



Secure Software Development

SE4030

[2022/JUL]

Assignment 2

Submitted to

Sri Lanka Institute of Information Technology

Development Group Assignment Report

2022_REG_25

17.10.2022

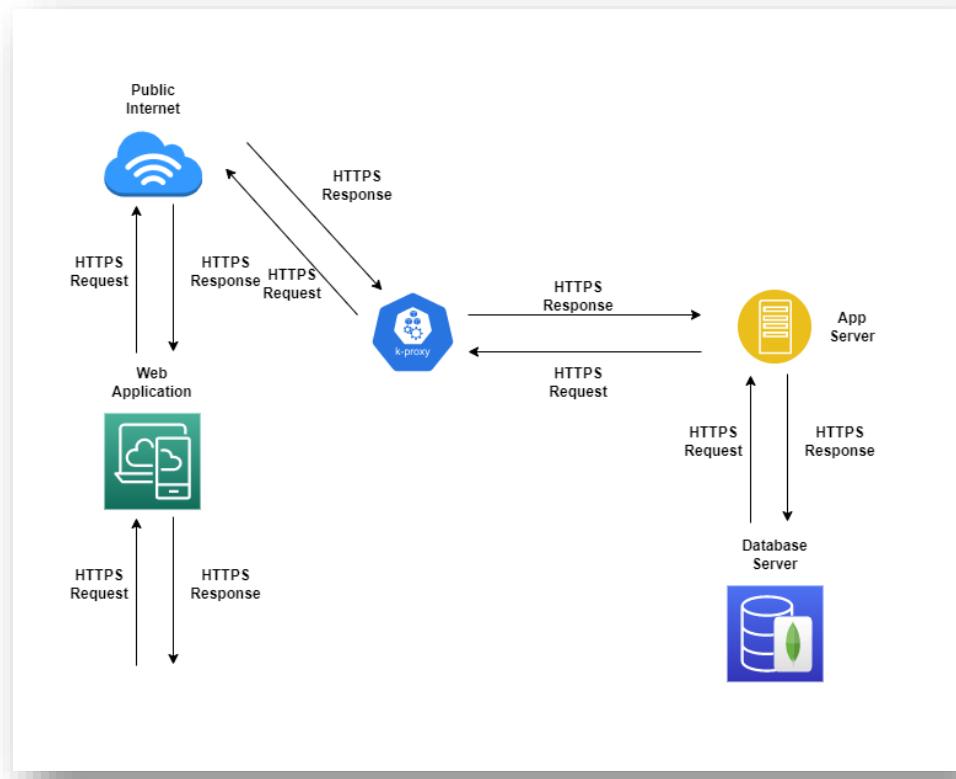
Name	Registration Number
Soujanyaah. K	IT19123400
Sivarajah. K	IT19119786
Viknatharshan. N	IT19107110
Sivashankar. S	IT19047102

Declaration

I certify that this report does not incorporate without acknowledgement, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief it does not contain any material previously published or written by another person, except where due reference is made in text.

Source Code Git Repository Link - https://github.com/soujanyaah/Group2022_Reg_25

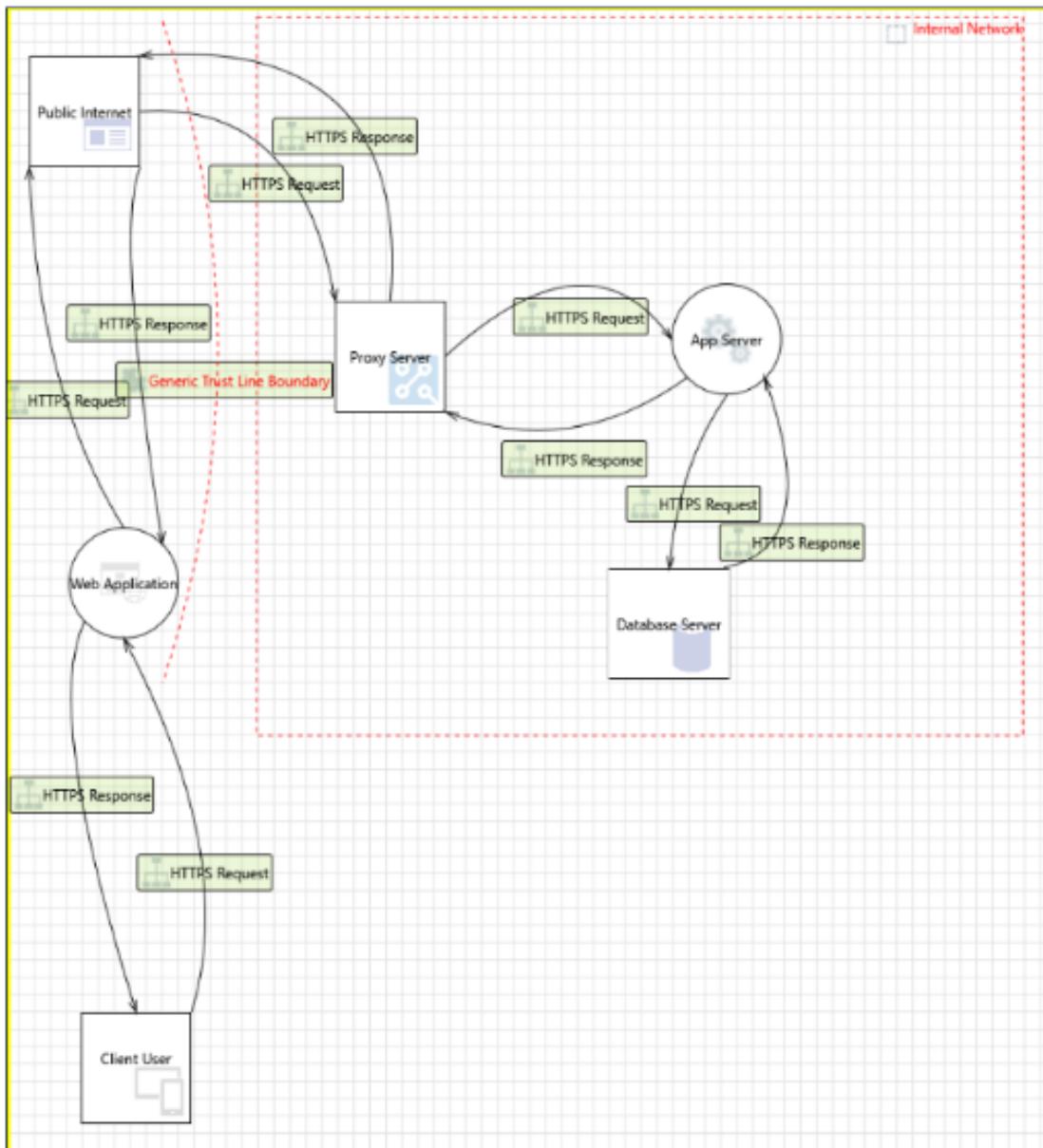
Software System Diagram



Thread Modeling

- We used **Microsoft Thread Modeling Tool** to design the main process of the software developing and identify the threats that the software faces and the mitigations that can be used to secure it.

Diagram: System Diagram



System Diagram Summary

Threat Modeling Report

Created on 16/11/2022 17:46:43

Threat Model Name: ABC Company Threat Model

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	35
Not Applicable	5
Needs Investigation	1
Mitigation Implemented	25
Total	66
Total Migrated	0

2. An adversary can deny actions performed on Database Server due to a lack of auditing [State: Not Applicable] [Priority: Medium]

Category: Regulation
Description: An adversary can deny actions performed on Cloud Storage due to a lack of auditing.
Justification: <no mitigation provided>
Possible: Enable auditing on Azure SQL Database Instances to track and log database events. After configuring and customizing the audited events, enable threat detection to receive alerts on anomalous database activities indicating potential security threats. Refer: https://aka.ms/tmo-th147
Mitigation(s): https://aka.ms/tmo-th142
SDL Phase: Design

3. A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments [State: Mitigation Implemented] [Priority: High]

Category: Elevation of Privileges
Description: A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments.
Justification: <no mitigation provided>
Possible: Enable Transparent Data Encryption (TDE) on Azure SQL Database instances to have data encrypted at rest. Refer: https://aka.ms/tmo-th145b. Use the Always Encrypted feature to allow client applications to encrypt sensitive data before it is sent to the Azure SQL Database.
Mitigation(s): https://aka.ms/tmo-th145
SDL Phase: Implementation

4. An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data.
Justification: <no mitigation provided>
Possible: Enable Transparent Data Encryption (TDE) on Azure SQL Database instances to have data encrypted at rest. Refer: https://aka.ms/tmo-th145a. Use the Always Encrypted feature to allow client applications to encrypt sensitive data before it is sent to the Azure SQL Database.
Mitigation(s): https://aka.ms/tmo-th145
SDL Phase: Implementation

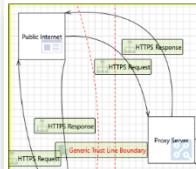
5. An adversary can read confidential data due to weak connection string configuration [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: An adversary can read confidential data due to weak connection string configuration.
Justification: <no mitigation provided>
Possible: Clients connecting to an Azure SQL Database instance using a connection string should ensure encrypt=true and trustedconnection=false are set. This configuration ensures that connections are encrypted only if there is a verifiable server certificate (otherwise the connection attempt fails). This helps protect against Man-in-The-Middle attacks. Refer: https://aka.ms/tmo-th144

6. Nonstandard threat to describe user specific conditions [State: Not Started] [Priority: High]

Category: User-defined
Description:
Justification: <no mitigation provided>
Possible Mitigation(s):
SDL Phase: Design

Interaction: HTTPS Request



7. An adversary may execute unknown code on Proxy Server [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may launch malicious code into Proxy Server and execute it.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that unknown code cannot execute on devices. Refer: https://aka.ms/tmtconfigmgmt#unknown-exec

SDL Phase: Design

Interaction: HTTPS Request



8. An adversary may gain unauthorized access to privileged features on Proxy Server [State: Mitigation Implemented] [Priority: High]

Category: Elevation of Privileges

Description: An adversary may get access to admin interface or privileged services like WiFi, SSH, File shares, FTP etc., on a device.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that all admin interfaces are secured with strong credentials. Refer: https://aka.ms/tmtconfigmgmt#admin-strong

SDL Phase: Implementation

10. An adversary may exploit known vulnerabilities in unpatched devices [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date. Refer: https://aka.ms/tmtconfigmgmt#cloud-firmware

SDL Phase: Design

11. An adversary may tamper Proxy Server and extract cryptographic key material from it [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may partially or wholly replace the software running on App Server potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.

Justification: <no mitigation provided>

Possible Mitigation(s): Store Cryptographic Keys securely on IoT Device. Refer: http://aka.ms/tmtcryptoto#keys-iot

Mitigation(s): Design

12. An adversary may tamper the OS of a device and launch offline attacks [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary may launch offline attacks made by disabling or circumventing the installed operating system, or made by physically separating the storage media from the device in order to attack the data separately.

