and UDP.

The follow targets showed open ports:

- app.acme.tld port 443

Directory & File Enumeration

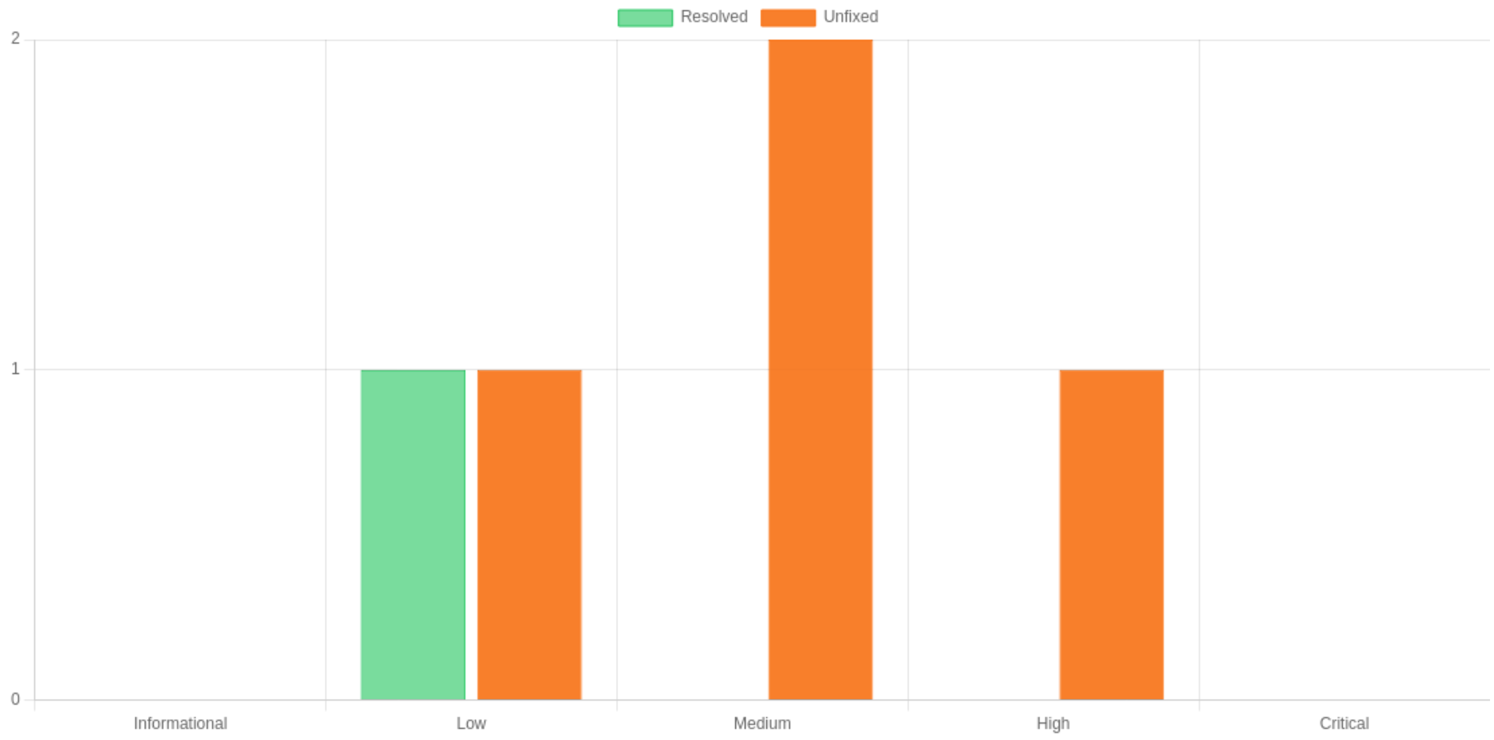
Directory enumeration was performed on all assets using predefined and customised word lists to help expand the scope and identify potentially sensitive information and additional targets.

Automated Scanning

Several automated scanning and testing tools were deployed on the previously found hosts to help ensure any potential findings were not missed. Results from automated tools were then processed and manually reviewed and tested to confirm the accuracy of the findings.

Findings

During the assessment, a total of 5 issues were identified. Of the findings, 1 is of high severity, 2 are of medium severity, 2 are of low severity, .



Asset List

This table presents an overview of the assets that were targeted during this assessment, along with the number of corresponding findings.

Asset #	Asset Location	Asset Type	Environment	Findings
ym3sai_1	app.acme.tld	DOMAIN	Production	5

Vulnerability & Findings List

The following lists contains summary information of vulnerabilities and findings identified during the assessment. Corresponding technical details can be found in the Vulnerability & Findings Details section.

