Sri Lanka Institute of Information Technology

IE2062

Web Security

Bug Bounty Report V

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 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

Password Transmitted over a Query String

Severity Level

Risk Level:
MEDIUM

Vulnerability Description

When a password or other sensitive information is sent as part of the URL in the query string parameters, it is referred to as "password transmitted over a query string." The portion of a URL that follows the question mark ("?") and contains key-value pairs of the format "key=value" is known as the query string.

Because query strings are frequently recorded in multiple places, such as server logs, proxy logs, and browser histories, such as these, transmitting passwords in this way is regarded as a security issue. Additionally, if shared or stored incorrectly, they can be quickly captured by network sniffers or accessed by unwanted parties.

Passwords should not be transmitted over query strings for a number of reasons:

- Passwords included in query strings are frequently logged in a variety of logs created by web servers, apps, or proxies. The possibility of unauthorized people accessing these logs and their potential long-term retention raise the danger of password compromise.
- URL visibility, the address bar, bookmarks, and history of most browsers display URLs that contain query strings. An attacker can readily harvest the credentials from any of these sources and utilize them for their own evil purposes.
- Links that are shared with others becomes hazardous when passwords are present in the query string. Even though the password was intended to remain private, it may be accessible to everyone with access to the URL if someone mistakenly discloses it.
- Transport layer security, since query strings are sent along with the URL, they are encrypted using the same transport layer security (TLS) protocol as the URL as a whole. The query string could, however, be displayed in plain text elsewhere if the URL is copied or bookmarked, which raises the risk of unwanted access.

Affected Components

Password transmission over a query string vulnerability affects the following components:

- Client-side code: The client-side code is in charge of creating the URL with the query string containing the password. This could be browser-based JavaScript code or any other client-side technology that creates URLs.
- Server-side code: The server-side code handles the request after receiving the URL and the query string that contains the password. It could potentially expose information if it records the incoming request, logs the URL, or handles the password improperly.
- Networking Devices: Routers, switches, proxies, and any other networking devices that help with data transfer through networks are all considered to be part of the network infrastructure. The password can be extracted if the network is compromised or accessed by unauthorized individuals, who can then intercept the URL with the query string.
- Logging programs: Passwords contained in URLs with query strings can be recorded by server logs, proxy logs, and other logging programs. The passwords may be made public if these logs are improperly secured, viewed by unauthorized people, or kept for an extended amount of time.
- End-user devices: End-user devices, such as laptops, mobile phones, or tablets, maintain bookmarks and browsing histories that may include URLs with passwords sent over query strings. The passwords can be retrieved if these devices are compromised or accessed by unauthorized people.

Impact Assessment

Password transmission over a query string vulnerability can have a severe negative effect and result in several security concerns and repercussions, including:

- Password exposure: Passwords sent over query strings in plain text are vulnerable to unwanted access. Attackers can quickly extract the password from the query string and use it to obtain unauthorized access to user accounts, sensitive data, or other systems if they manage to intercept or acquire access to the query string.

- Account breach: If an attacker gains access to a user's password, they may be able to compromise the account that goes with it. Unauthorized access to private information, financial information, secret documents, or other delicate resources may result from this.
- Data Breaches: Passwords transmitted over query strings that are intercepted by attackers may result in data breaches. Attackers may use the stolen credentials to access larger networks, databases, or systems, exposing or stealing private information belonging to people or businesses.
- Credential reuse: If users use the same password for several accounts, the compromise of just one password sent over a query string can have far-reaching consequences. Attackers may try to access additional accounts using the stolen password, which could result in unlawful access to a number of online services or systems.

