ExxonMobil

India Distributor Management System Vulnerability Assessment

Findings Report

Test Completion Date: July 25th, 2025

Table of Contents

Finding	Defect ID	Vulnerability	Status	Closed Date
1	67354	Weak Password Change Functionality | Score: High		
2	67424	HTTP Basic Authentication | Score: High		
3	67340	Session Management - Insufficient Server-Side Session Termination | Score: Medium		
4	67344	Outdated Software Components | Score: Medium		
5	67347	Session Management - Session Token in URL | Score: Medium		
6	67348	Cross-Site Scripting (Stored) | Score: Medium		
7	67351	Session Management - Insufficient Session Expiration | Score: Medium		
8	67353	Insecure Direct Object Reference | Score: Medium		
9	67355	Lack of Access Authorization | Score: Medium		
10	67356	Bypass Client-side Protection Mechanism | Score: Medium		
11	67331	Information Disclosure - Server Version Headers | Score: Low		
12	67338	Insufficient Origin Validation | Score: Low		
13	67339	Missing HTTP Security Response Headers | Score: Low		

14	67341	HttpOnly Cookie Flag Not Set | Score: Low
15	67342	Cookie Attribute - SameSite Attribute Missing or Misconfigured | Score: Low
16	67343	Username Enumeration (Error Responses) | Score: Low
17	67345	Weak Password Policy | Score: Low
18	67350	Session Management - Allows Concurrent Sessions | Score: Low
19	67352	Frameable HTTP Response | Score: Low
20	67357	Verbose Errors | Score: Low
21	67349	Unrestricted File Upload | Score: Informational

Assessment Information

Start Date: 07/21/2025

End Date: 07/25/2025

Name: India Distributor Management System

Testing Team: Cosme, Hugo - EXFIL Security </br>

Scope:

Web App

URL	Host Name	IP Address	Test Environment	In Scope
https://mobil-uat.bizgaze.app		172.105.61.152	Acceptance	True

Tester Comments:
The web application worked as expected.

Executive Summary

Application security testing focused on the India Distributor Management System. The web application facilitates distributor sales tracking, market analysis and related tools. Testing was conducted from July 21th, 2025 to July 25th, 2025 in an acceptance environment. The tester was provided with an administrator, PID and distributor roles. Testing followed best practice methodologies provided by industry-leading security organizations including OWASP and SANS. Overall, the application revealed two high-risk findings related to privilege escalation through the password change mechanism and the use of HTTP Basic Authentication. There were multiple medium-risk findings. There were also several low-risk and informational findings related to industry best practices.

Vulnerability Scores

The Vulnerability Testing Team defines Vulnerability Score as the difficulty of a vulnerability being exploited and the resulting technical impact to the computing environment. The scores are divided into the following categories:

Critical: *Gives attackers a foothold on our network or results in direct financial loss.*

High: *Provides a foothold on a system or network or exposes sensitive data.*

Medium: *Assists attackers in gaining a foothold on a system with minimal privileges or discloses sensitive information.*

Low: *Provides information about a system or does not follow a security best practice.*

Informational: *Suggestions of additional technical best practices or future considerations to follow.*

Additional information on Vulnerability Scores can be located here

1. Weak Password Change Functionality | Score: High

Description:

For any application that requires the user to authenticate with a password, there must be a mechanism by which the user reset their password at any time while authenticated on the application. This would prevent unauthorized access to the account if there is a suspicion that the account has been compromised. More information can be found at: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/09-Testing_for_Weak_Password_Change_or_Reset_Functionalities

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The One-Time Password used in the password change process is not securely tied to the user session, allowing it to be reused to reset passwords for other accounts.

The screenshots below show the OTP being sent to the email.