Affected Asset	Finding	Severity
app.acme.tld	HTTP Response Splitting	MEDIUM
app.acme.tld	Exposed Environment Secrets and API Keys	LOW
app.acme.tld	HTTP Strict Transport Security Not Enabled	MEDIUM
app.acme.tld	Insecure TLS Protocols In Use	HIGH
app.acme.tld	Outdated JavaScript Libraries with Known Vulnerabilities	LOW

Vulnerability & Findings Details

HTTP Response Splitting

#1040_1 Reported by GlitchScan

● medium ● unfixed

Category:

Server-Side Injection -> HTTP Response Manipulation

CWE(s):

CWE-113: Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting')

CVSS 3.1 Base Score:

5.4 **(Medium) -** CVSS3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N

Affected Assets

`app.acme.tld`

Overview

The target application was found to be vulnerable to HTTP response splitting. This vulnerability occurs when user-supplied data is not properly filtered or sanitised, allowing the the content to be copied into a response header in an unsafe way.

Technical Details

The following demonstrates the ability to inject arbitrary HTTP response headers when visiting a specially crafted URL.

Request:

```
GET /%0ASet-Cookie%3Aglitchsecure=true; HTTP/2
Host: app.acme.tld
[...omitted for brevity...]
```

Response:

```
HTTP/2 302 Found
Date: Sat, 01 June 2023 22:07:08 GMT
Content-Type: text/html; charset=utf-8
Location: /not-found/
X-Frame-Options: SAMEORIGIN
Vary: Authorization, Cookie
Set-Cookie: sessionid=17f93vk90ddnz3120bna7gcb7jigaj23; expires=Thu, 02 June 2023 17:06:19 GMT; HttpOnly; Max-Age=604800; Path=/; SameSite=Lax
Set-Cookie: glitchsecure=true;
Set-Cookie: glitchsecure=true;
[...omitted for brevity...]
```

Severity Detail

It is possible for a remote attacker to inject custom HTTP headers into the HTTP response of a specially crafted URL. This could allow an attacker to inject session cookies, change response headers to bypass protections, or inject arbitrary HTML code.

Remediation Steps

- Avoid using user-controllable input in response headers.
- Ensure input containing any characters with ASCII codes less than (0x20) including CR(0x13) and LF(0x10) are properly encoded or filtered.
- Ensure all libraries that handle HTTP requests and responses are up-to-date.

References

OWASP: HTTP Response Splitting
CAPEC-34: HTTP Response Splitting

Exposed Environment Secrets and API Keys

#1040_2 Reported by GlitchScan

● low ● resolved

Category:

Sensitive Data Exposure -> Disclosure of Secrets

CWE(s):

CWE-526: Exposure of Sensitive Information Through Environmental Variables

CVSS 3.1 Base Score:

5.8 **(Medium) -** CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N

Affected Assets

`app.acme.tld`

Affected Locations

`/env.json`

Overview

A file was found on the target host which publicly reveals potentially sensitive information such as environment secrets and API keys. Publicly exposing secrets and API keys can result in account and service compromise, data exposure and misuse resulting in unexpected charges on third party services.

Technical Details

Request:

```
GET /env.json HTTP/1.1
Host: app.acme.tld
[...ommitted for brevity...]
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 01 June 2023 00:41:45 GMT
Content-Type: application/json
Connection: keep-alive
last-modified: Wed, 31 May 2023 13:09:34 GMT
Original-Content-Encoding: gzip
Content-Length: 1311

{
  "ENVIRONMENT": "production",
  "PORTAL_URL": "https://app.acme.tld",
  "AUTH0_DOMAIN": "login.acme.tld",
  "AUTH0_CLIENT_ID": "5m93U2uEcjiMVhUWokiNFoF",
  "AUTH0_MEMBERS_CLIENT_ID": "LxV9hWC3WTAghZKAiHZuwPX5fprXryY",
  "AUTH0_DIRECT_AUDIENCE": "https://app.acme.tld/api/",
  "AUTH0_MEMBERS_AUDIENCE": "https://app.acme.tld/api/graphql/v1",
  "SEGMENT_API_KEY": "hUWokiNFoFhQLzpb7KXsLxV9hW",
  "DATADOG_APPLICATION_ID": "b4760803-098e-44a2-8a86-9259e77",
  "DATADOG_CLIENT_TOKEN": "pubce0b3d23ff5557f597d9277c",
  "ACP_REDIRECT_URL": "https://learningmanager.adobe.com/oauth/o/authorize?client_id=48ae3ba3-abad-1337-8123-
86ede1c64a17&redirect_uri=https://app.acme.tld/acp-oauth-
callback&scope=learner:read,learner:write&response_type=CODE&account=1337&logoutAfterAuthorize=true",
  "CONTENTFUL_ACCESS_TOKEN": "atYdQQSeC_GbEJvthEEsg9Ywom8-wRH7pDH0a-lgdF0",
  "CONTENTFUL_PREVIEW_ACCESS_TOKEN": "5zJ0A98S_I0yBk1cO0Bmd_gDH3_MWxcYg6luvoS8pmc",
  "CONTENTFUL_URL": "https://graphql.contentful.com/content/v1/spaces/acmeco",
  "FEATURE_FLAGS": {
    "CE_CREDITS": true,
    "HOME_OFFICE_CSV_EXPORT": true,
  }
}
```

