Sri Lanka Institute of Information Technology

IE2062

Web Security

Bug Bounty Report X

Submitted by:

Student Registration Number	Student Name
IT21197550	Nihila Premakanthan

Date of Submission: 28.05.2023

Acknowledgement

I would like to express my special thanks to our mentor Ms. Chethana Liyanapathirana and Dr. Lakmal Rupansighe for their time and efforts she provided throughout the course, and for the Web Security lecture panel for guiding us through this semester and for helping us by giving examples, guidelines, and advice about the project. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

Executive Summary

This report aims to provide an overview of the vulnerability identified in a particular domain. The bug bounty platform called Hackerone was used for this purpose. This report analyses the domain of http://aweb.suivo.com/

This report uses different tools to gather information detect vulnerabilities and perform penetration testing. The tool name Netsparker and Owasp Zap was mainly used to identify the vulnerability. Further this report provides the vulnerability title, vulnerability description, Affected Components, Impact Assessment, Steps to reproduce the vulnerability, proof of concept and the proposed mitigation.

By including these comprehensive details for each vulnerability, the report provides a comprehensive overview of the security weaknesses present within the system and offers actionable insights for remediation and improvement.

Contents

Vulnerability title

Stack Trace Disclosure

Security Level:

Risk Level:
MEDIUM

Affected URL: http://aweb.suivo.com/

Vulnerability Description

The term "Stack Trace Disclosure" is a security flaw that happens when a system or application unintentionally exposes its stack trace data to an attacker or other unauthorized users. The execution path and order of the functions or procedures used to reach a specific point in the code are described in detail by a stack trace, a useful troubleshooting tool.

Typically, a stack trace is produced as part of an application's error handling process whenever an error or exception occurs. The functions or methods that were called, along with the accompanying memory addresses and parameters, are detailed in this stack trace. It aids in program debugging and locating the error's root cause.

Affected components

A system or application's numerous components may be impacted by stack trace disclosure. The following are some of the elements that may be impacted:

- Implementing error handling techniques incorrectly: This can result in the leaking of the stack trace. This applies to circumstances when detailed error messages are enabled in a production environment or if error messages are not appropriately cleaned.
- User interfaces: When error messages are displayed to users, in particular, stack trace information may unintentionally leak through user interfaces. Stack trace exposure might occur if the user interface is not correctly constructed to process and show error messages safely.
- Logging Mechanisms: Stack traces may be included in application logs that record errors or exceptions. Stack trace leakage may result if an attacker acquires access to these logs or if unauthorized users have access to them. Implementing logging tools securely is important, along with appropriate access controls and encryption as needed.

- APIs and Web Services: If error responses are not handled properly, stack trace information may be made available through APIs or Web Services. An attacker may be able to obtain error messages or stack traces that are provided in API responses and learn more about how the application functions.
- Server Configuration: An incorrect server configuration may result in the leaking of a stack trace. For instance, the server may unintentionally reveal stack trace information in server responses if it is not set up properly to handle and conceal comprehensive error messages.
- Environment for Development: In some circumstances, stack trace disclosure may take place right during the development process. Stack traces may become visible during testing or debugging if development tools or environments are not correctly setup to handle faults or exceptions securely.

Impact assessment

Stack Trace Disclosure can have a big impact on the integrity and security of a system or application. Here are some possible effects and dangers linked to this vulnerability:

- Exposure of Sensitive Information: The release of stack trace data may reveal sensitive information about the inner workings of the application, such as function names, file paths, variable values, and possibly even database queries. Attackers can use this information to better understand the architecture of the application and create more specialized attacks.
- Increased Attack Surface: Stack trace disclosure increases an application's attack surface by giving attackers useful information about conceivable flaws and vulnerabilities. With this information, attackers are better equipped to locate and take advantage of application security holes.
- Information Gathering: Stack traces can be a useful tool for attackers to learn more about the target system and gather intelligence for further attacks. Attackers can learn about the application's dependencies, frameworks, and libraries, as well as possibly find other entry points for exploitation, by inspecting the stack trace.
- Exploitation of Vulnerabilities: Stack trace data may show coding flaws, incorrect setups, or vulnerable application components. Attackers can use this knowledge to take advantage of known flaws or use more advanced attack strategies to breach the security of the system.
- System Compromise: The revelation of a stack trace may serve as a springboard for additional assaults that compromise the entire system. Attackers can increase their privileges, acquire unauthorized access, exfiltrate sensitive data, or even run malicious code within the environment once they have found gaps or vulnerabilities.