Justification: <no mitigation provided>

Possible Mitigation(s): Encryt OS and additional partitions of IoT Device with BitLocker. Refer: https://aka.ms/tmtconfigmgmt#partition-iot

SDL Phase: Design

Interaction: HTTPS Request



13. An adversary can gain unauthorized access to database due to loose authorization rules	[State: Not Started] [Priority: High]
Category:	Elevation of Privileges
Description:	Database access should be configured with roles and privilege based on least privilege and need to know principle.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that least-privileged accounts are used to connect to Database server. Refer: https://aka.ms/tmtauthz#privileged-server. Implement Row Level Security (RLS) to prevent tenants from accessing each other's data. Refer: https://aka.ms/tmtauthz#rls-tenants. Sysadmin role should only have valid necessary users. Refer: https://aka.ms/tmtauthz#sysadmin-user.
SDL Phase:	Implementation
14. An adversary can gain unauthorized access to Azure SQL DB instances due to weak network security configuration.	[State: Not Applicable] [Priority: High]
Category:	Elevation of Privileges
Description:	An adversary can gain unauthorized access to Azure SQL DB instances due to weak network security configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Restrict access to Azure SQL Database Instances by configuring server-level and database-level firewall rules to permit connections from selected networks (e.g. a virtual network or a custom set of IP addresses) where possible. Refer: https://aka.ms/tmo-0143.
SDL Phase:	Implementation
15. An adversary can read confidential data due to weak connection string configuration	[State: Mitigation Implemented] [Priority: High]
Category:	Information Disclosure
Description:	An adversary can read confidential data due to weak connection string configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Client connection to an Azure SQL Database instance using a connection string should ensure encrypt=true and trustServerCertificate=false are set. This configuration ensures that connections are encrypted only if there is a verifiable server certificate (otherwise the connection attempt fails). This helps protect against Man-in-the-Middle attacks. Refer: https://aka.ms/tmo-0144.
Mitigation(s):	Implementation
16. An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data	[State: Mitigation Implemented] [Priority: High]
Category:	Information Disclosure
Description:	An adversary having access to the storage container (e.g. physical access to the storage media) may be able to read sensitive data.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Enable Transparent Data Encryption (TDE) on Azure SQL Database instances to have data encrypted at rest. Refer: https://aka.ms/tmo-0145a. Use the Always Encrypted feature to allow client applications to encrypt sensitive data before it is sent to the Azure SQL Database. Refer: https://aka.ms/tmo-0145b.
SDL Phase:	Implementation
17. A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments	[State: Mitigation Implemented] [Priority: High]
Category:	Elevation of Privileges
Description:	A compromised identity may permit more privileges than intended to an adversary due to weak permission and role assignments.
Justification:	<no mitigation provided>
Possible Mitigation(s):	It is recommended to review permission and role assignments to ensure the users are granted the least privileges necessary. Refer: https://aka.ms/tmt-0146.
SDL Phase:	Implementation
18. An adversary can deny actions performed on Database Server due to a lack of auditing	[State: Not Applicable] [Priority: Medium]
Category:	Repudiation
Description:	An adversary can deny actions performed on Database Server due to a lack of auditing.
Justification:	<no mitigation provided>
21. An adversary can gain unauthorized access to database due to lack of network access protection	[State: Not Started] [Priority: High]
Category:	Elevation of Privileges
Description:	If there is no restriction at network or host firewall level, to access the database then anyone can attempt to connect to the database from an unauthorized location.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Configure Windows Firewall for Database Engine Access. Refer: https://aka.ms/tmoconfiggntfirewall-db.
SDL Phase:	Implementation
20. An adversary may abuse weak Database Server configuration	[State: Not Applicable] [Priority: High]
Category:	Elevation of Privileges
Description:	An adversary may abuse weak Database Server configuration.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Enable SQL Vulnerability Assessment to gain visibility into the security posture of your Azure SQL Database instances. Acting on the assessment results help reduce attack surface and enhance your database security. Refer: https://aka.ms/tmo-0149.
SDL Phase:	Implementation
21. An adversary can gain access to sensitive PII or HBI data in database	[State: Not Started] [Priority: High]
Category:	Information Disclosure
Description:	Additional controls like Transparent Data Encryption, Column Level Encryption, EKM etc. provide additional protection mechanism to high value PII or HBI data.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Use encryption algorithms to encrypt data in the database. Refer: https://aka.ms/tmcrypt#protecting-db. Ensure that sensitive data in database columns is encrypted. Refer: https://aka.ms/tmcrypt#db-encryption. Ensure that database backups are encrypted. Refer: https://aka.ms/tmcrypt#backups. Use SQL server EKM to protect encryption keys. Refer: https://aka.ms/tmcrypt#store#keys. Use AlwaysEncrypted feature if encryption keys should not be revealed to Database engine. Refer: https://aka.ms/tmcrypt#key-engine.
SDL Phase:	Implementation
22. An adversary can gain access to sensitive data by performing SQL injection	[State: Not Started] [Priority: High]
Category:	Information Disclosure
Description:	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or its metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that login auditing is enabled on SQL Server. Refer: https://aka.ms/tmauditlog#identify-sensitive-entities. Ensure that least-privileged accounts are used to connect to Database server. Refer: https://aka.ms/tmtauthz#privileged-server. Enable Threat detection on Azure SQL Database. Refer: https://aka.ms/tmauditlog#threat-detection. Do not use dynamic queries in stored procedures. Refer: https://aka.ms/tmcrypt#stored-proc.
SDL Phase:	Implementation
23. An adversary can deny actions on database due to lack of auditing	[State: Not Started] [Priority: Medium]
Category:	Repudiation
Description:	Precise logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.
Justification:	<no mitigation provided>
Possible Mitigation(s):	Ensure that login auditing is enabled on SQL Server. Refer: https://aka.ms/tmauditlog#identify-sensitive-entities.
SDL Phase:	Implementation
24. An adversary can tamper critical database securities and deny the action	[State: Not Started] [Priority: High]
Category:	Damaging

24. An adversary can tamper critical database securities and deny the action [State: Not Started] [Priority: High]
<p>Category: Tampering</p> <p>Description: An adversary can tamper critical database securities and deny the action</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Add digital signature to critical database securities. Refer: https://aka.ms/tmtcrypto#securables-db</p> <p>SDL Phase: Design</p>
25. An adversary may leverage the lack of monitoring systems and trigger anomalous traffic to database [State: Not Started] [Priority: High]
<p>Category: Tampering</p> <p>Description: An adversary may leverage the lack of intrusion detection and prevention of anomalous database activities and trigger anomalous traffic to database</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Enable Threat detection on Azure SQL database. Refer: https://aka.ms/tmauditlog/threat-detection</p> <p>SDL Phase: Design</p>
Interaction: HTTPS Request
<pre> graph TD CU[Client User] -- "HTTPS Req" --> WA[Web Application] WA -- "HTTP Response" --> HR[HTTP Res] HR -- "HTTPS Res" --> CU </pre>
26. An adversary may jailbreak into a mobile device and gain elevated privileges [State: Mitigation Implemented] [Priority: High]
<p>Category: Elevation of Privileges</p> <p>Description: An adversary may jailbreak into a mobile device and gain elevated privileges</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Implement implicit jailbreak or rooting detection. Refer: https://aka.ms/tmauthz#rooting-detection</p>
27. An adversary can reverse weakly encrypted or hashed content [State: Mitigation Implemented] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary can reverse weakly encrypted or hashed content</p> <p>Justification: <no mitigation provided></p> <p>Mitigation(s): Do not expose security details in error messages. Refer: https://aka.ms/tmcmgnt#messages. Implement Default error handling page. Refer: https://aka.ms/tmcmgnt#default. Set Deployment Method to Retail in IS. Use approved block cipher modes and initialization vectors for symmetric ciphers. Refer: https://aka.ms/tmtcrypto#vector-ciphers. Use approved asymmetric algorithms, key lengths, and padding. Refer: https://aka.ms/tmcmgnt#padding. Use approved random number generators. Refer: https://aka.ms/tmcmgnt#rngs. Do not use symmetric stream ciphers. Refer: https://aka.ms/tmtcrypto#stream-ciphers. Use only approved cryptographic hash functions. Refer: https://aka.ms/tmcmgnt#hash-functions. Verify X.509 certificates used to authenticate SSL, TLS and DTLS connections. Refer: https://aka.ms/tmcrypto#x509-ids.</p> <p>SDL Phase: Implementation</p>
28. An adversary may gain access to sensitive data from log files [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary may gain access to sensitive data from log files</p> <p>Justification: <no mitigation provided></p> <p>Mitigation(s): Ensure that the application does not log sensitive user data. Refer: https://aka.ms/tmauditlog#log-sensitive-data. Ensure that Audit and Log Files have Restricted Access. Refer: https://aka.ms/tmauditlog#log-restricted-access.</p> <p>SDL Phase: Implementation</p>
29. An adversary can gain access to sensitive data by sniffing traffic from Mobile client [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary can gain access to sensitive data by sniffing traffic from Mobile client</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Implement Certificate Pinning. Refer: https://aka.ms/tmcmsecc#certs-pinning</p> <p>SDL Phase: Implementation</p>
30. An adversary can gain sensitive data from mobile device [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: If application saves sensitive PI or HBI data on phone SD card or local storage, then it may get stolen.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Encrypt sensitive or PI data written to phones local storage. Refer: https://aka.ms/tmdata#pii-phones</p> <p>SDL Phase: Implementation</p>
31. An adversary can gain access to sensitive information through error messages [State: Mitigation Implemented] [Priority: High]

31. An adversary can gain access to sensitive information through error messages [State: Mitigation Implemented] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary can gain access to sensitive data such as the following: through verbose error messages > Server names > Connection strings > Usernames > Passwords > SQL procedures > Details of dynamic SQL failures > Stack trace and lines of code > Variables stored in memory > Drive and folder locations > Application install points > Host configuration settings > Other internal application details</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Do not expose security details in error messages. Refer: https://aka.ms/tmconfig#messages. Implement Default error handling page. Refer: https://aka.ms/tmconfig#default. Set Deployment Method to Retail in IIS.</p> <p>SDL Phase: Implementation</p>
32. Attacker can deny the malicious act and remove the attack foot prints leading to reputation damage [State: Not Started] [Priority: Medium]
<p>Category: Reputation</p> <p>Description: Proper logging of all security events and user actions builds traceability in a system and denies any possible reputational issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure proper logging and logging is enforced on the application. Refer: https://aka.ms/tmauditlog#auditing. Ensure that log rotation and separation are in place. Refer: https://aka.ms/tmauditlog#log-rotation. Ensure that Audit Log Files have Restricted Access. Refer: https://aka.ms/tmauditlog#log-restricted-access. Ensure that User Management Events are Logged. Refer: https://aka.ms/tmauditlog#user-management.</p> <p>SDL Phase: Implementation</p>
33. An adversary can spoof the target web application due to insecure TLS certificate configuration [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: Ensure that TLS certificate parameters are configured with correct values</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: https://aka.ms/tmcommsec#s09-ssls</p> <p>SDL Phase: Implementation</p>
34. An adversary can steal sensitive data like user credentials [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: Hackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if Credentials are stored and sent in clear text. Weak input validation coupled with dynamic SQL queries, Password retrieval mechanism are poor</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Refer: https://aka.ms/tmtdata#autocomplete-input. Perform input validation and filtering on all string type Model properties. Refer: https://aka.ms/tmtdata#jytemodel. Validate all redirects within the application are closed or done safely. Refer: https://aka.ms/tmtdataval#redirecs-safe. Enable step up or adaptive authentication. Refer: https://aka.ms/tmauthn#step-up-adaptive-auth. Implement forget password functionalities securely. Refer: https://aka.ms/tmauthn#forgo-pwrd-fnm. Ensure that password and account policy are implemented. Refer: https://aka.ms/tmauthn#pwrd-account-policy. Implement input validation on all string type parameters accepted by Controller methods. Refer: https://aka.ms/tmtdataval#string-methods.</p> <p>SDL Phase: Implementation</p>
35. An adversary can create a fake website and launch phishing attacks [State: Not Started] [Priority: High]
<p>Category: Spoofing</p> <p>Description: Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: https://aka.ms/tmcommsec#s09-ssls. Ensure that authenticated ASP.NET pages incorporate UI Redressing or clickjacking defences. Refer: https://aka.ms/tmconfig#ui-defenses. Validate all redirects within the application are closed or done safely. Refer: https://aka.ms/tmtdataval#redirecs-safe.</p> <p>SDL Phase: Implementation</p>
36. An adversary may spoof Client User and gain access to Web Application [State: Not Started] [Priority: High]
<p>Category: Spoofing</p> <p>Description: If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Consider using a standard authentication mechanism to authenticate to Web Application. Refer: https://aka.ms/tmauthn#standard-authn-web-app.</p> <p>SDL Phase: Design</p>
37. An adversary can reverse engineer and tamper binaries [State: Needs Investigation] [Priority: High]
<p>Category: Tampering</p> <p>Description: An adversary can use various tools, reverse engineer binaries and abuse them by tampering</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Obfuscate generated binaries before distributing to end users. Refer: https://aka.ms/tmtdata#binaries-end</p> <p>SDL Phase: Design</p>
38. An adversary can gain access to sensitive data by performing SQL injection through Web App [State: Not Started] [Priority: High]
<p>Category: Tampering</p> <p>Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that type-safe parameters are used in Web Application for data access. Refer: https://aka.ms/tmtdataval#ypesafe.</p> <p>SDL Phase: Implementation</p>
39. An adversary can gain access to sensitive data stored in Web App's config files [State: Mitigation Implemented] [Priority: High]
<p>Category: Tampering</p> <p>Description: An adversary can gain access to the config files, and if sensitive data is stored in it, it would be compromised.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Encrypt sections of Web App's configuration files that contain sensitive data. Refer: https://aka.ms/tmtdata#encrypt-data.</p> <p>SDL Phase: Implementation</p>

Interaction HTTPS Response



Interaction: HTTPS Response



40. An adversary can gain access to sensitive data stored in Web App's config files [State: Not Started] [Priority: High]

Category: Tampering

Description: An adversary can gain access to the config files, and if sensitive data is stored in it, it would be compromised.

