Report 02

URL: datacamp.com

# *Out-of-date Version (Lodash)*

**CRITICAL**

Lodash is a widely used JavaScript utility library that provides a collection of helper functions and utilities for simplifying common programming tasks. It offers a wide range of functions that assist with data manipulation, array and object handling, string manipulation, functional programming, and more. However, using an outdated version of Lodash can introduce security vulnerabilities and potential weaknesses that may be exploited by attackers. Outdated versions of Lodash may contain known security vulnerabilities that have been addressed in newer releases. Attackers can exploit these vulnerabilities to execute arbitrary code, perform remote code execution, or compromise the integrity of the application.

- *Identified Version - 4.17.4*
- *Latest Version - 4.17.21*

- Prototype Pollution (CVE-2019-10744)

  Versions lower than 4.17.12 are vulnerable to Prototype Pollution. The function ***defaultsDeep*** can be tricked into adding or modifying properties of ***Object.prototype*** using a constructor payload.

  CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:H

## Affected Components

- Application Code: The web application's codebase that utilizes the Lodash library is directly affected by the vulnerabilities present in the outdated version. This includes any parts of the code that call Lodash functions or rely on its functionalities.

- User Inputs: If the web application allows user inputs that interact with Lodash functions, such as passing user-supplied data as arguments to Lodash methods, the vulnerability can potentially be triggered through these user inputs.

- Data Manipulation: Any data manipulation or transformation processes that involve Lodash functions can be impacted by the vulnerabilities. This includes operations such as merging objects, deep cloning, filtering, sorting, and other data manipulation tasks performed using Lodash methods.

## Impact Assessment

The use of an outdated version of Lodash in a web application can have several potential impacts on the security, functionality, and stability of the application.

- Increased Security Risks: Outdated versions of Lodash may contain known vulnerabilities that have been addressed in newer releases. These vulnerabilities can be exploited by attackers to compromise the application's security, leading to unauthorized access, data breaches, injection attacks, or other malicious activities.

- Exposure to Exploits: Using an outdated version of Lodash exposes the application to potential exploits that target the specific vulnerabilities present in the outdated version. Attackers may leverage these exploits to gain control over the application, manipulate data, or execute arbitrary code within the application's context.

- Lack of Security Patches: An outdated version of Lodash may lack the latest security patches and bug fixes. This means that any security vulnerabilities discovered in the library since the version being used are not addressed, leaving the application exposed to potential threats and increasing the likelihood of successful attacks.

- Missed Performance Enhancements: Updated versions of Lodash may include performance optimizations and efficiency improvements. By sticking with an outdated version, the application may miss out on these enhancements, leading to suboptimal performance, slower execution, or inefficient resource utilization.

- Inability to Leverage New Features: Updates to Lodash often introduce new features, capabilities, and programming conveniences. By using an outdated version, the application may be limited in utilizing these new features, potentially hindering development productivity, code readability, and maintainability.

## Steps to Reproduce

- Identify the current version of Lodash being used in the target web application. This information can typically be found in the application's dependencies or configuration files.

- Determine the latest stable version of Lodash available by referring to the official documentation, release notes, or the Lodash GitHub repository.

- Set up a test environment to replicate the target web application, ensuring it matches the configuration and dependencies of the original application as closely as possible.

- Install the outdated version of Lodash identified in step 1 in the test environment.

- Identify any specific features, functions, or areas of the web application that heavily rely on Lodash.

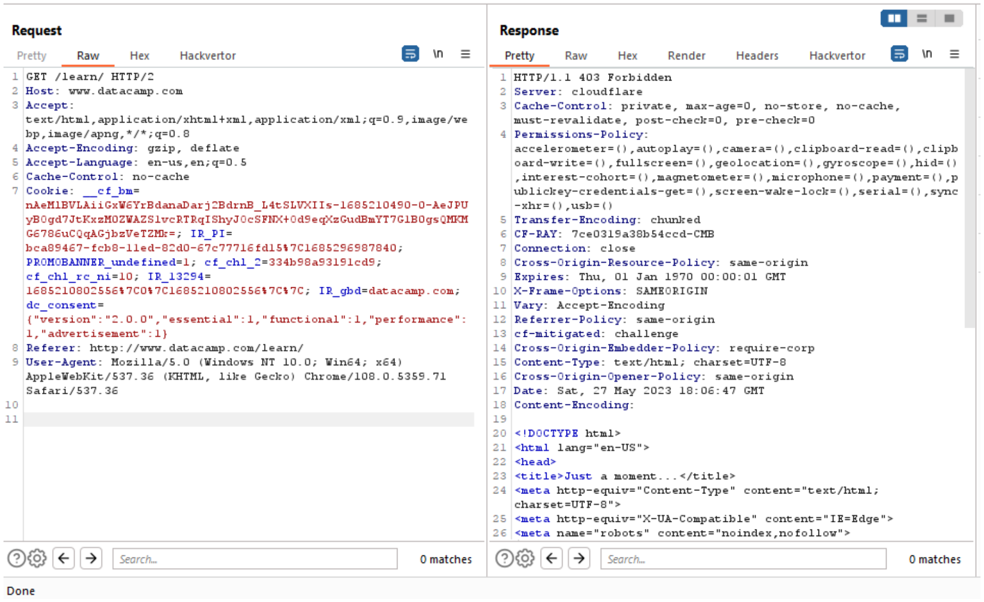
- Perform actions or trigger events within the test environment that utilize the Lodash functions identified in step 5. This can involve interacting with user inputs, performing data manipulation operations, or any other functionality that relies on Lodash.