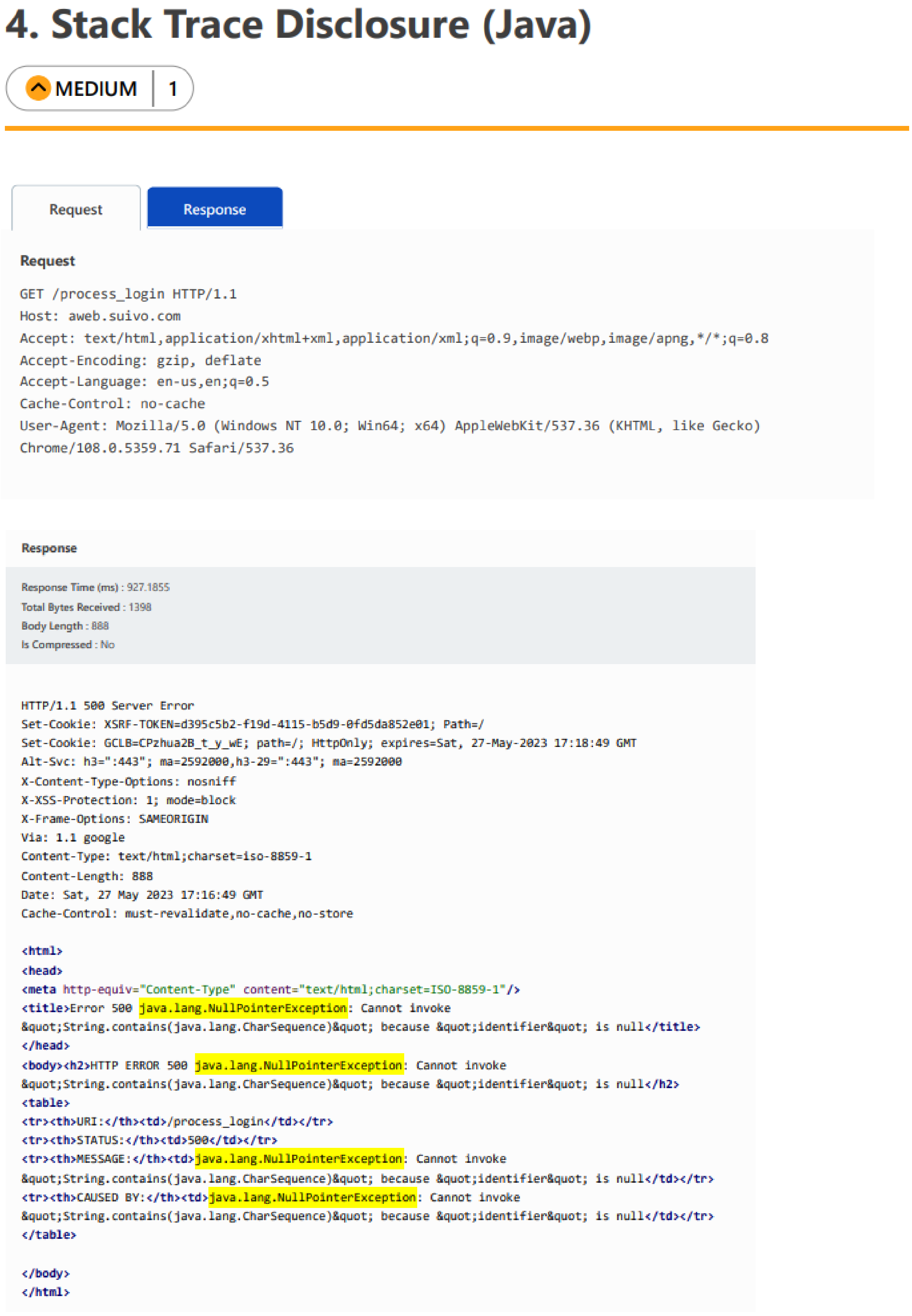
- Reputational Damage: Organizations may suffer serious reputational damage as a result of the disclosure of stack trace data. Users' faith and confidence in the security of the application are undermined, which could result in lost sales, legal repercussions, or harm to the organization's reputation.