Justification: <no mitigation provided>

Possible Mitigation(s): Encrypt sections of Web Apps configuration files that contain sensitive data. Refer: https://aka.ms/tmtdata#encript-data

SDL Phase: Implementation

41. An adversary can gain access to sensitive data by performing SQL injection through Web App [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.

Justification: <no mitigation provided>

Possible Mitigation(s): Ensure that type-safe parameters are used in Web Application for data access. Refer: https://aka.ms/tmtpub#tysafe

SDL Phase: Implementation

42. An attacker steals messages off the network and replays them in order to steal a user's session [State: Not Started] [Priority: High]

Category: Tampering

Description: An attacker steals messages off the network and replays them in order to steal a user's session

Justification: <no mitigation provided>

42. An attacker steals messages off the network and replays them in order to steal a user's session [State: Not Started] [Priority: High]

Category: Tampering

Description: An attacker steals messages off the network and replays them in order to steal a user's session

Justification: <no mitigation provided>

SDL Phase: Implementation

43. An adversary can deface the target web application by injecting malicious code or uploading dangerous files [State: Not Started] [Priority: High]

Category: Tampering

Description: Website defacement is an attack on a website where the attacker changes the visual appearance of the site or a webpage.

Justification: <no mitigation provided>

Possible Mitigation(s): Implement Content Security Policy (CSP), and disable inline JavaScript. Refer: https://aka.ms/tmtnconfigimg#nop-javascript. Enable browser's XSS filter. Refer: https://aka.ms/tmtnconfigimg#xss-filter. Access third party JavaScripts from trusted sources only. Refer: https://aka.ms/tmtnconfigimg#trusted. Enable ValidateRequest attribute on ASP.NET Pages. Refer: https://aka.ms/tmtnconfigimg#valdata-aspx. Ensure that each page that could contain user-controllable content opts out of automatic MIME sniffing. Refer: https://aka.ms/tmtnpub#auto-sniffing. Use locally-hosted latest versions of JavaScript libraries. Refer: https://aka.ms/tmtnconfigimg#ecmaljs. Implement Content Security Policy (CSP) on the system. Refer: https://aka.ms/tmtnconfigimg#csp. Disable automatic MIME sniffing. Refer: https://aka.ms/tmtnconfigimg#mimesniff. Perform strict validation of file extensions. Refer: https://aka.ms/tmtnconfigimg#fileext. Implement Input Validation. Refer: https://aka.ms/tmtnpub#input-validation. Implement Input Validation on all string type parameters accepted by Controller methods. Refer: https://aka.ms/tmtnpub#string-method. Avoid using HTML Raw in Razor views. Refer: https://aka.ms/tmtnpub#razor. Sanitization should be applied on form fields that accept all characters e.g. rich text editor. Refer: https://aka.ms/tmtnpub#richedit. Do not assign DOM elements to binds that do not have Inbuilt Encoding. Refer: https://aka.ms/tmtnpub#inbuilt-encoding.

SDL Phase: Implementation

44. An adversary may spoof Public Internet and gain access to Web Application [State: Not Started] [Priority: High]

Category: Spoofing

Description: If proper authentication is not in place, an adversary can spoof a source process or external entity and gain unauthorized access to the Web Application

Justification: <no mitigation provided>

Possible Mitigation(s): Consider using a standard authentication mechanism to authenticate to Web Application. Refer: https://aka.ms/tmtnauth#standard-authn-web-app

SDL Phase: Design

45. An adversary can create a fake website and launch phishing attacks [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Phishing is attempted to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a Web Server which is a trustworthy entity in electronic communication.

Justification: <no mitigation provided>

Possible Mitigation(s): Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: https://aka.ms/tmtncommack#sd5-sets. Ensure that authenticated ASP.NET pages incorporate UI Redressing or Clickjacking defenses. Refer: https://aka.ms/tmtnconfigimg#ui-defenses. Validate all redets within the application are closed or done safely. Refer: https://aka.ms/tmtnpub#redetsafe.

SDL Phase: Implementation

46. Attackers can steal user session cookies due to insecure cookie attributes [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: The session cookie is the identifier by which the server knows the identity of current user for each incoming request, if the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.

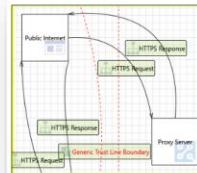
Justification: <no mitigation provided>

Possible Mitigation(s): Applications available over HTTPS must use secure cookies. Refer: https://aka.ms/tmtnmgmt#https-secure-cookies. All http based application should specify http only for cookie definition. Refer: https://aka.ms/tmtnmgmt#cookie-definition.

46. Attackers can steal user session cookies due to insecure cookie attributes [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Applications available over HTTPS must use secure cookies. Refer: https://aka.ms/tmsmgmt#https-secure-cookies. All http based application should specify http only for cookie definition. Refer: https://aka.ms/tmsmgmt#cookie-definition</p> <p>SDL Phase: Implementation</p>
47. An adversary can steal sensitive data like user credentials [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: Attackers can exploit weaknesses in system to steal user credentials. Downstream and upstream components are often accessed by using credentials stored in configuration stores. Attackers may steal the upstream or downstream component credentials. Attackers may steal credentials if Credentials are stored and sent in clear text. Weak input validation coupled with dynamic SQL queries. Password rehashes.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs. Refer: https://aka.ms/tmtdata#autocomplete-input. Perform input validation and filtering on all string type Model properties. Refer: https://aka.ms/tmtpuval#redirections. Enable step up or adaptive authentication. Refer: https://aka.ms/tmauthn#step-up-adaptive-auth. Implement forgot password functionalities securely. Refer: https://aka.ms/tmauthn#forgot-pwrd-fin. Ensure that password and account policy are implemented. Refer: https://aka.ms/tmauthn#pwrd-account-policy. Implement input validation on all string type parameters accepted by Controller methods. Refer: https://aka.ms/tmtpuval#string-methods</p> <p>SDL Phase: Implementation</p>
48. An adversary can spoof the target web application due to insecure TLS certificate configuration [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: Ensure that TLS certificate parameters are configured with correct values</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Verify x.509 certificates used to authenticate SSL, TLS, and DTLS connections. Refer: https://aka.ms/tmcommsec#ssl-tls</p> <p>SDL Phase: Implementation</p>
49. An adversary can get access to a user's session due to insecure coding practices [State: Not Started] [Priority: High]
<p>Category: Spoofing</p> <p>Description: The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure Validation of attributes on ASP.NET Pages. Refer: https://aka.ms/tmconfig#aspnet-validation. Encodes untrusted web output prior to rendering. Refer: https://aka.ms/tmtpuval#rendering. Ensure Validation of attributes in Razor views. Refer: https://aka.ms/tmtpuval#razor. Sanitization should be applied on form fields that accept all characters e.g. rich text editor. Refer: https://aka.ms/tmtpuval#richedit. Do not assign DOM elements to sinks that do not have Inout encoding. Refer: https://aka.ms/tmtpuval#inout-encoding</p> <p>SDL Phase: Implementation</p>
50. An adversary can get access to a user's session due to improper logout and timeout [State: Mitigation Implemented] [Priority: High]
<p>Category: Spoofing</p> <p>Description: The session cookies is the identifier by which the server knows the identity of current user for each incoming request. If the attacker is able to steal the user token he would be able to access all user data and perform all actions on behalf of user.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Set up session for activity lifetime. Refer: https://aka.ms/tmsmgmt#activity-lifetime. Implement proper logout from the application. Refer: https://aka.ms/tmsmgmt#proper-app-logout</p> <p>SDL Phase: Implementation</p>
51. Attacker can deny the malicious act and remove the attack foot prints leading to repudiation issues [State: Not Started] [Priority: Medium]
<p>Category: Repudiation</p> <p>Description: Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that auditing and logging is enforced on the application. Refer: https://aka.ms/tmauditlog#auditing. Ensure that log rotation and separation are in place. Refer: https://aka.ms/tmauditlog#log-rotation. Ensure that Audit and Log Files have Restricted Access. Refer: https://aka.ms/tmauditlog#log-restricted-access. Ensure that User Management Events are Logged. Refer: https://aka.ms/tmauditlog#user-management</p> <p>SDL Phase: Implementation</p>
52. An adversary may gain access to sensitive data from uncleared browser cache [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary may gain access to sensitive data from uncleared browser cache</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that sensitive content is not cached on the browser. Refer: https://aka.ms/tmtdata#cache-browser</p> <p>SDL Phase: Implementation</p>
53. An adversary can gain access to sensitive information through error messages [State: Mitigation Implemented] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary can gain access to sensitive data such as the following, through verbose error messages - Server names - Connection strings - Usernames - Passwords - SQL procedures - Details of dynamic SQL failures - Stack trace and lines of code - Variables stored in memory - Drive and folder locations - Application initial points - Host configuration settings - Other internal application details</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Do not expose security details in error messages. Refer: https://aka.ms/tmconfig#messages. Implement Default error handling page. Refer: https://aka.ms/tmconfig#default. Set Deployment Method to Retail in IIS. Refer: https://aka.ms/tmconfig#deploy-manage. Exceptions should fail safely. Refer: https://aka.ms/tmconfig#fail. ASP.NET applications must disable tracing and debugging prior to deployment. Refer: https://aka.ms/tmconfig#trace-deploy. Implement controls to prevent username enumeration. Refer: https://aka.ms/tmauthn#controls-username-enum</p> <p>SDL Phase: Implementation</p>
54. An adversary can gain access to sensitive data by sniffing traffic to Web Application [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary may conduct man in the middle attack and downgrade TLS connection to clear text protocol, or forcing browser communication to pass through a proxy server that he controls. This may happen because the application may use mixed content or HTTP Strict Transport Security policy is not ensured.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Applications available over HTTPS must use secure cookies. Refer: https://aka.ms/tmsmgmt#https-secure-cookies. Enable HTTP Strict Transport Security (HSTS). Refer: https://aka.ms/tmcommsec#http-hsts</p> <p>SDL Phase: Implementation</p>
55. An adversary can gain access to certain pages or the site as a whole. [State: Not Started] [Priority: Medium]
<p>Category: Information Disclosure</p> <p>Description: Robots.txt is often found in your sites root directory and exists to regulate the bots that crawl your site. This is where you can grant or deny permission to all or some specific search engine robots to access certain pages or your site as a whole. The standard for this file was developed in 1994 and is known as the Robots Exclusion Standard or Robots Exclusion Protocol. Detailed info about the robots.txt protocol can be found at robots.txt.org</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that administrative interfaces are appropriately locked down. Refer: https://aka.ms/tmauthn#admin-interface-lockdown</p> <p>SDL Phase: Implementation</p>