Request

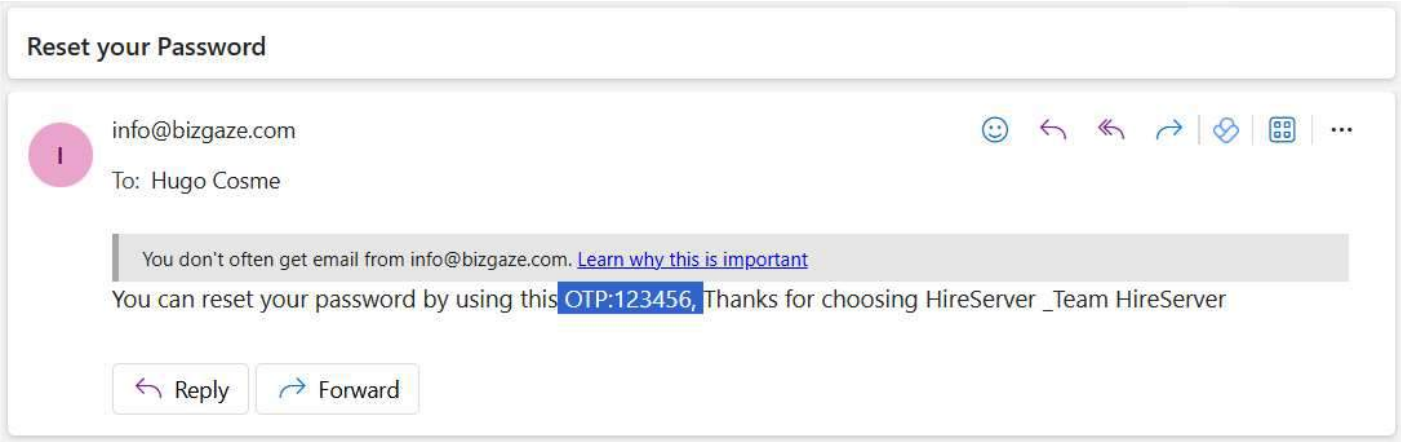
Pretty Raw Hex

```
POST /account/sendotp HTTP/1.1
Host: mobil-uat.bizgaze.app
Cookie: _idty0=; _idty1=; _cnames=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 250
Referer: https://mobil-uat.bizgaze.app/index.html
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "FirstName":"",
  "LastName":"",
  "ContactNumber":"",
  "Email":"hugo.cosme@exfilsecurity.com",
  "TenantName":"",
  "ContactOrEmail":"hugo.cosme@exfilsecurity.com",
  "IsSignup":false,
  "IsRegisterUser":false,
  "IsForgotPswd":true,
  "UnibaseId":"",
  "OtpId":0,
  "UserOtp":""
}
```

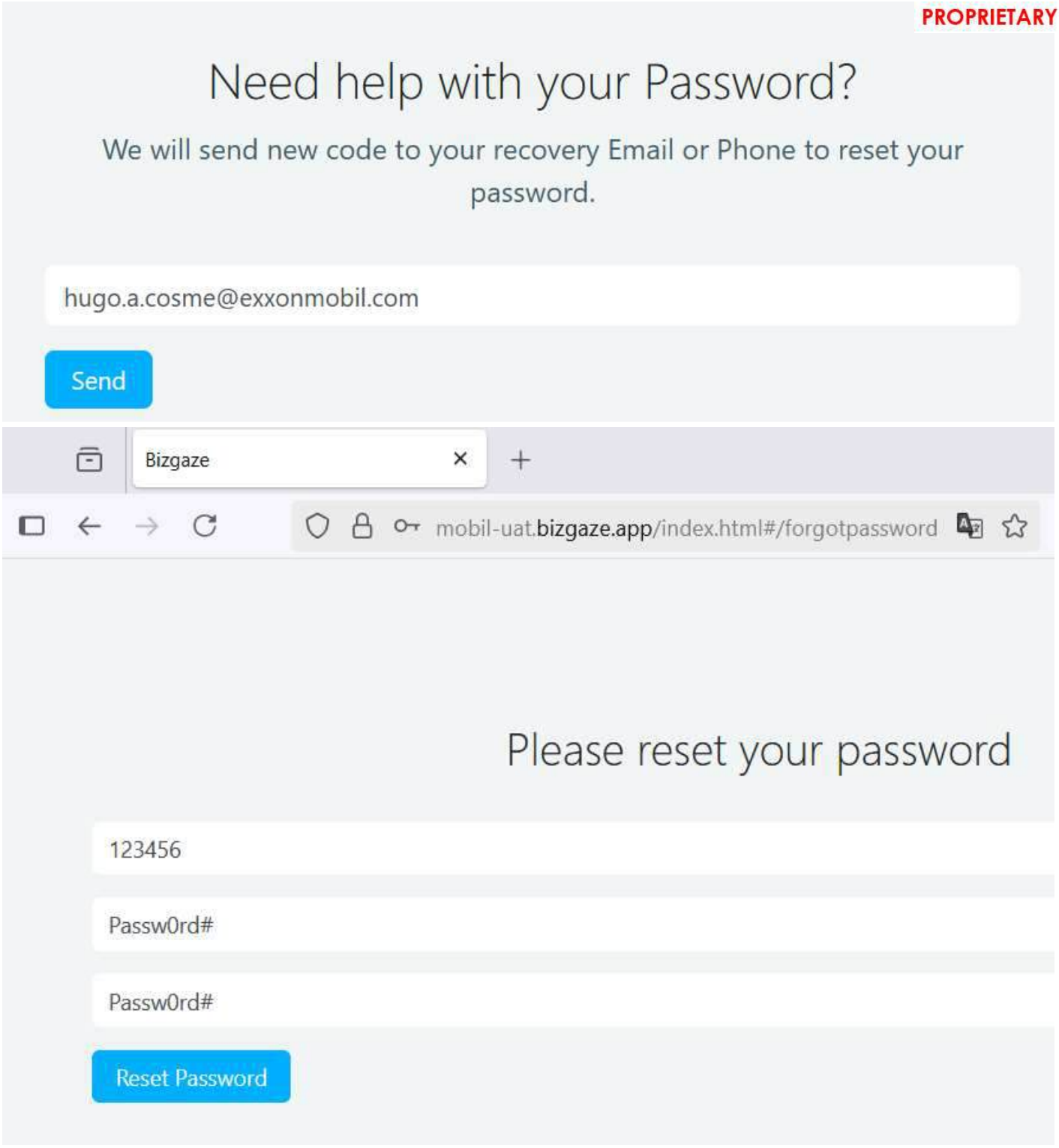
Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 16:51:52 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 121