Steps to reproduce

Determine the intended application: Choose a program or system where you think there might be a Stack Trace Disclosure vulnerability. It can be a program you're creating for educational purposes or a well-known open-source undertaking.

- Create an error or exception by purposefully setting one off within the application. Incorrect input, unexpected conditions, or purposeful code flow manipulation can all accomplish this.
- The error answer is as follows: Note any error notifications or error correction techniques that are brought on by the error or exception. Watch for indications that the application may be logging or showing stack trace data.
- Analyze messages: Examine error messages that are displayed to users or that are recorded in system logs. Make sure the error messages don't contain any sensitive information like function names, file paths, or other specific stack trace information.
- Reconfirm the disclosure: Check to see if attackers or unauthorized users may access the stack trace data. This can be achieved by viewing error pages or making an effort to use the information that has been made public.

Proof of concept



4. Stack Trace Disclosure (Java)

MEDIUM | 1

Request

```
GET /process_login HTTP/1.1
Host: aweb.suivo.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
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
```

Response

Response Time (ms) : 927.1855
Total Bytes Received : 1398
Body Length : 888
Is Compressed : No

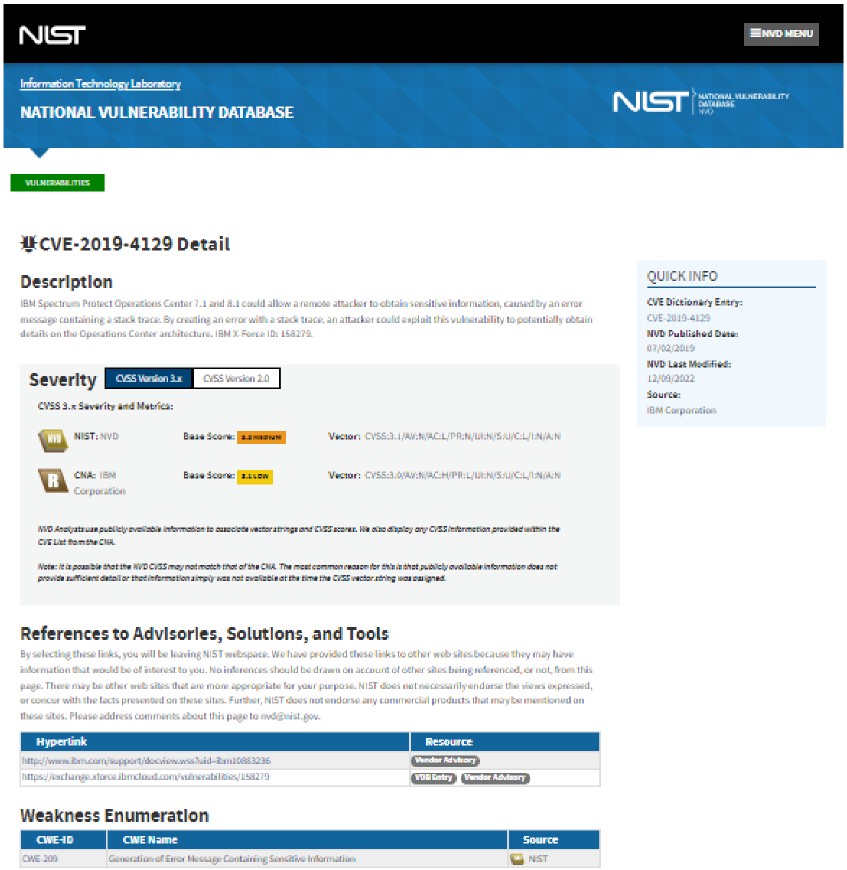
```
HTTP/1.1 500 Server Error
Set-Cookie: XSRF-TOKEN=d395c5b2-f19d-4115-b5d9-0fd5da852e01; Path=/
Set-Cookie: GCLB=CPzhua2B_t_y_wE; path=/; HttpOnly; expires=Sat, 27-May-2023 17:18:49 GMT
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Via: 1.1 google
Content-Type: text/html;charset=iso-8859-1
Content-Length: 888
Date: Sat, 27 May 2023 17:16:49 GMT
Cache-Control: must-revalidate,no-cache,no-store

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1"/>
<title>Error 500 java.lang.NullPointerException: Cannot invoke
&quot;String.contains(java.lang.CharSequence)&quot; because &quot;identifier&quot; is null</title>
</head>
<body><h2>HTTP ERROR 500 java.lang.NullPointerException: Cannot invoke
&quot;String.contains(java.lang.CharSequence)&quot; because &quot;identifier&quot; is null</h2>
<table>
<tr><th>URI:</th><td>/process_login</td></tr>
<tr><th>STATUS:</th><td>500</td></tr>
<tr><th>MESSAGE:</th><td>java.lang.NullPointerException: Cannot invoke
&quot;String.contains(java.lang.CharSequence)&quot; because &quot;identifier&quot; is null</td></tr>
<tr><th>CAUSED BY:</th><td>java.lang.NullPointerException: Cannot invoke
&quot;String.contains(java.lang.CharSequence)&quot; because &quot;identifier&quot; is null</td></tr>
</table>

</body>
</html>
```