56. An adversary may gain access to unmasked sensitive data such as credit card numbers [State: Not Started] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary may gain access to unmasked sensitive data such as credit card numbers</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that sensitive data displayed on the user screen is masked. Refer: https://aka.ms/tmtdata#data-mask</p> <p>SDL Phase: Implementation</p>
57. An adversary may gain access to sensitive data from log files [State: Mitigation Implemented] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary may gain access to sensitive data from log files</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that the application does not log sensitive user data. Refer: https://aka.ms/tmtauditlog#log-sensitive-data. Ensure that Audit and Log Files have Restricted Access. Refer: https://aka.ms/tmtauditlog#log-restricted-access</p> <p>SDL Phase: Implementation</p>
58. An adversary can reverse weakly encrypted or hashed content [State: Mitigation Implemented] [Priority: High]
<p>Category: Information Disclosure</p> <p>Description: An adversary can reverse weakly encrypted or hashed content</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Do not expose security details in error messages. Refer: https://aka.ms/tmmsgm#message. Implement Default error handling page. Refer: https://aka.ms/tmmsgm#default. Set Deployment Version in IIS. Refer: https://aka.ms/tmmsgm#deploy. Use only approved symmetric block cipher and key lengths. Refer: https://aka.ms/tmtpyc#keylength. Use approved block cipher model and initialization vectors for symmetric ciphers. Refer: https://aka.ms/tmtpyc#vector-ciphers. Use approved asymmetric algorithms, key lengths, and padding. Refer: https://aka.ms/tmtpyc#padding. Use approved random number generators. Refer: https://aka.ms/tmtpyc#random-number-generators. Do not use symmetric stream ciphers. Refer: https://aka.ms/tmtpyc#stream-ciphers. Use approved MAC/HMAC/Keyed hash algorithms. Refer: https://aka.ms/tmtpyc#mac-hash. Use only approved cryptographic hash functions. Refer: https://aka.ms/tmtpyc#hash-functions. Verify X.509 certificates used to authenticate SSL/TLS and DTLS connections. Refer: https://aka.ms/tmtpyc#ssl-tls-dtls</p> <p>SDL Phase: Implementation</p>
59. An adversary may bypass critical steps or perform actions on behalf of other users (victims) due to improper validation logic [State: Mitigation Implemented] [Priority: High]
<p>Category: Elevation of Privileges</p> <p>Description: Failure to validate the privileges and access rights to the application to individuals who require the privileges or access rights may result into unauthorized use of data due to inappropriate rights settings and validation.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that administrative interfaces are appropriately locked down. Refer: https://aka.ms/tmtauthz#admin-interface-lockdown. Enforce sequential step order when processing business logic flows. Refer: https://aka.ms/tmtauthz#sequential-logic. Ensure that proper authorization is in place and principle of least privilege is followed. Refer: https://aka.ms/tmtauthz#principle-least-privilege. Business logic and resource access authorization decisions should not be based on incoming request parameters. Refer: https://aka.ms/tmtauthz#logic-request-parameters. Ensure that content and resources are not enumerable or accessible via forceful browsing. Refer: https://aka.ms/tmtauthz#enumerable-browsing</p> <p>SDL Phase: Implementation</p>
60. An adversary can perform action on behalf of other user due to lack of controls against cross domain requests [State: Not Started] [Priority: High]
<p>Category: Denial of Service</p> <p>Description: Failure to restrict requests originating from third party domains may result in unauthorized actions or access of data</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that authenticated ASP.NET pages incorporate UI Redressing or clickjacking defences. Refer: https://aka.ms/tmtconfigmgmt#ui-defenses. Ensure that only trusted origins are allowed if CORS is enabled on ASP.NET Web Applications. Refer: https://aka.ms/tmtconfigmgmt#cors-aspnet. Mitigate against Cross-Site Request Forgery (CSRF) attacks on ASP.NET web pages. Refer: https://aka.ms/tmtconfigmgmt#asp</p> <p>...
Implementation</p>

Interaction: HTTPS Response

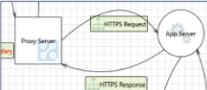


61. An adversary may gain unauthorized access to privileged features on Proxy Server [State: Mitigation Implemented] [Priority: High]
<p>Category: Elevation of Privileges</p> <p>Description: An adversary may get access to admin interface or privileged services like WiFi, SSH, File shares, FTP etc., on a device</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that all admin interfaces are secured with strong credentials. Refer: https://aka.ms/tmtconfigmgmt#admin-strong</p> <p>SDL Phase: Implementation</p>
62. An adversary may exploit unused services or features in Public Internet [State: Not Started] [Priority: High]
<p>Category: Elevation of Privileges</p> <p>Description: An adversary may use unused features or services on Public Internet such as UI, USB port etc. Unused features increase the attack surface and serve as additional entry points for the adversary</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that only the minimum services/features are enabled on devices. Refer: https://aka.ms/tmtconfigmgmt#min-enable</p> <p>SDL Phase: Implementation</p>
63. An adversary may exploit known vulnerabilities in unpatched devices [State: Not Started] [Priority: High]
<p>Category: Tainting</p> <p>Description: An adversary may leverage known vulnerabilities and exploit a device if the firmware of the device is not updated</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date. Refer: https://aka.ms/tmtconfigmgmt#cloud-firmware</p> <p>SDL Phase: Design</p>
64. An adversary may tamper Proxy Server and extract cryptographic key material from it [State: Not Started] [Priority: High]
<p>Category: Tainting</p> <p>Description: An adversary may partially or wholly replace the software running on Public Internet, potentially allowing the replaced software to leverage the genuine identity of the device if the key material or the cryptographic facilities holding key materials were available to the illicit program. For example an attacker may leverage extracted key material to intercept and suppress data from the device on the communication path and replace it with false data that is authenticated with the stolen key material.</p> <p>Justification: <no mitigation provided></p> <p>Possible Mitigation(s): Store Cryptographic keys securely on IoT Device. Refer: https://aka.ms/tmtcrypt#keys-iot</p>

65. An adversary may tamper the OS of a device and launch offline attacks [State: Not Started] [Priority: High]

Category: Tampering
 Description: An adversary may launch offline attacks made by disabling or circumventing the installed operating system, or made by physically separating the storage media from the device in order to attack the data separately.
 Justification: <no mitigation provided>
 Possible Mitigation(s): Encrypt OS and additional partitions of IoT Device with BitLocker. Refer: <https://aka.ms/bmtconfigmgmt#partition-iot>
 SDL Phase: Design

Interaction: HTTPS Response



66. An adversary may execute unknown code on Proxy Server [State: Mitigation Implemented] [Priority: High]

Category: Tampering
 Description: An adversary may launch malicious code into Proxy Server and execute it.
 Justification: <no mitigation provided>
 Possible Mitigation(s): Ensure that unknown code cannot execute on devices. Refer: <https://aka.ms/bmtconfigmgmt#unknown-eie>
 SDL Phase: Design

Threats, Mitigations Suggested and Mitigations Implemented

01. User authentication

- Auth Token expire
- Tokens expire

02. Access control -

- User privileges

03. Integrity -

- Validating the user inputs to prevent xss
- Using middleware validating user inputs

04. Confidentiality -

- Message encrypted
- Password hashing

05. Message authentication -

- Handling the error messages
- Handling try catch method

Testing Plan and Results

Test Plan

- Manual Testing

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.

- Functional Testing – **Sivarajah. K**
- API Testing – **Sivashankar. S**
- Burp Suite Security Testing – **Soujanyaah. K**

Burp Suite is an integrated platform/graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

- Password Hashing Testing – **Viknatharshan. N**

Password Hashing makes password storage and management more secure. Hashing uses a formula to transform a password into a predictable, yet encrypted form that obscures the actual password and makes it much harder for bad actors to decipher it.

Test Results

Manual Testing

Functional Testing

Test Case No	Test Case 01
Description	Login as an admin with valid credentials.
Test Steps	<ol style="list-style-type: none">1. Go to the login page.2. Enter the valid email address and password.3. Click the “Login” button.
Expected Result	Should display the “Add New Staff” form.
Actual Result	Displayed the “Add New Staff” form.
Test Status	Pass
User Role	Admin

Test Case No	Test Case 02
Description	Login as an admin with invalid credentials.
Test Steps	<ol style="list-style-type: none">1. Go to the login page.2. Enter the invalid email address and password.3. Click the “Login” button.
Expected Result	<i>Should display the error message “Something went wrong Server error please reload”.</i>

Actual Result	Displayed the error message “Something went wrong Server error please reload”.
Test Status	Pass
User Role	Admin

Test Case No	Test Case 03
Description	Check the required field validation of login form
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the email address and put the password field as empty. 3. Click the “Login” button.
Expected Result	<i>Should display the error message “Please fill out this field”.</i>
Actual Result	Displayed the error message “Please fill out this field”.
Test Status	Pass
User Role	Admin

Test Case No	Test Case 04
Description	Check the required field validation of add new staff form
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the email address and password.

	<p>3. Click the “Login” button. (Navigate to add new staff form)</p> <p>4. Click the “Create New Account” button without filling any field in “Add New Staff” form.</p>
Expected Result	<i>Should display the error message “Please fill out this field”.</i>
Actual Result	Displayed the error message “Please fill out this field”.
Test Status	Pass
User Role	Admin

Test Case No	Test Case 05
Description	Create an account for worker and manager.
Test Steps	<p>1. Go to the login page.</p> <p>2. Enter the email address and password.</p> <p>3. Click the “Login” button. (Navigate to add new staff form)</p> <p>4. Enter the email, first name, last name, password and select the user role.</p> <p>5. Click the “Create New Account” button.</p>

Expected Result	<i>Should display the success message “User Added Successfully!”.</i>
Actual Result	Displayed the success message “User Added Successfully!”.
Test Status	Pass
User Role	Admin

Test Case No	Test Case 06
Description	Login as a manager with valid credentials.
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the valid email address and password. 3. Click the “Login” button.
Expected Result	Should display the message and upload file fields.
Actual Result	Displayed the message and upload file fields.
Test Status	Pass
User Role	Manager

Test Case No	Test Case 07
--------------	--------------

Description	Login as a manager with invalid credentials.
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the invalid email address and password. 3. Click the “Login” button.
Expected Result	<i>Should display the error message “Something went wrong Server error please reload”.</i>
Actual Result	Displayed the error message “Something went wrong Server error please reload”.
Test Status	Pass
User Role	Manager

Test Case No	Test Case 08
Description	Click the “Save message” button after enter the message.
Test Steps	<ol style="list-style-type: none"> 1. Login as a manager. 2. Enter the message in message field. 3. Click the “Save message” button.
Expected Result	<i>Should display the success message “Message has been saved”.</i>
Actual Result	Displayed the success message “Message has been saved”.
Test Status	Pass

User Role	Manager
-----------	---------

Test Case No	Test Case 09
Description	Click the “Upload File” button without selecting any file.
Test Steps	<ol style="list-style-type: none"> 1. Login as a manager. 2. Click the “Upload file” button.
Expected Result	<i>Should display the error message “Please select a file to upload”.</i>
Actual Result	Displayed the error message “Please select a file to upload”.
Test Status	Pass
User Role	Manager

Test Case No	Test Case 10
Description	Click the “Upload File” button after select any file.
Test Steps	<ol style="list-style-type: none"> 1. Login as a manager. 2. Choose any file. 3. Click the “Upload file” button.
Expected Result	<i>Should display the success message “File has been uploaded, Successfully”.</i>
Actual Result	Displayed the success message “File has been uploaded, Successfully”.

Test Status	Pass
User Role	Manager

Test Case No	Test Case 11
Description	Login as a worker with valid credentials.
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the valid email address and password. 3. Click the “Login” button.
Expected Result	Should display the message field.
Actual Result	Displayed the message field.
Test Status	Pass
User Role	Worker

Test Case No	Test Case 12
Description	Login as a worker with invalid credentials.
Test Steps	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter the invalid email address and password. 3. Click the “Login” button.

Expected Result	<i>Should display the error message “Something went wrong Server error please reload”.</i>
Actual Result	Displayed the error message “Something went wrong Server error please reload”.
Test Status	Pass
User Role	Worker

Test Case No	Test Case 13
Description	Click “Save message” button after enter the message.
Test Steps	<ol style="list-style-type: none"> 1. Login as a worker. 2. Enter the message in message field. 3. Click the “Save message” button.
Expected Result	<i>Should display the success message “Message has been saved”.</i>
Actual Result	Displayed the success message “Message has been saved”.
Test Status	Pass
User Role	Worker

Test Case No	Test Case 14
Description	Click “Save message” button without entering the message.

