MyGate Web Applications Security Assessment Report

MyGate Web App

REPORT PUBLISH DATE **Monday Mar 18 2024 16:55 GMT+0000 (UTC)**

Table of Contents

1 Executive Summary

1.1 Strategic Recommendation

It is recommended to fix all critical, high and medium vulnerabilities before releasing the application to customer.

1.2 Scope of Work

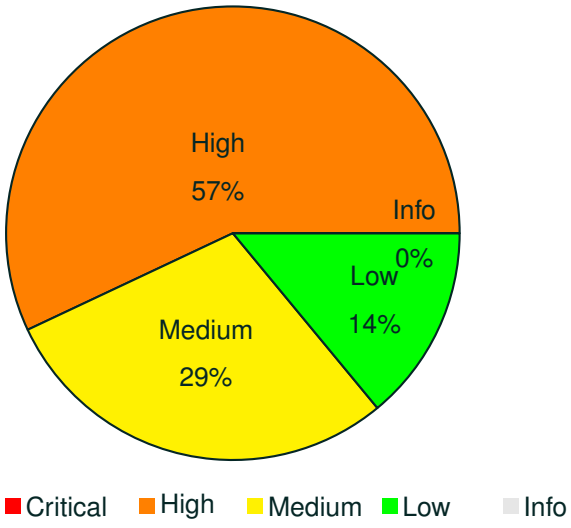
The scope of this penetration test was limited to the URL mentioned below:

Scope Details			
Sr. No.	**Application Name**	**Application URL**	**Scope**
1.	Callyzer	http://65.21.6.24/	Callyzer web Application Manually & using Burpsuite

1.3 Summary of Findings

- Graphical Summary

Vulnerability v/s Severity Pie Chart



Vulnerability Summary

Severity	Count
Critical	0
High	8
Medium	4
Low	2
Informational	0
Total	14

2 CVSS: Score Vulnerabilities

there are three types of scores that can be calculated: a base score, a temporal score and an environmental score. For purposes of reporting in this document, the CVSS base score will be provided. The base score assesses the following characteristics:

Characteristics	Description
Attack Vector	Assesses whether or an adversary can mount attack from a remote network, a local network or if an adversary must be logged on to the target of evaluation or physically connected.
Attack Complexity	Assesses the complexity of an attack dependent on how many of the attack variables are within the control of the adversary.
Privileges Required	Assesses the level of access that an attacker needs to mount a successful attack.
User Interaction	Assesses the extent to which actions of the victim are required for an attack to be successful.
Scope	Assess whether the impact of an attack is limited to the target of evaluation or if the attack has impact on other systems as well.
Confidentiality	Assesses the negative impact that an attack can have on the target of evaluation's confidentiality.
Integrity	Assesses the negative impact that an attack can have on the target of evaluation's integrity.
Availability	Assesses the negative impact that an attack can have on the target of evaluation's availability.

As indicated above, the assessment of these characteristics results in a severity score which ranges from 1-10. This score can be further broken down into the following rating levels:

Range	Rating	Description
9.0 - 10.0	**Critical**	These types of vulnerabilities should be reviewed immediately for impact to the business. This rating usually indicates that an exploit exists that could easily be use severely impact confidentiality, integrity and/or availability.
7.0 - 8.9	**High**	These types of vulnerabilities need to be assessed in the short term for impact to the business. A score in this range indicates that a vulnerability could be exploited with low to medium complexity and could have moderate or high impact on confidentiality, integrity and/or availability.
4.0 - 6.9	**Medium**	These vulnerabilities should also be evaluated for impact to the business, but the base score shows that these types of vulnerabilities may be only exploitable with increased effort or have little impact to confidentiality, integrity and/or availability.
0.1 - 3.9	**Low**	These vulnerabilities should also be evaluated, but from evaluating the base characteristics, the exploitation of these vulnerabilities is likely to result in little negative impact to confidentiality, integrity and/or availability.

The CVSS score provided in this report is meant to serve as a tool to assist with the prioritizing vulnerability resolution. This score, however, does not take into consideration the context of the business. For some business IT contexts some lower-scored vulnerabilities could have serious business impact. Hence, all of the reported vulnerabilities should be taken into consideration.