Proposed mitigation

It is necessary to adhere to safe coding guidelines like these to reduce the risk of stack trace disclosure:

- Apply appropriate error handling: Make sure error messages are handled securely and refrain from showing users critical information. Error messages shouldn't include specific stack trace information and should instead be generic.
- Disable detailed error warnings in environments intended for production: Configure the application to log the detailed stack trace internally for troubleshooting while displaying generic error messages to users.
- Secure logging procedures: Make sure that application logs including stack trace data are securely saved and that only authorized individuals have access to them. Examine and keep an eye on log files frequently for any indications of unauthorized access.
- Validate and sanitize user input: To ward against injection attempts that can result in the triggering of exceptions and the exposure of stack trace data.

External Reference


NIST
≡NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NIST NATIONAL VULNERABILITY DATABASE NVD
VULNERABILITIES

CVE-2019-4129 Detail

Description

IBM Spectrum Protect Operations Center 7.1 and 8.1 could allow a remote attacker to obtain sensitive information, caused by an error message containing a stack trace. By creating an error with a stack trace, an attacker could exploit this vulnerability to potentially obtain details on the Operations Center architecture. IBM X-Force ID: 158279.

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD Base Score: [illegible] MEDIUM Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

CNA: IBM Corporation Base Score: [illegible] LOW Vector: CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:N/A:N

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: It is possible that the NVD CVSS may not match that of the CNA. The most common reason for this is that publicly available information does not provide sufficient detail or that information simply was not available at the time the CVSS vector string was assigned.

QUICK INFO

CVE Dictionary Entry:
CVE-2019-4129
NVD Published Date:
07/02/2019
NVD Last Modified:
12/09/2022
Source:
IBM Corporation

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
http://www.ibm.com/support/docview.wss?uid=ibm10883236	Vendor Advisory
https://exchange.xforce.ibmcloud.com/vulnerabilities/158279	VDB Entry Vendor Advisory

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-209	Generation of Error Message Containing Sensitive Information	NIST