Test Steps	<ol style="list-style-type: none"> 1. Login as a worker. 2. Click the “Save message” button.
Expected Result	<i>Should display the error message “Message Field is Empty”.</i>
Actual Result	Displayed the error message “Message Field is Empty”.
Test Status	Pass
User Role	Worker

Test Case No	Test Case 15
Description	Click “Save message” button without entering the message.
Test Steps	<ol style="list-style-type: none"> 1. Login as a manager. 2. Click the “Save message” button.
Expected Result	<i>Should display the error message “Message Field is Empty”.</i>
Actual Result	Displayed the error message “Message Field is Empty”.
Test Status	Pass
User Role	Manager

API Testing

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:5000/api/auth/signup`. The response status is 200 OK, time 233 ms, size 547 B. The response body is a JSON object:

```
1 "token": "eyJhbGciOiJIUzI1NiIsInR5cC1kXVCIJ9.eyJpc3MiOiJsb2dpbiIsInN1YiI6ImhbmFnZXIifSwiaWF0IjoxNjY4NjEyNTAwLCJleHAiOjE2Njg5NzI1MD89.3ILQnQGvS0JavJw41vG3DT5sSngFkMIVkGAz4kCJs",  
2 "name": "vs",  
3 "id": "637501946b24d42174871957",  
4 "role": "manager"
```

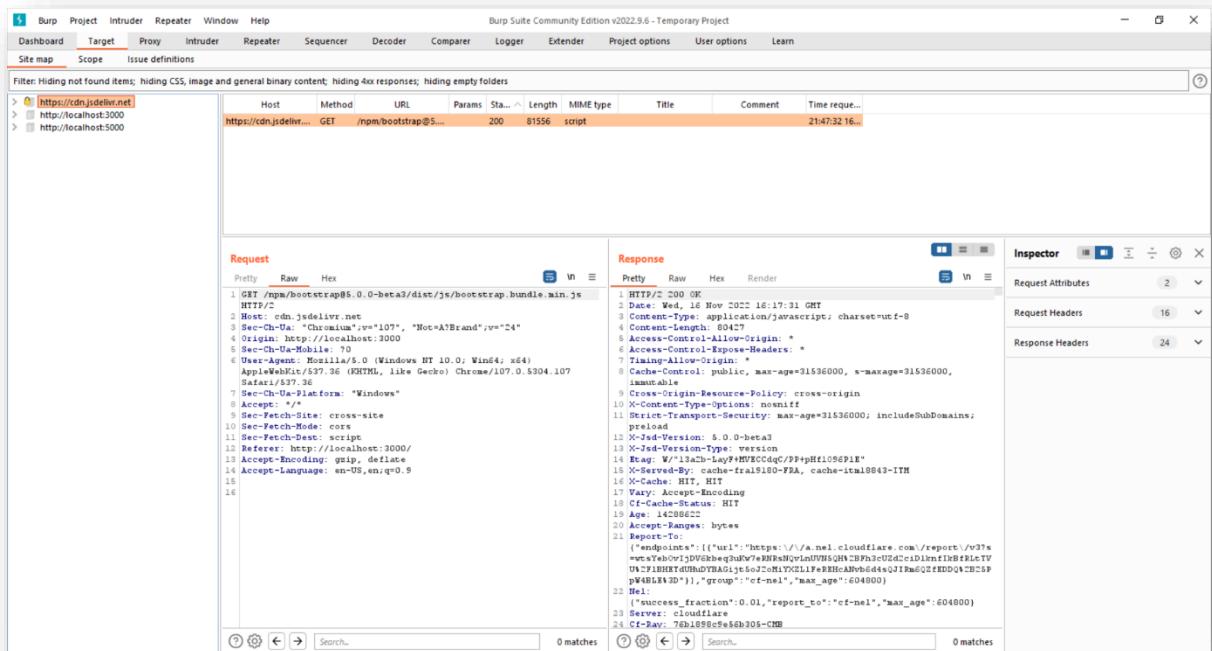
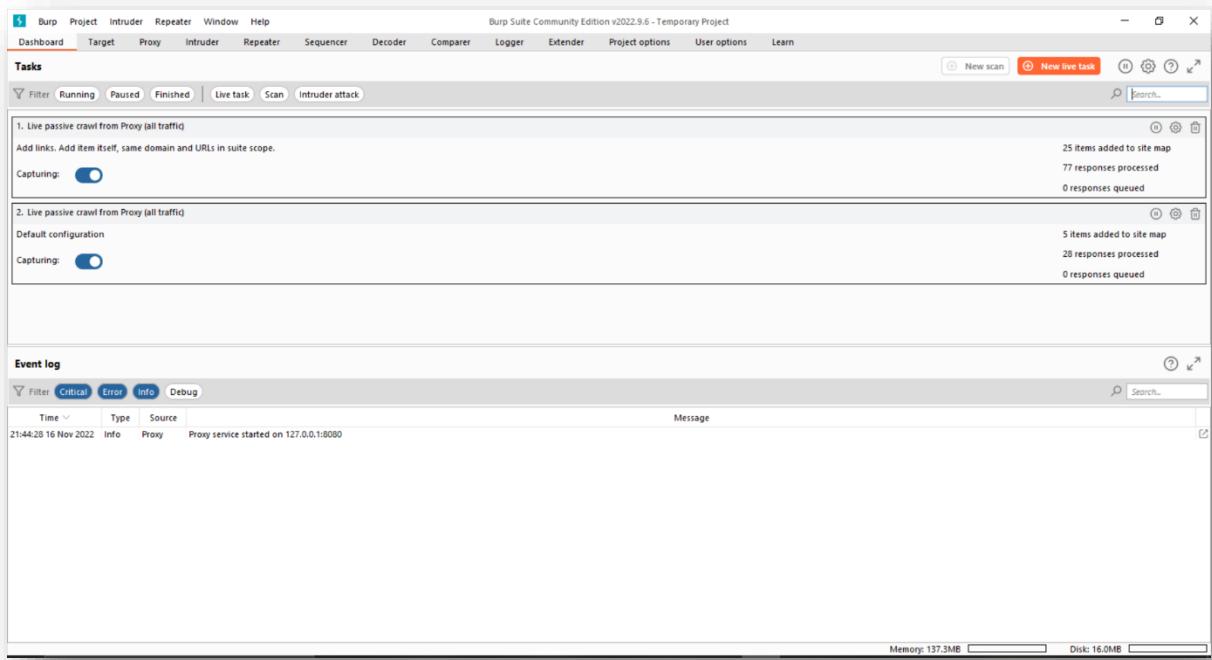
The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:5000/api/auth/save`. The response status is 200 OK, time 318 ms, size 281 B. The response body is a JSON object:

```
1 "message": "hi"
```

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:5000/api/auth/login`. The request body is JSON, containing `{"email": "manager123@gmail.com", "password": "123456"}`. The response status is 404 Not Found, with the message `"msg": "No user found for this email."`. The operating system taskbar at the bottom shows it's 10:40 PM on 11/16/2022.

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:5000/api/auth/login`. The request body is JSON, containing `{"email": "manager@gmail.com", "password": "123456"}`. The response status is 200 OK, with the message `"msg": "login successfully!"`. The operating system taskbar at the bottom shows it's 10:42 PM on 11/16/2022.

Burp Suite Security Testing



Burp Suite Community Edition v2022.9.6 - Temporary Project

Site map Scope Issue definitions

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requ...
http://localhost:3000	GET	/sockjs-node		101	129				21:50:12 16...
http://localhost:3000	GET	/		304	173	HTML			21:50:11 16...
http://localhost:3000	GET	/favicon.ico		304	237				21:49:25 16...
http://localhost:3000	GET	/logo192.png		304	238				21:50:13 16...
http://localhost:3000	GET	/main.js		304	237				21:50:14 16...
http://localhost:3000	GET	/static/js/bundle.js		304	174	script			21:50:11 16...
http://localhost:3000	GET	/static/js/main.chun...		304	175	JSON			21:50:11 16...
http://localhost:3000	GET	/static/js/vendors~m...		304	176	JSON			21:50:15 16...
http://localhost:3000	GET	/static/js/vendors~m...		304	176				21:50:16 16...

Request

```

1 GET /sockjs-node HTTP/1.1
2 Host: localhost:3000
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
7 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107
8 Safari/537.36
9 Upgrade: websocket
10 Origin: http://localhost:3000
11 Sec-WebSocket-Version: 13
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Sec-WebSocket-Key: 0GcaH0PU9vh0QoObCSBfAA==
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
645
646
647
647
648
649
649
650
651
652
653
654
655
656
656
657
658
658
659
659
660
661
662
663
664
665
665
666
667
667
668
668
669
669
670
671
672
673
674
675
675
676
677
677
678
678
679
679
680
681
682
683
684
684
685
685
686
686
687
687
688
688
689
689
690
691
692
693
694
694
695
695
696
696
697
697
698
698
699
699
700
701
702
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488

```

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

	Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requ...
1	http://cdn.jsdelivr.net	POST	/api/auth/login		✓ 200	1588	JSON			21:47:46 16...
2	http://localhost:3000	POST	/api/auth/login		✓ 200	1663	JSON			21:49:46 16...
3	http://localhost:5000	POST	/api/auth/login		✓ 200	709	JSON			21:49:03 16...
4	http://localhost:5000	OPTION...	/api/auth/login		✓ 204	301				21:49:46 16...

Request

Pretty	Raw	Hex	Header
1	POST /api/auth/login HTTP/1.1		
2	Host: localhost:5000		
3	Content-Length: 49		
4	sec-ch-ua: "Chromium";v="107", "Not A?Brand";v="24"		
5	Accept: application/json, text/plain, */*		
6	Accept-Type: application/json		
7	sec-ch-ua-mobile: ?0		
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36		
9	sec-ch-ua-platform: "Windows"		
10	Origin: http://localhost:3000		
11	Sec-Fetch-Site: same-site		
12	Sec-Fetch-Mode: cors		
13	Sec-Fetch-Dest: empty		
14	Referer: http://localhost:2000/		
15	Accept-Encoding: gzip, deflate		
16	Accept-Language: en-US,en;q=0.9		
17	Connection: close		
18			
19	{		
20	"email": "admin@gmail.com",		
21	"password": "admin123"		

Response

Pretty	Raw	Hex	Header
1	HTTP/1.1 200 OK		
2	X-Powered-By: Express		
3	Access-Control-Allow-Origin: *		
4	Content-Type: application/json; charset=utf-8		
5	Content-Length: 1346		
6	Tag: 1424:00000000000000000000000000000000		
7	Date: Wed, 16 Nov 2022 16:17:46 GMT		
8	Connection: close		
9			
10	{		
11	"token":		
12	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleCiyIp7ImlkIjoiMjM3NGJyNTc3MGUJMAlmIrg0ZT7RcOGI1l1wicmVzZXlG1mPbW1nWosTaRhDlGNTY2ODYzNTc3MjE2Owv1jouHjY4tC1mDYZfQ._mSwIHvERECG2AASftUt1clqunllAAm3H7p3pewN0V"		
13	"firstName": "admin",		
14	"lastName": "admin",		
15	"id": "6374bc7f50e5205504e1d8be",		
16	"email": "admin@gmail.com",		
17	"firstName": "admin",		
18	"lastName": "admin",		
19	"messages": [
20	{		
21	"_id": "6374bc7f50e5205504e1d8bf",		
22	"message": "Quick"		
23	},		

Inspector

Request Attributes: 2

Request Headers: 16

Response Headers: 7

The screenshot shows the Burp Suite interface with the following details:

Request (POST /api/auth/login):