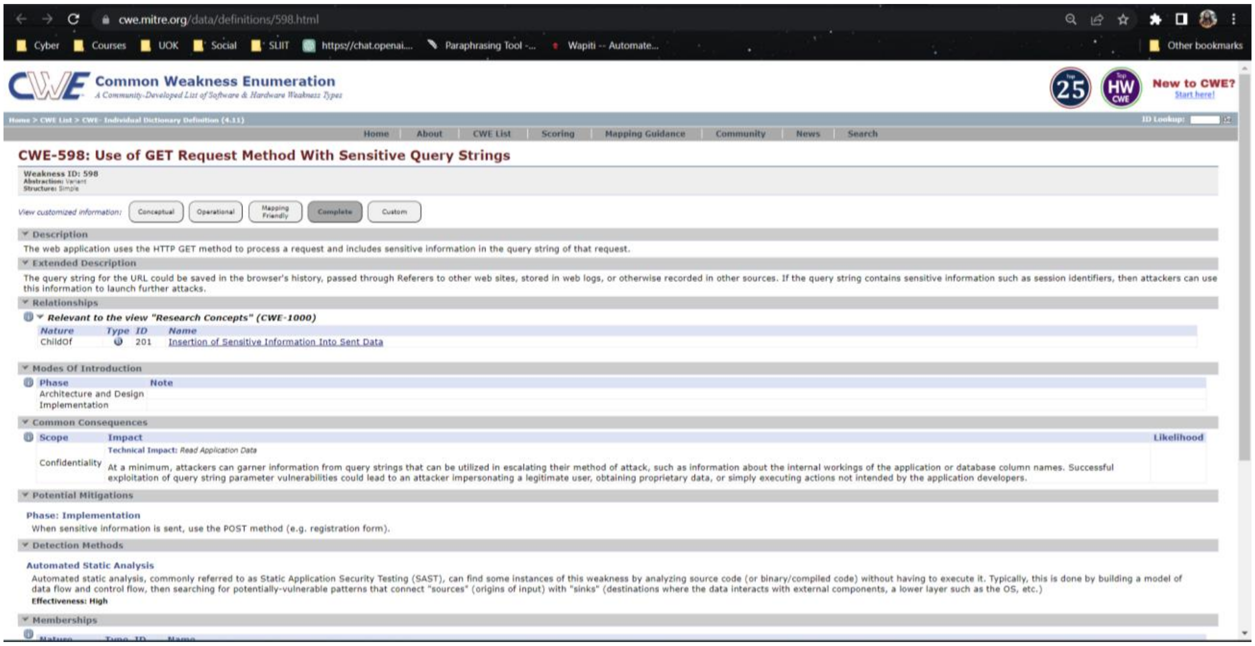
Proposed mitigation

Consider the following best practices to lessen the risk of sending passwords over a query string:

- Use secure transmission protocols: To make sure that passwords and other sensitive data are sent across encrypted channels, such as HTTPS. This lessens the chance of interception by encrypting the communication between the client and server.
- Utilize HTTP POST requests: To convey sensitive data, utilize HTTP POST requests rather than include passwords in query strings. The data is transmitted in the request body for POST requests, which is not directly visible in the URL or recorded in the server logs.
- Employ reliable authentication techniques: To increase the security of user accounts, employ reliable authentication techniques like token-based authentication or multi-factor authentication (MFA). Because of this, even if passwords are intercepted, illegal access is largely avoided.
- Password hashing and salting: Use powerful hashing algorithms (like bcrypt, Argon2, or scrypt) along with a different salt for each user when storing passwords. Even if an attacker is able to obtain the hashes that are saved, hashing makes it very challenging for them to recover the original password.
- Passwords shouldn't be kept in logs: Make sure that sensitive data, such as passwords, is not logged or saved in plaintext or another format that may be easily retrieved. To prevent unwanted access, implement logging procedures that exclude sensitive data, and periodically evaluate and protect log storage.

- Informing users Increase user awareness about safe password handling procedures. Encourage them to practice proper password hygiene, which includes using strong and distinct passwords for each account, and to refrain from sharing passwords through unsafe methods, such as query strings.
- Implement security testing: To find and fix potential vulnerabilities in your system, undertake regular security assessments, such as vulnerability scanning and penetration testing. This makes it more likely that password-related problems will be found and fixed.
- Follow coding practices: Make that developers adhere to secure coding standards, which should include input validation, output encoding, and secure handling of sensitive data. To reduce the risk of vulnerabilities, implement security frameworks and controls like the OWASP (Open Web Application Security Project) recommendations.

External Reference



CWE-598: Use of GET Request Method With Sensitive Query Strings

Weakness ID: 598
Abstraction: Variant
Structure: Simple

Description

The web application uses the HTTP GET method to process a request and includes sensitive information in the query string of that request.

Extended Description

The query string for the URL could be saved in the browser's history, passed through Referers to other web sites, stored in web logs, or otherwise recorded in other sources. If the query string contains sensitive information such as session identifiers, then attackers can use this information to launch further attacks.

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	●	201	Insertion of Sensitive Information Into Sent Data

Modes Of Introduction

Phase	Note
Architecture and Design	
Implementation	

Common Consequences

Scope	Impact	Likelihood
Confidentiality	Technical Impact: Read Application Data At a minimum, attackers can garner information from query strings that can be utilized in escalating their method of attack, such as information about the internal workings of the application or database column names. Successful exploitation of query string parameter vulnerabilities could lead to an attacker impersonating a legitimate user, obtaining proprietary data, or simply executing actions not intended by the application developers.	

Potential Mitigations

Phase: Implementation

When sensitive information is sent, use the POST method (e.g. registration form).