Severity Detail

The GlitchSecure team manually verified each exposed API key and secret. Of note, the `CONTENTFUL_ACCESS_TOKEN` exposed API secret allowed the team to access the full GraphQL API within the context of the `acmeco` space provided by Contentful. Within this GraphQL API, the team was able to run introspection queries and access all data contained there. While the Acme Inc team indicated that data available through the linked GraphQL API is publicly accessible content, the site in which it originates from is not ordinarily available to unauthenticated users which increases the potential risk. All other exposed secrets found did not have an increased risk on confidentiality or integrity as each was verified to have an exception of public or client-side use.

Remediation Steps

- Review exposed content in the impacted Contentful resource to verify the absence of potentially confedntial information.
- Ensure that access to client-side tokens is placed behind the same authentication requirements as other are of the impacted host.
- Reduce the reliance and exposure of API keys and secrets even ones expected to be used by clients whenever technically possible.

References

- Contentful GraphQL API Documentaion
- OWASP Secrets Management Cheat Sheet

HTTP Strict Transport Security Not Enabled

#1040_3 Reported by GlitchScan

● medium ● unfixed

Category:

Server Security Misconfiguration -> Lack of Security Headers

CWE(s):

CWE-319: Cleartext Transmission of Sensitive Information CWE-523: Unprotected Transport of Credentials

CVSS 3.1 Base Score:

6.8 **(Medium) -** CVSS3.1/AV:C/AC:V/PR:S/UI:S/S:3/C:1/I:A/A:V

Affected Assets

`app.acme.tld`

Overview

A misconfiguration on the target host reveals the HTTP Strict Transport Security (HSTS) header is not properly implemented. As the targets do not use HSTS, it may allow man-in-the-middle (MITM) attacks.

HTTP headers are well known and its implementation can make your application more secure. HTTP Strict Transport Security (HSTS) is a web security policy mechanism which helps to protect websites against protocol downgrade attacks and cookie hijacking. It allows web servers to declare that web browsers (or other complying user agents) should only interact with it using secure HTTPS connections and never via the insecure HTTP protocol. HSTS is an IETF standards track protocol and is specified in RFC 6797. A server implements an HSTS policy by supplying a header (Strict-Transport-Security) over an HTTPS connection (HSTS headers over HTTP are ignored).

Technical Details

```
Testing HTTP header response @ "/"


HTTP Status Code             200 OK
HTTP clock skew              0 sec from localtime
Strict Transport Security    not offered
Public Key Pinning           --
Server banner                cloudflare
Application banner           --
Cookie(s)                    (none issued at "/")
```

Severity Detail

The impacted domain does not use HSTS, which may allow man-in-the-middle (MITM) attacks.

If an attacker is able to intercept traffic of a user visiting the impacted domain they would be able to force the target user to only interact with the site over insecure HTTP. Doing this would allow the attacker to inject malicious content or intercept all user information including passwords, OTP codes, and other sensitive data.

Remediation Steps

For the strongest protection and to prevent protocol downgrade attacks, it is advised that you follow best practices by enabling the HSTS banner on all affected hosts.

Additionally, it is recommended to set the max-age value to at least 15768000 seconds (6 months) and ideally to 31536000 (one year).