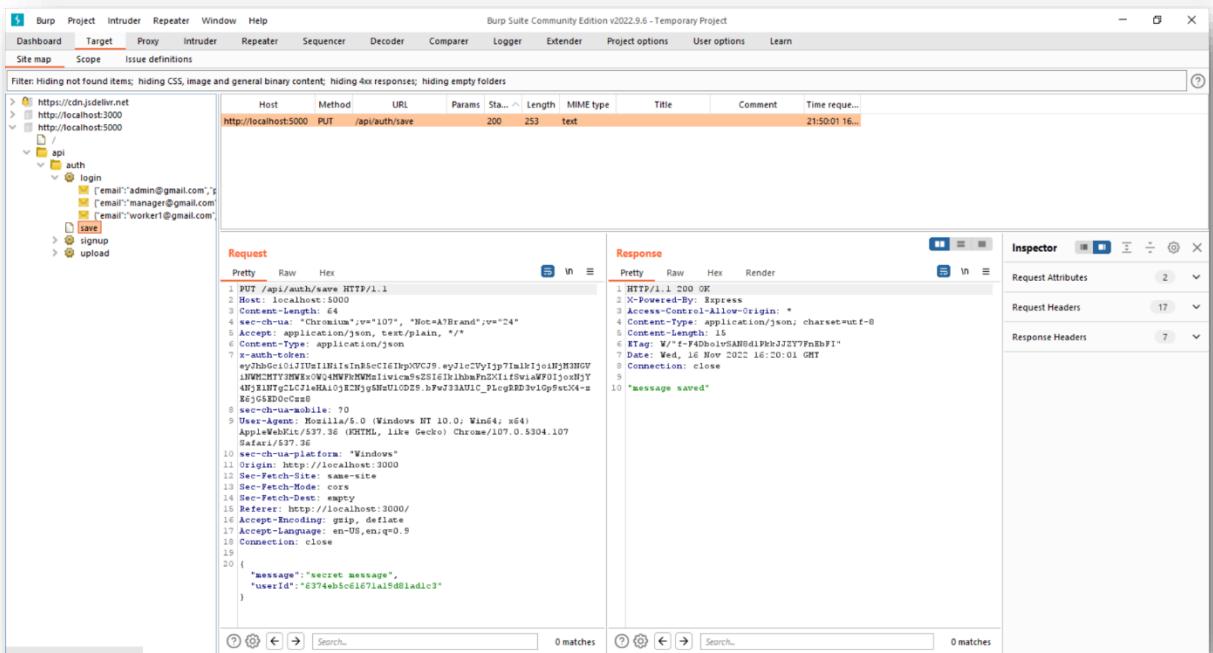
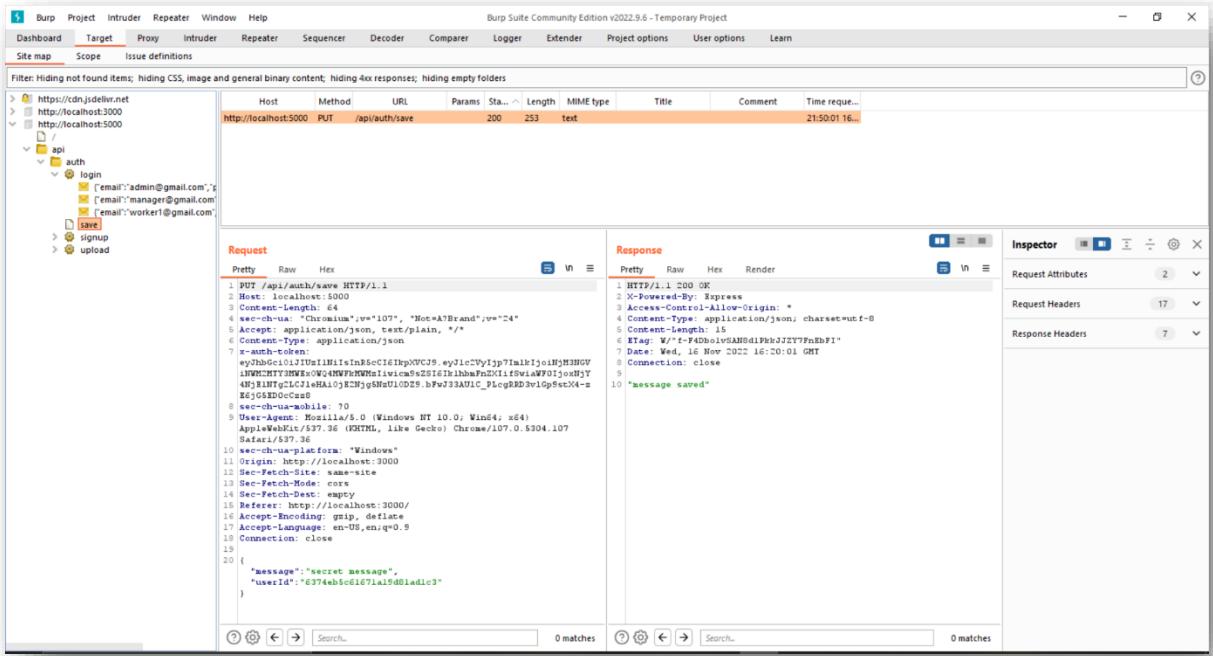
```
POST /api/auth/login HTTP/1.1
Host: localhost:5000
Content-Type: application/json
Accept: application/json, text/plain, */*
Content-Length: 49
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Sec-CH-UA: "Chromium";v="107", "Not A Brand";v="24"
Sec-CH-UA-Mobile: ?0
Sec-CH-UA-Platform: "Windows"
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:5000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: close

{
  "email": "admin@gmail.com",
  "password": "admin123"
}
```

Response (HTTP/1.1 200 OK):

```
HTTP/1.1 200 OK
Date: Wed, 16 Nov 2022 18:17:46 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1346
ETag: W/"54c-1d9c90210d0e839909ab07qwfgr"
Server: Werkzeug/2.0.7 Python/3.10.6
Set-Cookie: session_id=6274c36090e5205504e1d0bf; expires=Wed, 16-Nov-2022 18:17:46 GMT; path=/; secure; HttpOnly
Connection: close

{
  "token": "eyJhbGciOiJIUzI1NiIsInBkIjoiCg18IpKjVCJ36_wyJlcCPyIjp7ImkiIjoiNjM3NGJyZTgXMGUjMaIuHtgZG7Fk9G2l11wicm+2zS1m7hW1uIno1alndIGMTyCDYmH7CjNiwjZDwvlijosHtY40TclINEYCfQ__m8wIhV8EJCjCaASFcQtldTlgunLAAmGHS3pjeNBw",
  "user": {
    "id": "6274c36090e5205504e1d0bf",
    "username": "admin",
    "lastName": "admin",
    "id": "6274c36090e5205504e1d0bf",
    "role": "admin",
    "user": {
      "gender": "Male",
      "role": "admin",
      "_id": "6274c36090e5205504e1d0bf",
      "email": "admin@gmail.com",
      "firstName": "admin",
      "lastName": "admin",
      "messages": [
        {
          "_id": "6274c36090e5205504e1d0bf",
          "message": "Quick"
        }
      ],
      "token": "eyJhbGciOiJIUzI1NiIsInBkIjoiCg18IpKjVCJ36_wyJlcCPyIjp7ImkiIjoiNjM3NGJyZTgXMGUjMaIuHtgZG7Fk9G2l11wicm+2zS1m7hW1uIno1alndIGMTyCDYmH7CjNiwjZDwvlijosHtY40TclINEYCfQ__m8wIhV8EJCjCaASFcQtldTlgunLAAmGHS3pjeNBw"
    }
  }
}
```



Burp Suite Community Edition v2022.9.6 - Temporary Project

Target

Host: http://localhost:5000

Method: POST /api/auth/signup

Params: ✓ 200 522 Length: MIME type: JSON

Time requ...

Request

```
POST /api/auth/signup HTTP/1.1
Host: localhost:5000
Content-Length: 108
sec-ch-ua: "Chromium";v="107", "Not A Brand";v="24"
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
sec-ch-ua-mobile: 0
Origin: http://localhost:3000
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
{
  "firstName": "worker1",
  "lastName": "work",
  "password": "worker1",
  "email": "worker1@gmail.com",
  "role": "Worker"
}
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 281
ETag: W/"119-prdrE85pnshVUOrtGYPBcgt+eCUA"
Date: Wed, 16 Nov 2022 16:18:33 GMT
Connection: close
{
  "token": "eyJhbGciOiJIUzI1NiIsInBkIjoiMjM3MjHgJjM2YXM0MjM1MjM1NTg0ZTRkOGJ1Iiwiexp3ZS16iFbWluiMhC1SHTYODyJN7Qniw1ZdwijjsnY407e1MDYzTf0..._m9d5V1HVRCECG2a8PfcQitCtgwniAkA3Hg...",
  "name": "worker1",
  "id": "c2750a59321ca25714cabde0",
  "role": "Worker"
}
```

Inspector

Request Attributes: 2

Request Headers: 17

Response Headers: 7

Burp Suite Community Edition v2022.9.6 - Temporary Project

Target

Host: http://localhost:5000

Method: POST /api/auth/upload

Params: ✓ 201 284 Length: MIME type: JSON

Time requ...

Request

```
POST /api/auth/upload HTTP/1.1
Host: localhost:5000
Content-Length: 62926
sec-ch-ua: "Chromium";v="107", "Not A Brand";v="24"
Accept: application/json, text/plain, */*
Content-Type: multipart/form-data;
boundary: -----WebKitFormBoundary4tZ1XKQm8SDgNG
x-auth-token: eyJhbGciOiJIUzI1NiIsInBkIjoiMjM3MjHgJjM2YXM0MjM1MjM1NTg0ZTRkOGJ1Iiwiexp3ZS16iFbWluiMhC1SHTYODyJN7Qniw1ZdwijjsnY407e1MDYzTf0..._m9d5V1HVRCECG2a8PfcQitCtgwniAkA3Hg...
Origin: http://localhost:3000
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
-----WebKitFormBoundary4tZ1XKQm8SDgNG
Content-Disposition: form-data; name="file"; filename="DDOS.png"
Content-Type: image/png
23
```