- Observe and document any unexpected behavior, errors, crashes, or security vulnerabilities that occur during the test.

- Compare the observed results with the expected behavior and functionality of the web application.

- Repeat the test using the latest stable version of Lodash obtained in step 2.

## Proof of Concept



```
GET /learn/ HTTP/2
Host: www.datacamp.com
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/we
bp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: __cf_bm=
nAeMlBVLAiiGxW6YrBdanaDarj2BdrnB_L4tSLVXIIs-1685210490-0-AeJPU
yB0gd7JtKxzMOZWAZSlvcRTRqIShyJ0cSFNX+0d9eqXzGudBmYT7GlB0gsQMKM
G6786uCQqAGjbzVeTZMk=; IR_PI=
bca89467-fcb8-11ed-82d0-67c77716fd15%7C1685296987840;
PROMOBANNER_undefined=1; cf_chl_2=334b98a93191cd9;
cf_chl_rc_ni=10; IR_13294=
1685210802556%7C0%7C1685210802556%7C%7C; IR_gbd=datacamp.com;
dc_consent=
{"version":"2.0.0","essential":1,"functional":1,"performance":
1,"advertisement":1}
Referer: http://www.datacamp.com/learn/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.71
Safari/537.36


```

```
HTTP/1.1 403 Forbidden
Server: cloudflare
Cache-Control: private, max-age=0, no-store, no-cache,
must-revalidate, post-check=0, pre-check=0
Permissions-Policy:
accelerometer=(),autoplay=(),camera=(),clipboard-read=(),clipb
oard-write=(),fullscreen=(),geolocation=(),gyroscope=(),hid=()
,interest-cohort=(),magnetometer=(),microphone=(),payment=(),p
ublickey-credentials-get=(),screen-wake-lock=(),serial=(),sync
-xhr=(),usb=()
Transfer-Encoding: chunked
CF-RAY: 7ce0319a38b54ccd-CMB
Connection: close
Cross-Origin-Resource-Policy: same-origin
Expires: Thu, 01 Jan 1970 00:00:01 GMT
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Referrer-Policy: same-origin
cf-mitigated: challenge
Cross-Origin-Embedder-Policy: require-corp
Content-Type: text/html; charset=UTF-8
Cross-Origin-Opener-Policy: same-origin
Date: Sat, 27 May 2023 18:06:47 GMT
Content-Encoding:

<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Just a moment...</title>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
<meta name="robots" content="noindex,nofollow">
```

## Proposed Mitigation or Fix

- ✓ Upgrade Lodash: Update the installation of Lodash to the latest stable version. This ensures that known vulnerabilities are addressed, and the latest bug fixes, security patches, performance improvements, and features are incorporated into the application.

- ✓ Regularly Update Dependencies: Keep all dependencies, including Lodash, up to date. Regularly check for updates and apply them to ensure that your application benefits from the latest security patches and improvements. Utilize tools like package managers and automated dependency update services to streamline the update process.

- ✓ Monitor Security Advisories: Stay informed about security advisories, vulnerability reports, and patches related to Lodash. Subscribe to security mailing lists, follow official Lodash channels, or use vulnerability databases to receive notifications about any vulnerabilities or updates specific to Lodash. Promptly apply relevant patches and fixes to address security issues.

- ✓ Conduct Regular Security Audits: Perform periodic security audits of your application, including an assessment of Lodash usage. Use static code analysis tools, security scanners, and manual code reviews to identify any security weaknesses or vulnerabilities related to Lodash or other dependencies. Address any identified issues promptly.