References

- RFC 6797 - HTTP Strict Transport Security (HSTS)
- Test HTTP Strict Transport Security (OTG-CONFIG-007)
- Enforce Web Policy with HTTP Strict Transport Security (HSTS)
- The Importance of a Proper HTTP Strict Transport Security Implementation

Insecure TLS Protocols In Use

#1040_4 Reported by GlitchScan

● high ● unfixed

Category:

Server Security Misconfiguration -> Insecure SSL

CWE(s):

CWE-326: Inadequate Encryption Strength

CVSS 3.1 Base Score:

6.5 **(Medium) -** CVSS3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N

Affected Assets

`app.acme.tld`

Overview

During testing, it was found that the target offers the outdated and insecure TLS v1.0 and v1.1 protocols.
As of March 25, 2021, the Internet Engineering Task Force (IETF) released RFC8996, which formally deprecated the use of TLS v1.1. Additionally, PCI DSS forbids the use of legacy protocols such as TLS 1.0 and recommends users to adopt protocol TLS 1.2+.

Technical Details

```
Testing protocols via sockets except NPN+ALPN


SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   not offered
ALPN/HTTP2 h2, http/1.1 (offered)
```

Severity Detail

The use of depreciated and known insecure encryption protocols posses a risk to the integrity and confidentiality of any data or communications sent to this end-point.

Remediation Steps

Best practices outlined in RFC-7525 give reasons why it is discouraged to use protocol TLS 1.0 and TLS 1.1.
As of March 25, 2021, the Internet Engineering Task Force (IETF) released RFC8996, which formally deprecated the use of TLS v1.1.
Additionally PCI DSS forbids the use of legacy protocols such as TLS 1.0 and recommends users to adopt protocol TLS 1.2+.

References

- RFC 8996 - Deprecating TLS 1.0 and TLS 1.1
- RFC 7525 - Recommendations for Secure Use of Transport Layer Security (TLS)
- PCI SSC - Migrating from SSL and Early TLS

Outdated JavaScript Libraries with Known Vulnerabilities

#1040_5 Reported by GlitchScan

● low ● unfixed

Category:

Using Components with Known Vulnerabilities -> Outdated Software Version

CWE(s):

CWE-1104: Use of Unmaintained Third Party Components

CVSS 3.1 Base Score:

6.5 **(Medium) -** CVSS3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Affected Assets

`app.acme.tld`

Affected Locations

`/assets/modernizr.js`

Overview

It was observed that the target web application utilises outdated javascript libraries, including ones with known vulnerabilities. Reliance on components that are no longer maintained can make it difficult or impossible to fix significant bugs, vulnerabilities, or quality issues. In effect, unmaintained code can become obsolete.

Technical Details

The following outdated javascript libraries were identified:

jQuery 3.3.1:

- All pages of `app.acme.tld` via `https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js`

Modernizr 3.11.7:

- `https://app.acme.tld/assets/modernizr.js`

Severity Detail

jQuery version 3.3.1 released in Janaury 2018 is known to be impacted by the following vulnerabilities:

- CVE-2019-11358: jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution
- CVE-2020-11022: Regex in its jQuery.htmlPrefilter sometimes may introduce XSS
- CVE-2020-11023: Regex in its jQuery.htmlPrefilter sometimes may introduce XSS

Additional outdated libraries noted were not observed to have outdated vulnerabilties.

While the inclusion of outdated libraries does not always directly result in an exploitable condition or vulnerability, if certain parameters are met, they may result in flaws such as cross-site scripting.

Remediation Steps

- Upgrade the affected version of JQuery to the latest version (3.6.4 as of 08 March 2023).
- Upgrade the affected version of Modernizer to the latest version (3.12.0 as of 15 Feb 2022).
- Develop a patch-management strategy to ensure that security updates are promptly applied to all third-party libraries in your application.
- Consider reducing your attack surface by removing any libraries that are no longer in use.

References

- OWASP Top 10: A06:2021 – Vulnerable and Outdated Components
- jQuery 3.6.4 Release Note
- CVE-2019-11358
- CVE-2020-11022
- CVE-2020-11023

Conclusion & General Comments

Overall, the security level of the Acme Co application was deemed fair, exhibiting only a limited number of vulnerabilities. However, it is strongly recommended that additional testing of related assets should be conducted to identify similar potential issues. Furthermore, it is advised to explore supplementary testing of pivot points into the internal network.
Additionally, the following points should be considered:

- Acme Co should continue to implement a consistent patch management cycle to include plugins and third-party libraries in use on all sites and infrastructure.
- Acme Co should provide public information for the desired point of contact and the process of reporting future issues. The transparency helps foster a collaborative environment and allows for the assistance of other potential researchers who find issues.
- Acme Co should consider performing additional penetration testing.

Document Change Log

Version	Date	Comment
v1	16 May 2023	Initial Report

Note: Document change log does not reflect updates to finding statuses as these are rendered dynamicly when downloading the report.
This PDF was generated and downloaded from the GlitchSecure platform on 02 August 2023 00:33 UTC.