Response

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 40
ETag: W/"28-3GVqgCD0gq4KEd1KdgdyTqIAp4"
Date: Wed, 16 Nov 2022 16:20:05 GMT
Connection: close
{
  "status": "File uploaded, successfully"
}
```

Inspector

Request Attributes: 2

Request Body Parameters: 2

Request Headers: 17

Response Headers: 7

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
Filter: Hiding CSS, Image and general binary content																
1	http://localhost:3000	GET	/			200	2874	HTML		ABC Company		127.0.0.1			21:45:49 16... 8080	
2	http://localhost:3000	GET	/			200	2874	HTML		ABC Company		127.0.0.1			21:47:05 16... 8080	
6	https://cdn.jsdelivr.net	GET	/npm/bootstrap@5.0.0-beta3/dl...			200	81556	script	js			✓ 104.16.86.20			21:47:31 16... 8080	
7	http://localhost:3000	GET	/static/js/bundle.js			200	39042	script	js			127.0.0.1			21:47:31 16... 8080	
9	http://localhost:3000	GET	/static/js/main.chunk.js			200	129116	JSON	js			127.0.0.1			21:47:31 16... 8080	
10	http://localhost:3000	GET	/sockjs-node		101	129						127.0.0.1			21:47:32 16... 8080	
11	https://fonts.gstatic.com	GET	/s/poppins/v20/pxByp8kv8JHgF...			200	8562	woff2				✓ 142.250.182.227			21:47:32 16... 8080	
12	https://fonts.gstatic.com	GET	/s/poppins/v20/pxByp8kv8JHgF...			200	8814	woff2				✓ 142.250.182.227			21:47:32 16... 8080	
13	http://localhost:3000	GET	/manifest.json			304	237	script	json			127.0.0.1			21:47:33 16... 8080	
15	http://localhost:5000	OPTIO...	/api/auth/login	✓		200	301					127.0.0.1			21:47:45 16... 8080	
16	http://localhost:5000	POST	/api/auth/login			200	1580	JSON				127.0.0.1			21:47:45 16... 8080	
17	https://passwordleakche...	POST	/v1/leaks/lookupSingle	✓		400	639	script				✓ 142.250.77.74			21:47:46 16... 8080	
18	https://passwordleakche...	POST	/v1/leaks/lookupSingle			200	314					127.0.0.1			21:48:31 16... 8080	
19	http://localhost:5000	POST	/api/auth/signup	✓		200	322	JSON				127.0.0.1			21:48:31 16... 8080	
20	https://passwordleakche...	POST	/v1/leaks/lookupSingle			400	639	script				✓ 142.250.77.74			21:48:50 16... 8080	
21	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:48:50 16... 8080	
22	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:48:50 16... 8080	
23	http://localhost:3000	GET	/static/js/vendors~main.chunk.j...			304	176	map				127.0.0.1			21:48:50 16... 8080	
24	http://localhost:3000	GET	/static/js/vendors~main.chunk.j...			304	176	map				127.0.0.1			21:48:50 16... 8080	
25	http://localhost:3000	GET	/			304	173					127.0.0.1			21:48:54 16... 8080	
29	http://localhost:3000	GET	/static/js/bundle.js			304	174	script	js			127.0.0.1			21:48:54 16... 8080	
30	http://localhost:3000	GET	/static/js/main.chunk.js			304	175	script	js			127.0.0.1			21:48:55 16... 8080	
31	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:48:55 16... 8080	
32	http://localhost:3000	GET	/sockjs-node		101	129						127.0.0.1			21:48:55 16... 8080	
33	http://localhost:3000	GET	/manifest.json			304	237	script	json			127.0.0.1			21:49:02 16... 8080	
36	http://localhost:5000	OPTIO...	/api/auth/login			204	301					127.0.0.1			21:49:03 16... 8080	
37	http://localhost:5000	POST	/api/auth/login	✓		200	709	JSON				127.0.0.1			21:49:03 16... 8080	
38	https://fonts.gstatic.com	GET	/s/poppins/v20/pxByp8kv8JHgF...			200	8678	woff2				✓ 142.250.182.227			21:49:03 16... 8080	
39	https://passwordleakche...	POST	/v1/leaks/lookupSingle	✓		400	639	script				✓ 142.250.77.74			21:49:03 16... 8080	
40	http://localhost:5000	OPTIO...	/api/auth/save			204	314					127.0.0.1			21:49:16 16... 8080	
41	http://localhost:5000	POST	/api/auth/save	✓		200	253	text				127.0.0.1			21:49:16 16... 8080	
42	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:49:21 16... 8080	
43	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:49:21 16... 8080	
44	http://localhost:3000	GET	/static/js/vendors~main.chunk.j...			304	176	map				127.0.0.1			21:49:21 16... 8080	
45	http://localhost:3000	GET	/static/js/vendors~main.chunk.j...			304	176	map				127.0.0.1			21:49:24 16... 8080	
46	http://localhost:3000	GET	/			304	173					127.0.0.1			21:49:24 16... 8080	
50	http://localhost:3000	GET	/static/js/bundle.js			304	174	script	js			127.0.0.1			21:49:24 16... 8080	
51	http://localhost:3000	GET	/static/js/main.chunk.js			304	175	script	js			127.0.0.1			21:49:24 16... 8080	
52	http://localhost:3000	GET	/static/js/vendors~main.chunk.js			304	176	script	js			127.0.0.1			21:49:24 16... 8080	
53	http://localhost:3000	GET	/manifest.json			304	237	script	json			127.0.0.1			21:49:25 16... 8080	
54	http://localhost:3000	GET	/sockjs-node		101	129						127.0.0.1			21:49:25 16... 8080	
57	http://localhost:4000	OPTIO...	/api/auth/login			204	511					127.0.0.1			21:49:48 16... 8080	

#	URL	Direction	Edited	Length	Comment	TLS	Time	Listener port	WebSocket ID
Filter: Showing all items									
2	http://localhost:3000/sockjs-node	← To client	34				21:47:32 16... 8000	1	
3	http://localhost:3000/sockjs-node	← To client	14				21:47:32 16... 8000	1	
4	http://localhost:3000/sockjs-node	← To client	21				21:47:32 16... 8000	1	
5	http://localhost:3000/sockjs-node	← To client	45				21:47:32 16... 8000	1	
6	http://localhost:3000/sockjs-node	← To client	957				21:47:32 16... 8000	1	
7	http://localhost:3000/sockjs-node	← To client	34				21:48:55 16... 8000	2	
8	http://localhost:3000/sockjs-node	← To client	14				21:48:55 16... 8000	2	
9	http://localhost:3000/sockjs-node	← To client	21				21:48:55 16... 8000	2	
10	http://localhost:3000/sockjs-node	← To client	45				21:48:55 16... 8000	2	
11	http://localhost:3000/sockjs-node	← To client	34				21:49:25 16... 8000	3	
12	http://localhost:3000/sockjs-node	← To client	14				21:49:25 16... 8000	3	
13	http://localhost:3000/sockjs-node	← To client	21				21:49:25 16... 8000	3	
14	http://localhost:3000/sockjs-node	← To client	45				21:49:25 16... 8000	3	
15	http://localhost:3000/sockjs-node	← To client	957				21:49:25 16... 8000	3	
16	http://localhost:3000/sockjs-node	← To client	34				21:50:12 16... 8000	4	
17	http://localhost:3000/sockjs-node	← To client	14				21:50:12 16... 8000	4	
18	http://localhost:3000/sockjs-node	← To client	21				21:50:12 16... 8000	4	
19	http://localhost:3000/sockjs-node	← To client	45				21:50:12 16... 8000	4	
20	http://localhost:3000/sockjs-node	← To client	957				21:50:12 16... 8000	4	

Proxy

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Add	Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/> 127.0.0.1:8080			Per-host	Default	
<input type="button" value="Remove"/>						

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

Intercept requests based on the following rules: *Master interception is turned off*

Add	Enabled	Operator	Match type	Relationship	Condition
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...	
<input type="button" value="Remove"/>		Or	Request	Contains parameters	
<input type="button" value="Up"/>		Or	HTTP method	Does not match	(get post)
<input type="button" value="Down"/>		And	URL	Is in target scope	

Automatically fix missing or superfluous new lines at end of request
 Automatically update Content-Length header when the request is edited

Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

Intercept responses based on the following rules: *Master interception is turned off*

Issue Definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
Path traversal	High	0x010210
SQL injection (second order)	High	0x01020280
ASP.NET tracing enabled	High	0x010040
File path traversal	High	0x010030
XML external entity injection	High	0x010040
LDAP injection	High	0x010050
XPath injection	High	0x010060
XML injection	Medium	0x010070
ASP.NET debugging enabled	Medium	0x010080
Out-of-bounds read/write (method)	Medium	0x010090
Out-of-bounds read/write (HTTP)	High	0x010080
File path manipulation	High	0x010080
PHP code injection	High	0x010090
Server-side JavaScript code injection	High	0x010040
Perl code injection	High	0x010060
Ruby code injection	High	0x010070
Python code injection	High	0x010010
Expression Language injection	High	0x010020
Universal code injection	High	0x010030
Server-side template injection	High	0x010100
SSI injection	High	0x010110
Cross-site scripting (stored)	High	0x020010
HTTP request smuggling	High	0x0200140
Client-side dnsync	High	0x0200141
Web cache poisoning	High	0x0200180
HTTP response header injection	High	0x0200200
Cross-site scripting (reflected)	High	0x0200300
Client-side template injection	High	0x0200308
Cross-site scripting (DOM-based)	High	0x0200309
Cross-site scripting (reflected DOM-based)	High	0x0200311
Cross-site scripting (stored DOM-based)	High	0x0200312
Client-side prototype pollution	Information	0x0200316
JavaScript injection (DOM-based)	High	0x0200320
JavaScript injection (reflected DOM-based)	High	0x0200321
JavaScript injection (stored DOM-based)	High	0x0200322
Path-relative style sheet import	Information	0x0200328
Client-side SQL injection (DOM-based)	High	0x0200330
Client-side SQL injection (reflected DOM-based)	High	0x0200331
Client-side SQL injection (stored DOM-based)	High	0x0200332

OS command injection

Description

Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into a command that is processed by a shell command interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that is executed, and inject arbitrary further commands that will be executed by the server.

OS command injection vulnerabilities are usually very serious and may lead to compromise of the server hosting the application, or of the application's own data and functionality. It may also be possible to use the server as a platform for attacks against other systems. The exact potential for exploitation depends upon the security context in which the command is executed, and the privileges that this context has regarding sensitive resources on the server.

Remediation

If possible, applications should avoid incorporating user-controllable data into operating system commands. In almost every situation, there are safer alternative methods of performing server-level tasks, which cannot be manipulated to perform additional commands than the one intended.

If it is considered unavoidable to incorporate user-supplied data into operating system commands, the following two layers of defense should be used to prevent attacks:

- The user data should be strictly validated. Ideally, a whitelist of specific accepted values should be used. Otherwise, only short alphanumeric strings should be accepted. Input containing any other data, including any conceivable shell metacharacter or whitespace, should be rejected.
- The application should use command APIs that launch a specific process via its name and command-line parameters, rather than passing a command string to a shell interpreter that supports command chaining and redirection. For example, the Java API Runtime.exec and the ASP.NET API Process.Start do not support shell metacharacters. This defense can mitigate the impact of an attack even in the event that an attacker circumvents the input validation defenses.