2.1 How to use this report

The vulnerabilities reported in this document provide a view of the target of evaluation's security posture at the time of testing. This timeframe, "at the time of testing", is important to highlight because the report cannot address future changes to the target of evaluation, changes in the systems that support the target of evaluation and emerging, publicly disclosed exploits that could have an impact on the target of evaluation.

The goal of this document is to provide input to help identify and prioritize the vulnerabilities that were detected at the time of testing and to provide some guidance as to how the vulnerabilities might be mitigated.

Characteristics	Description
Status	This field will contain either "Verified" or "Detected". If this value is "Verified", then the tester exploited this vulnerability during the penetration test. If it is "Detected", then evidence of the vulnerability was found, but it was not exploited during testing. There are many reasons why a tester may not be able to exploit a vulnerability during testing. Examples include threat of system instability after exploit, lack of time during testing and/or inability to find a vector by which a vulnerability could be exploited.
CVSSv3.1 Scoring	This provides the overall severity score for a vulnerability including the individual assessments for attack vector, attack complexity, privileges required, user interaction, scope, confidentiality, integrity and availability.
Vulnerability Description	This provides an overview of the identified vulnerability including how it could be useful to an adversary.
Proof of Concept	This provides a description of how the vulnerability was detected and/or a description of how it can be reproduced for testing purposes.
Affected Uri	This provides a list of the url that are relevant to the vulnerability.
Recommendation	This provides suggestions on how to mitigate the vulnerability.
References	This provides links to CVEs, CWEs and other known resources to learn more about the vulnerability and how to mitigate the vulnerability.

The report is broken up into three major sections: an executive summary, a technical detail report and an appendix. The executive summary will provide a high-level overview of the vulnerabilities detected during the penetration test.

The technical detail report will provide the details of the vulnerabilities identified during the penetration test. Each vulnerability will include the following descriptors.

The appendix will contain information about the testing environment and further details gathered during testing that do not fit within the first three chapters. This information is necessary to have a complete picture of the penetration test, but it is in the appendix to make accessing the testing results more userfriendly.

3 Findings Overview

The following table summarizes the list of findings discovered during the security assessment

Summary Table				
Sr. No.	**Vulnerability Name**	**OWASP Category**	**Severity**	**CVSS Score++**
1	**Application Does Not Implement HSTS Best Practices**	**A05-security Misconfiguration**	**Info**	2.0
2	**Inadequate, Inconsistent or Missing Cookie Attributes**	**A05-security Misconfiguration**	**Info**	8.0
3	**Missing Content Security Policy Header**	**A05-security Misconfiguration**	**Info**	4.2

4 Technical Reports

The following findings were made during the assessment.

Finding Name	Remediation Effort
Critical Severity Findings	
High Severity Findings	
Medium Severity Findings	
Reflected Cross Site Scripting (XSS)	Quick
Low Severity Findings	
Informational Findings	
Application Does Not Implement HSTS Best Practices	Planned
Inadequate, Inconsistent or Missing Cookie Attributes	Planned
Missing Content Security Policy Header	Planned

4.1 Application Does Not Implement HSTS Best Practices

Status: New

Severity: **Informational**

OWASP Category: Security Misconfiguration

CVSS Score: 2.0

Affected Hosts/URLs:

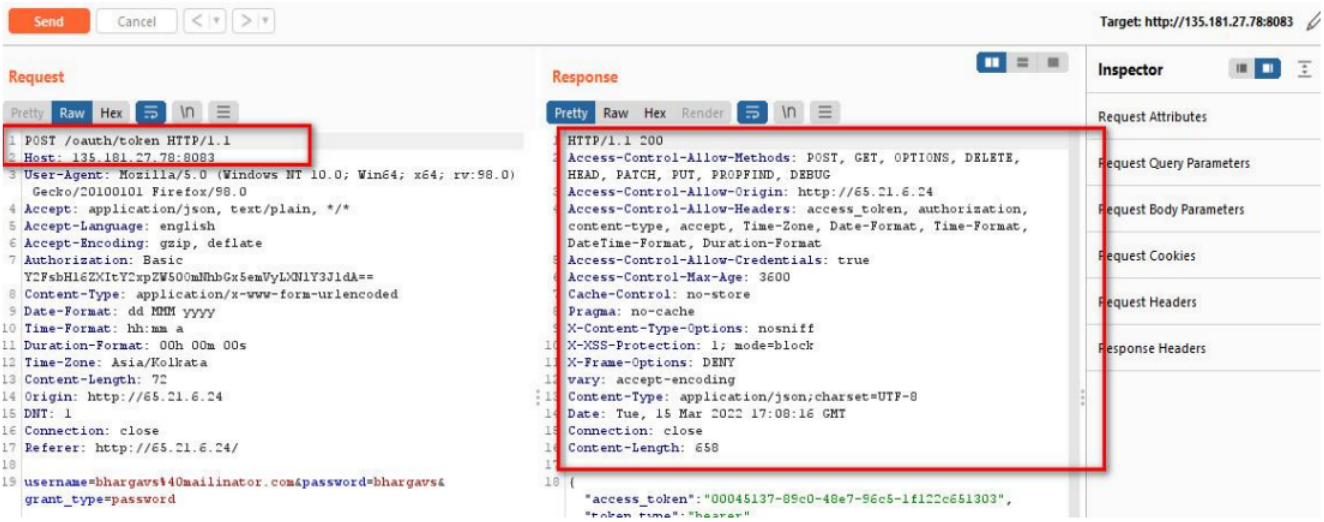
- http://65.21.6.24/dashBoard

Summary:

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

proof of concept:



Remediation:

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

Reference:

https://www.techtarget.com/searchsecurity/definition/email-spoofing

4.2 Inadequate, Inconsistent or Missing Cookie Attributes

Status: New

Severity: **Informational**

OWASP Category: Security Misconfiguration

CVSS Score: 8.0

Affected Hosts/URLs:

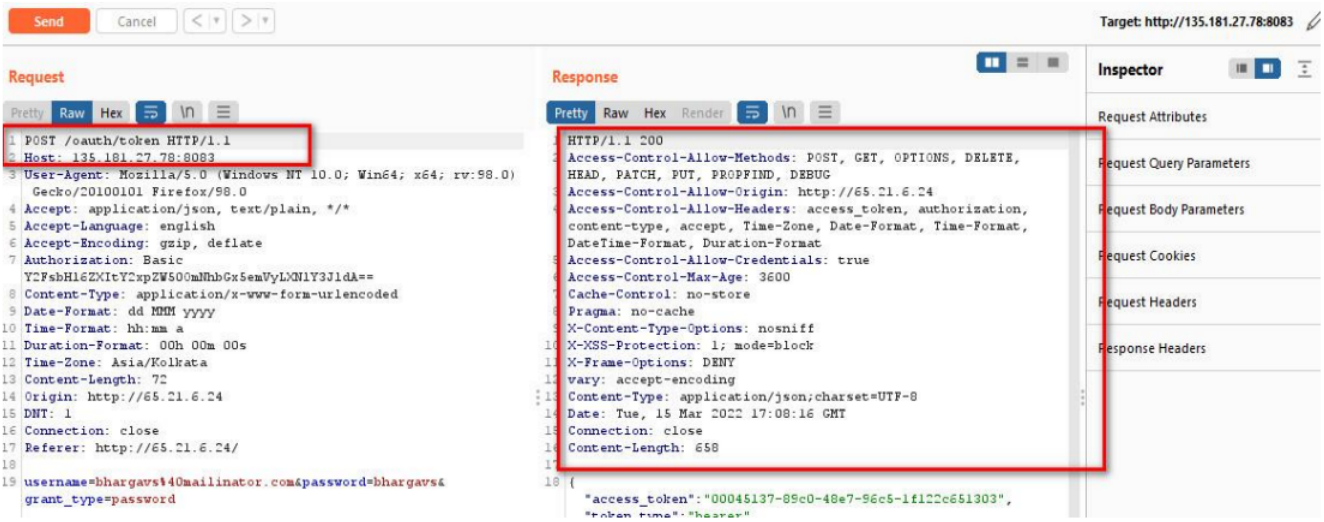
- http://65.21.6.24/dashBoard

Summary:

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

proof of concept:



Remediation:

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

Reference:

https://hstspreload.org

4.3 Missing Content Security Policy Header

Status: New

Severity: **Informational**

OWASP Category: Security Misconfiguration

CVSS Score: 4.2

Affected Hosts/URLs:

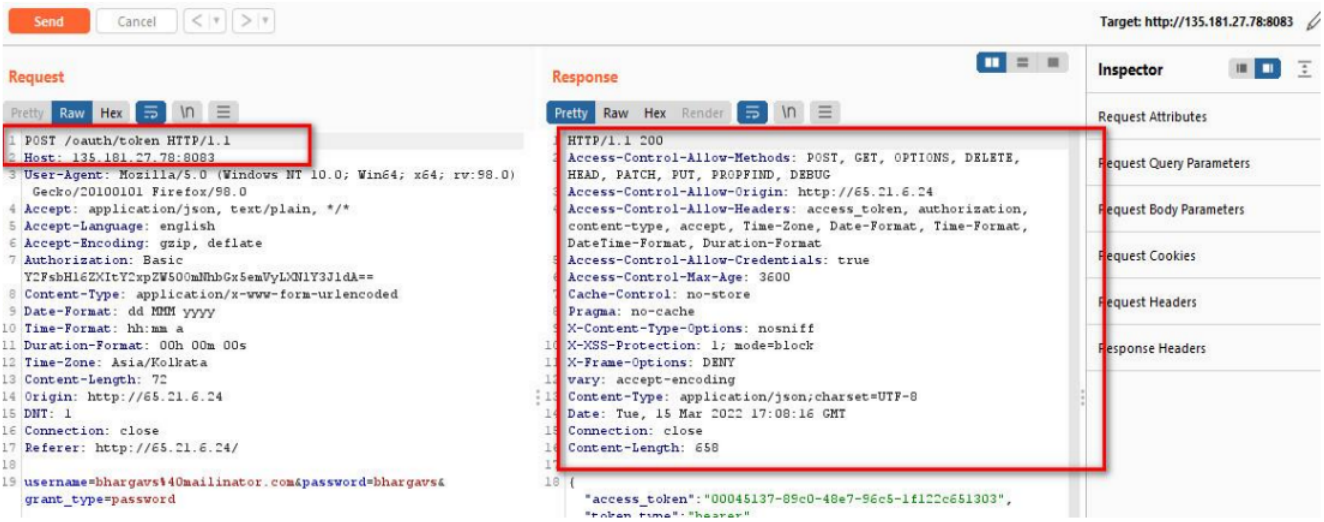
- https://65.21.6.24/dashBoard

Summary:

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks HTTP Strict Transport Security (HSTS) instructs a web browser to only connect to a web site using HTTPS. It was detected that your web application's HTTP StrictTransport Security (HSTS) implementation is not as strict as is typically advisable. The browser will expire the HSTS header after the number of seconds configured in the max-age attribute. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

proof of concept:



Remediation:

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year). The strongest protection is to ensure that all requested resources use only TLS with a well-formed HSTS header. It is advisable to assign the max-age directive's value to be greater than 10368000 seconds (120days) and ideally to 31536000 (one year).

Reference:

http://65.21.6.24/dashBoard

5 Annexures

5.1 OWASP TOP 10:2021

OWASP TOP 10:2021	
Name	**Description**
A01:2021-Broken Access Control	Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.
A02:2021 - Cryptographic Failures	Previously known as Sensitive Data Exposure, Cryptographic Failures involve protecting data in transit and at rest. This includes passwords, credit card numbers, health records, personal information, and business secrets that require extra protection, especially if that data falls under privacy.
A03:2021-Injection	Injection flaws, such as SQL, OS, XXE, XSS and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A04:2021-Insecure Design (Currently out of scope)	Insecure design is a broad category representing different weaknesses, expressed as "missing or ineffective control design." Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods.
A05:2021-Security Misconfiguration	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
A06:2021-Vulnerable and Outdated Components	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defences and enable a range of possible attacks and impacts.

A07:2021- Identification and Authentication Failures	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently).
A08:2021 - Software and Data Integrity Failures (Currently out of scope)	Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. This new category is making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity.
A09:2021-Security Logging and Monitoring Failures	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.
A10:2021-ServerSide Request Forgery(SSRF)	SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

5.2 Tools Used

Tools:	
Name	**Description**
Burp suite	**Burp Suite is an integrated platform for attacking web applications.** http://portswigger.net/suite
Nmap	**Nmap is a network mapper tool to scan for SSL related vulnerabilities** https://nmap.org