Detection Methods

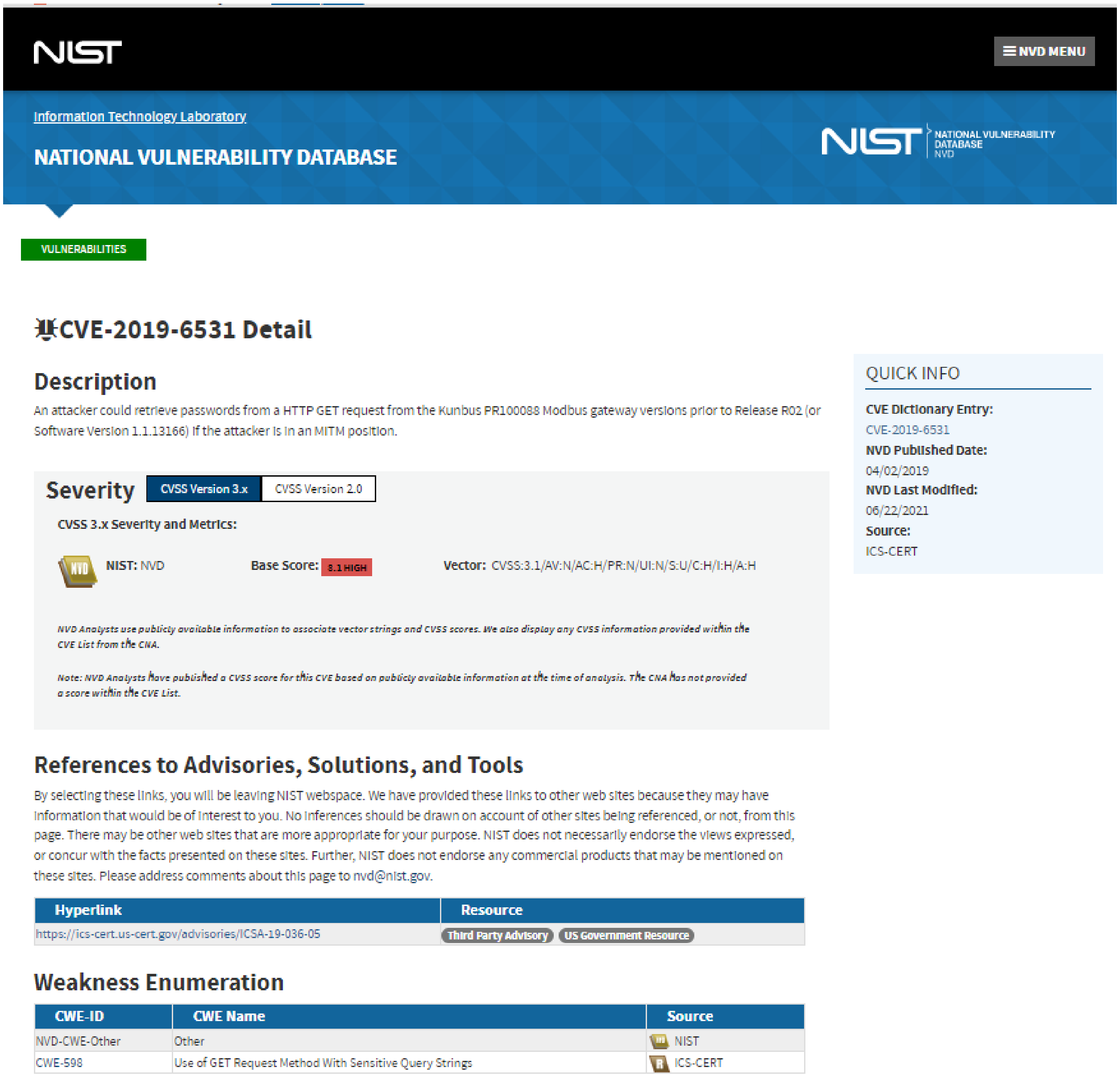
Automated Static Analysis

Automated static analysis, commonly referred to as Static Application Security Testing (SAST), can find some instances of this weakness by analyzing source code (or binary/compiled code) without having to execute it. Typically, this is done by building a model of data flow and control flow, then searching for potentially-vulnerable patterns that connect "sources" (origins of input) with "sinks" (destinations where the data interacts with external components, a lower layer such as the OS, etc.)

Effectiveness: High

Memberships

Nature Type ID Name



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

CVE-2019-6531 Detail

Description

An attacker could retrieve passwords from a HTTP GET request from the Kunbus PR100088 Modbus gateway versions prior to Release R02 (or Software Version 1.1.13166) if the attacker is in an MITM position.

Severity

CVSS Version 3.x
CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 8.1 HIGH **Vector:** CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

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: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

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
https://ics-cert.us-cert.gov/advisories/ICSA-19-036-05	Third Party Advisory US Government Resource

Weakness Enumeration

CWE-ID	CWE Name	Source
NVD-CWE-Other	Other	NIST
CWE-598	Use of GET Request Method With Sensitive Query Strings	ICS-CERT

QUICK INFO

CVE Dictionary Entry:
CVE-2019-6531
NVD Published Date:
04/02/2019
NVD Last Modified:
06/22/2021
Source:
ICS-CERT