References

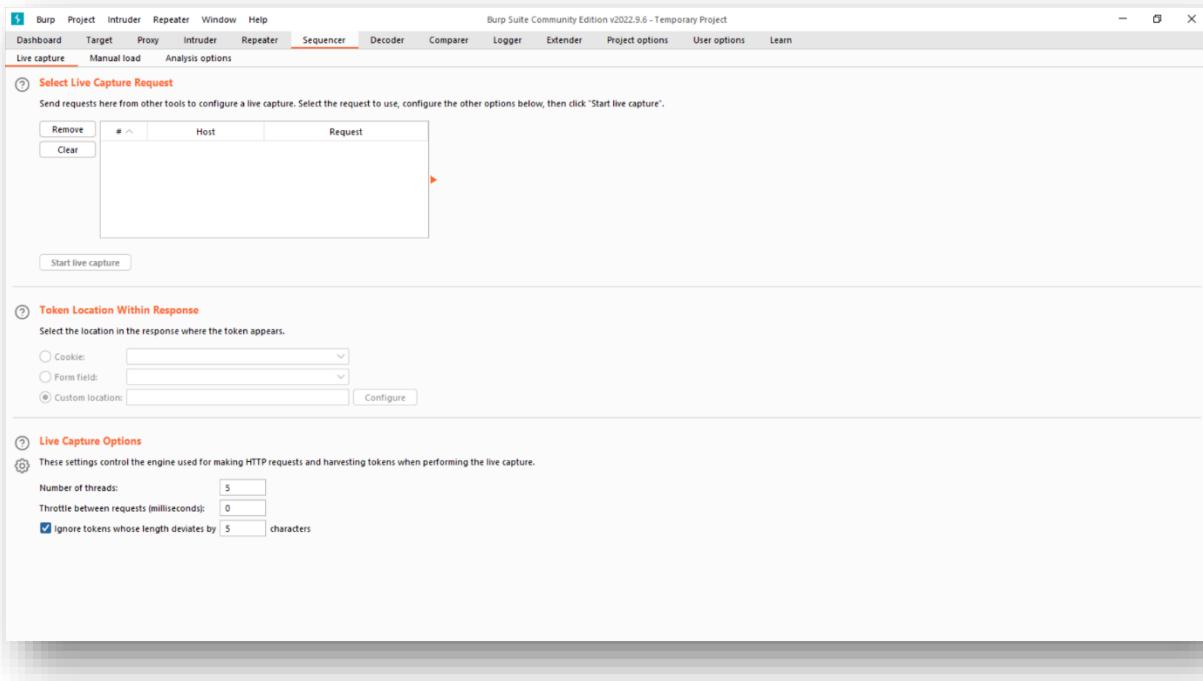
- [Web Security Academy: OS command injection](#)

Vulnerability classifications

- [CWE-77: Improper Neutralization of Special Elements used in a Command \('Command Injection'\)](#)
- [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [SAFEC-248: Command Injection](#)

Typical severity

High



#	Time	Tool	Method	Host	Path	Query	Param count	Status	Length	Start response timer	Comment
1	21:47:30 16 Nov 2022	Proxy	GET	localhost	/		0	200	2874	3	
2	21:47:30 16 Nov 2022	Proxy	GET	localhost	/		0	200	2874	3	
3	21:47:31 16 Nov 2022	Proxy	GET	fonts.googleapis.com	/css2	family=Poppins:ital,...	2	200	11592	207	
4	21:47:31 16 Nov 2022	Proxy	GET	cdn.jsdelivr.net	/npm/bootstrap@5....		0	200	155985	362	
5	21:47:31 16 Nov 2022	Proxy	GET	pro.fontawesome.c...	/release/v5.10.0/css...		0	200	156831	83	
6	21:47:31 16 Nov 2022	Proxy	GET	cdn.jsdelivr.net	/npm/bootstrap@5....		0	200	81584	204	
7	21:47:31 16 Nov 2022	Proxy	GET	localhost	/static/js/bundle.js		0	200	39042		
9	21:47:32 16 Nov 2022	Proxy	GET	localhost	/static/js/main.chun...		0	200	129111		
11	21:47:32 16 Nov 2022	Proxy	GET	fonts.gstatic.com	/ipoppins/n20/pxlB...		0	200	8562	411	
12	21:47:32 16 Nov 2022	Proxy	GET	fonts.gstatic.com	/ipoppins/n20/pxlE...		0	200	8814	355	
13	21:47:33 16 Nov 2022	Proxy	GET	localhost	/manifest.json		0	304	237		
14	21:47:34 16 Nov 2022	Proxy	GET	localhost	/logo192.png		0	304	238		
15	21:47:45 16 Nov 2022	Proxy	OPTION...	localhost	/api/auth/login		0	204	301		
16	21:47:45 16 Nov 2022	Proxy	POST	localhost	/api/auth/login		2	200	1588	301	
17	21:47:46 16 Nov 2022	Proxy	POST	passwordleakcheck.../V1/leaks/leakupS...			8	400	639	385	
18	21:48:32 16 Nov 2022	Proxy	OPTION...	localhost	/api/auth/signup		0	204	314		
20	21:48:50 16 Nov 2022	Proxy	POST	localhost	/api/auth/signup		5	204	522	268	
20	21:48:50 16 Nov 2022	Proxy	POST	passwordleakcheck.../V1/leaks/leakupS...			4	400	639	903	
21	21:48:50 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors-m...		0	304	176		
22	21:48:50 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors-m...		0	304	176	6	
23	21:48:50 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors-m...		0	304	176	9	
24	21:48:50 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors-m...		0	304	176		
25	21:48:54 16 Nov 2022	Proxy	GET	localhost	/		0	304	173		
26	21:48:54 16 Nov 2022	Proxy	GET	cdn.jsdelivr.net	/npm/bootstrap@5....		0	200	155991	330	
27	21:48:54 16 Nov 2022	Proxy	GET	fonts.googleapis.com	/css2	family=Poppins:ital,...	2	200	11592	562	
28	21:48:54 16 Nov 2022	Proxy	GET	pro.fontawesome.c...	/release/v5.10.0/css...		0	200	156831	474	
29	21:48:54 16 Nov 2022	Proxy	GET	localhost	/static/js/bundle.js		0	304	173		
30	21:48:55 16 Nov 2022	Proxy	GET	localhost	/static/js/main.chun...		0	304	175	11	
31	21:48:55 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors-m...		0	304	176	9	
32	21:48:55 16 Nov 2022	Proxy	GET	localhost	/sockjs-node		0	101	129		
33	21:48:55 16 Nov 2022	Proxy	GET	localhost	/manifest.json		0	304	237		
34	21:48:55 16 Nov 2022	Proxy	GET	localhost	/favicon.ico		0	304	237		
35	21:48:56 16 Nov 2022	Proxy	GET	localhost	/logo192.png		0	304	238	1	
36	21:49:01 16 Nov 2022	Proxy	OPTION...	localhost	/api/auth/login		0	204	301		
37	21:49:03 16 Nov 2022	Proxy	POST	localhost	/api/auth/signup		2	200	797	455	
39	21:49:03 16 Nov 2022	Proxy	GET	fonts.gstatic.com	/ipoppins/n20/pxlB...		0	200	8678	795	
40	21:49:16 16 Nov 2022	Proxy	POST	passwordleakcheck.../V1/leaks/leakupS...			9	400	639	297	
41	21:49:16 16 Nov 2022	Proxy	OPTION...	localhost	/api/auth/login		0	204	314	1	
42	21:49:21 16 Nov 2022	Proxy	PUT	localhost	/api/auth/login		2	200	253	332	
							0	304	176	7	

Burp Suite Community Edition v2022.9.6 - Temporary Project

Capture filter: Logger memory limit set to 100MB | Capturing requests up to 1MB; capturing responses up to 1MB

View filter: Showing all items

Logging: On

Columns

#	Time	Tool	Method	Host	Path	Query	Param count	Status	Length	Start response timer	Comment
37	21:49:03 16 Nov 2022	Proxy	POST	localhost	/api/auth/login		2	200	709	455	
38	21:49:03 16 Nov 2022	Proxy	GET	fonts.gstatic.com	/s/woffs/v2/pxIB...		0	200	8678	795	
39	21:49:03 16 Nov 2022	Proxy	POST	passwordleakcheck...v1/leaks/lookupSIn...			9	400	639	297	
40	21:49:16 16 Nov 2022	Proxy	OPTION	localhost	/api/auth/ave		0	204	314	1	
41	21:49:16 16 Nov 2022	Proxy	PUT	localhost	/api/auth/ave		2	200	253	332	
42	21:49:21 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	7	
43	21:49:21 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	5	
44	21:49:21 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	5	
45	21:49:21 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	4	
46	21:49:24 16 Nov 2022	Proxy	GET	localhost	/		0	304	173		
47	21:49:24 16 Nov 2022	Proxy	GET	cdn.jsdelivr.net	/npm/bootstrap@5....		0	200	155889	313	
48	21:49:24 16 Nov 2022	Proxy	GET	fonts.googleapis.com	/rrz2/family=Poppinsital_...		2	200	11592	547	
49	21:49:24 16 Nov 2022	Proxy	GET	pro.fontawesome.c...	/releases/v5.10.0/css...		0	200	156831	467	
50	21:49:24 16 Nov 2022	Proxy	GET	localhost	/static/js/bundle.js		0	304	174		
51	21:49:24 16 Nov 2022	Proxy	GET	localhost	/static/js/main.chun...		0	304	175		
52	21:49:24 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176		
53	21:49:25 16 Nov 2022	Proxy	GET	localhost	/manifest.json		0	304	237	1	
54	21:49:25 16 Nov 2022	Proxy	GET	localhost	/logo192.png		0	304	129		
55	21:49:25 16 Nov 2022	Proxy	GET	localhost	/favicon.ico		0	304	237	2	
56	21:49:25 16 Nov 2022	Proxy	GET	localhost	/logo192.png		0	304	238	1	
57	21:49:46 16 Nov 2022	Proxy	OPTION	localhost	/api/auth/login		0	204	301	1	
58	21:49:46 16 Nov 2022	Proxy	POST	localhost	/api/auth/login		2	200	1663	270	
59	21:49:47 16 Nov 2022	Proxy	POST	passwordleakcheck...v1/leaks/lookupSIn...			6	400	639	750	
60	21:50:00 16 Nov 2022	Proxy	OPTION	localhost	/api/auth/ave		0	204	314	1	
61	21:50:02 16 Nov 2022	Proxy	PUT	localhost	/api/auth/ave		2	200	253	323	
62	21:50:09 16 Nov 2022	Proxy	OPTION	localhost	/api/auth/upload		0	204	301		
63	21:50:11 16 Nov 2022	Proxy	POST	localhost	/api/auth/upload		3	204	204	15	
64	21:50:11 16 Nov 2022	Proxy	GET	localhost	/		0	304	173	2	
65	21:50:11 16 Nov 2022	Proxy	GET	cdn.jsdelivr.net	/npm/bootstrap@5....		0	200	155887	359	
66	21:50:11 16 Nov 2022	Proxy	GET	fonts.googleapis.com	/rrz2/family=Poppinsital_...		2	200	11592	670	
67	21:50:11 16 Nov 2022	Proxy	GET	pro.fontawesome.c...	/releases/v5.10.0/css...		0	200	156831	501	
68	21:50:11 16 Nov 2022	Proxy	GET	localhost	/static/js/bundle.js		0	304	174	1	
69	21:50:11 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	6	
70	21:50:11 16 Nov 2022	Proxy	GET	localhost	/static/js/main.chun...		0	304	175	6	
71	21:50:12 16 Nov 2022	Proxy	GET	localhost	/sockjs-node		0	101	129		
72	21:50:12 16 Nov 2022	Proxy	GET	localhost	/manifest.json		0	304	237	4	
73	21:50:12 16 Nov 2022	Proxy	GET	localhost	/logo192.png		0	304	238	1	
74	21:50:15 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176		
75	21:50:15 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	10	
76	21:50:15 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	6	
77	21:50:16 16 Nov 2022	Proxy	GET	localhost	/static/js/vendors~m...		0	304	176	6	

Burp Suite Community Edition v2022.9.6 - Temporary Project

Connections HTTP TLS Sessions Misc

Platform Authentication

These settings are configured within user options but can be overridden here for this specific project.

Override user options

Upstream Proxy Servers

These settings are configured within user options but can be overridden here for this specific project.

Override user options

SOCKS Proxy

These settings are configured within user options but can be overridden here for this specific project.

Override user options

Timeouts

These settings specify the timeouts to be used for various network tasks. Values are in seconds. Set an option to zero or leave it blank to never timeout that task.

Connect:	120
Normal:	120
Open-ended responses:	10
Domain name resolution:	300
Failed domain name resolution:	60

Hostname Resolution

Add entries here to override your computer's DNS resolution.

Add	Enabled	Hostname	IP address
Edit			
Remove			

Inspector

Request Attributes 2 ^

Protocol **HTTP/1** HTTP/2

Name	Value
Method	POST >
Path	/api/auth/upload >

Request Body Parameters 2 ^

Request Headers 17 ^

Response Headers 7 ^

Inspector

Request Attributes 2 ^

Request Body Parameters 2 ^

Name	Value
file	OPNGIHDR... >
userId	6374eb5c61671a1... >

Request Headers 17 ^

Response Headers 7 ^

Inspector	
Request Headers 17 ^	
Name	Value
Host	localhost:5000 >
Content-Length	62936 >
sec-ch-ua	"Chromium";v="1..." >
Accept	application/json,... >
Content-Type	multipart/form-d... >
x-auth-token	eyJhbGciOiJIUzI1... >
sec-ch-ua-mobile	?0 >
User-Agent	Mozilla/5.0 (Win... >
sec-ch-ua-platform	"Windows" >
Origin	http://localhost:... >
Sec-Fetch-Site	same-site >
Sec-Fetch-Mode	cors >
Sec-Fetch-Dest	empty >
Referer	http://localhost:... >
Accept-Encoding	gzip, deflate >
Accept-Language	en-US,en;q=0.9 >
Connection	close >
Response Headers 7 ▼	
Request Attributes 2 ▼	
Request Body Parameters 2 ▼	
Request Headers 17 ▼	
Response Headers 7 ^	
Name	Value
X-Powered-By	Express >
Access-Control-Allow-Origin	*
Content-Type	application/json; ... >
Content-Length	40 >
ETag	W/"28-3GVFqs2D..." >
Date	Wed, 16 Nov 2022... >
Connection	close >

Password Hashing Testing

Valid Match

Bcrypt Hash Verifier

Plain Text

Hash

The supplied hash matches with supplied plain text

VERIFY HASH

Online Bcrypt Hashed Matcher

Enter the Bcrypt Hashed Password

G ↴

Enter the Plain Text Password

G ↴

Match

Result:

Password matched.

Invalid Match

Online Bcrypt Hashed Matcher

Enter the Bcrypt Hashed Password

\$2a\$10\$ezmf4AET8TGYoCmU98dqo.MTE4256f
p/LiM3G4zHls1mqxAWSqQvC

Enter the Plain Text Password

admin

Match

Result:

Invalid password

The screenshot shows a web-based password checker. At the top, it says "Online Bcrypt Hashed Matcher". Below that, there are two input fields: one for the "Bcrypt Hashed Password" containing a long string of characters starting with "\$2a\$10\$ezmf4AET8TGYoCmU98dqo.MTE4256f" and another for the "Plain Text Password" containing "admin". A blue button labeled "Match" is present. Below the inputs, the word "Result:" is followed by a box containing the text "Invalid password". The entire interface has a clean, modern design with a white background and light gray borders for the input fields.

Valid Match

Online Bcrypt Hashed Matcher

Enter the Bcrypt Hashed Password

Enter the Plain Text Password

Match

Result:

Password matched.

Invalid Match

Bcrypt Hash Verifier

Plain Text

Hash

The plain text does not match the supplied hash.

VERIFY HASH