{
  "result":6115,
  "code":"0",
  "message":"OTP Sent Successfully",
  "serviceName":null,
  "status":0,
  "errors":null,
  "totalRecords":0
}
```



Reusing the same OTP allows changing the password from the administrator account (obtained through the IDOR finding).



The password is successfully modified and administrator access is granted.

Request

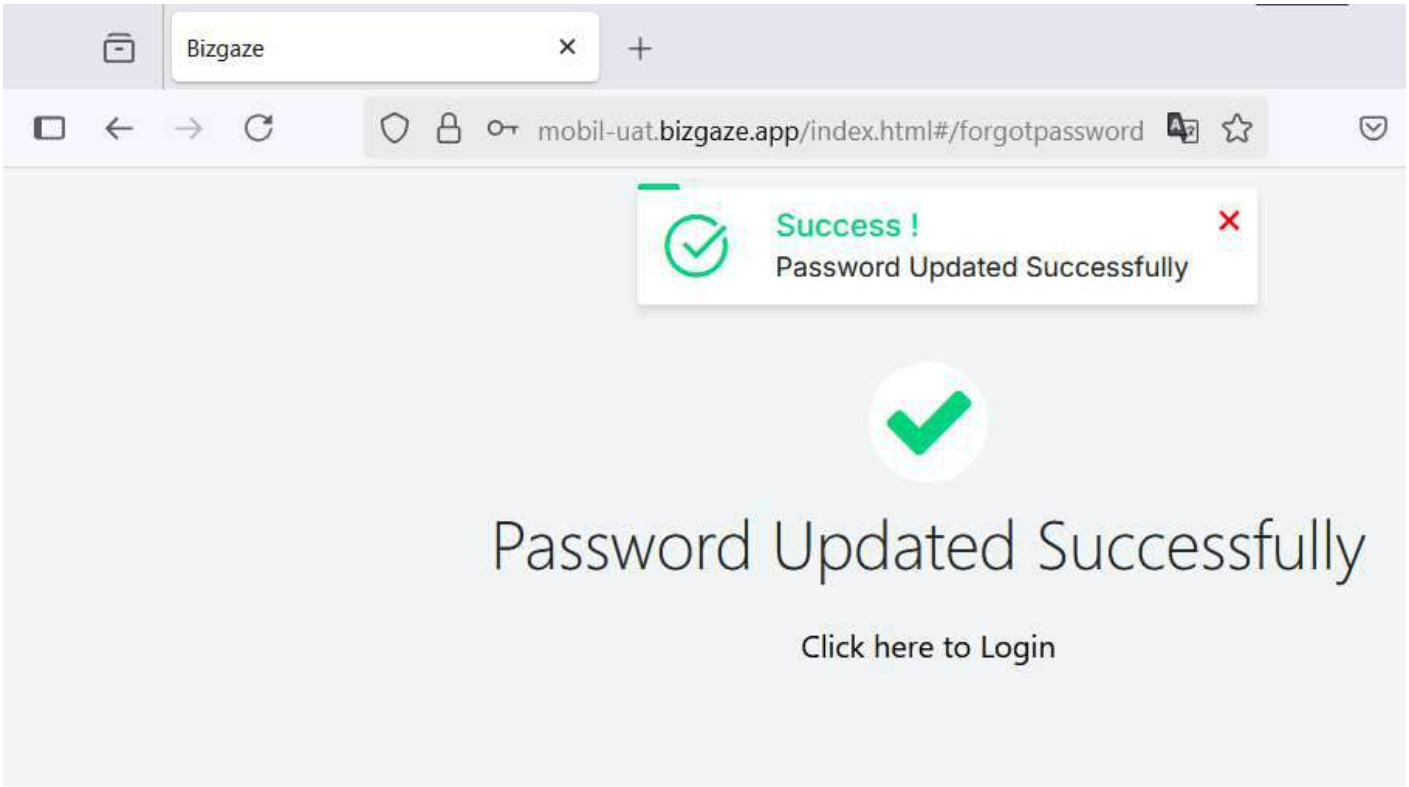
Pretty Raw Hex

```
POST /account/updatepassword HTTP/1.1
Host: mobil-uat.bizgaze.app
Cookie: _idty0=; _idtyl=; _cnames=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 89
Referer: https://mobil-uat.bizgaze.app/index.html
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "Password":"Passw0rd#",
  "OtpId":6115,
  "UserOtp":"123456",
  "OldPassword":"",
  "IsReset":false
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 16:53:24 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 129

{
  "result":null,
  "code":"0",
  "message":"Password Updated Successfully",
  "serviceName":null,
  "status":0,
  "errors":null,
  "totalRecords":0
}
```



Implication:

An attacker with access to an active session could change the password of the logged in user. This would allow them to then access the application within the context of that user.

Recommendation:

Once the user has proved their identity (either through a password reset link, a recovery code, or by logging in on the application) they should be able to change their password.

If a user has been provided with a temporary password after user account creation, the application should require the user to change the password upon initial authentication.

While currently authenticated to the application, the password change functionality should require the old password of the user, and a two-factor authentication mechanism.

More information can be found at:
https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#change-password-feature

Additional Tester Input:

Reference:

WSTG-ATHN-09: Testing for Weak Password Change or Reset Functionalities
Defect ID: 67354

2. HTTP Basic Authentication | Score: High

Description:

HTTP Basic Authentication has a variety of technical and usability concerns that lend itself to an increased likelihood of loss of confidentiality to user credentials. Web applications that deploy HTTP Basic Authentication to authenticate and establish user browser sessions have the following issues:* Lack of Discernibility: The UI is not developer customizable (CSS, JavaScript, Fonts, etc.) to ensure a common look and feel across browsers. There is no guiding standardization to the appearance of the Logon Prompt across browser vendors and versions. There exists no browser location bar clearly indicating the domain that originates the authentication challenge. As a result, browser vendors have failed to help clearly identify the origin of the authentication request. This is problematic in a browser thin client application.* Lack of Supplemental Authentication Attempt Throttling Capability: Developers cannot embed CAPTCHAs, One-Time-Passwords, and other authentication throttling mechanisms to prevent Account Lockout Denial of Service attacks, Credential Bruteforcing, etc.* Lack of Session Token: An established socket is the sole evidence of authentication status. No session cookies are required for subsequent requests for resources upon authentication. As such, is more vulnerable to impersonation attacks.* Credential Leakage: While RFC3986 deprecates the capability due to , browsers still support HTTP Basic Authentication URLs. That is, http://userid:password@domain.com where userid:password are the credentials used for Basic Authentication. The likelihood of credential leakage is increased when users bookmark websites which are stored in their "favorites" unencrypted. Also problematic is the treatment of HTTP Basic Authentication challenges originating from cross-domain static content. For instance, applications that have embedded images to p0wn.com could be vulnerable to credential harvesting if the domain was compromised and HTTP Basic Authentication was enabled. User's may unwittingly provide their credentials to attackers in such a case. Therefore, it's imperative that this susceptibility to provide credentials to dubious HTTP Basic Authentication prompts is removed from the environment by not deploying such authentication mechanisms for web browser sessions.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The application was found to be using Basic Authentication with a static UUID. This method exposes the token throughout the application, increasing the risk of leakage and misuse.

Implication:

HTTP Basic Authentication lacks the qualities of strong server identification and offers opportunities for credential leakage through browser bookmarking. Applications that continue to utilize this antiquated authentication mechanism increase the likelihood of users leaking and exposing their credentials to attackers. If the credentials are used to access other network resources, then the organization risks are increased beyond the system data exposure.

Recommendation:

Implement SSO authentication or login web form for user authentication. Further, when doing so, implement best practices for secure session management, session token generation, and expiration. HTTP Basic Authentication is deprecated for web browser user sessions, as such, the use of other security controls such as TLS does not exempt its continued use.

Additional Tester Input:

Reference:

CWE: 312, 319
Defect ID: 67424

3. Session Management - Insufficient Server-Side Session Termination | Score: Medium

Description:

It is still possible to interact with the web application and/or its API even after the user has already logged out.

Instance:

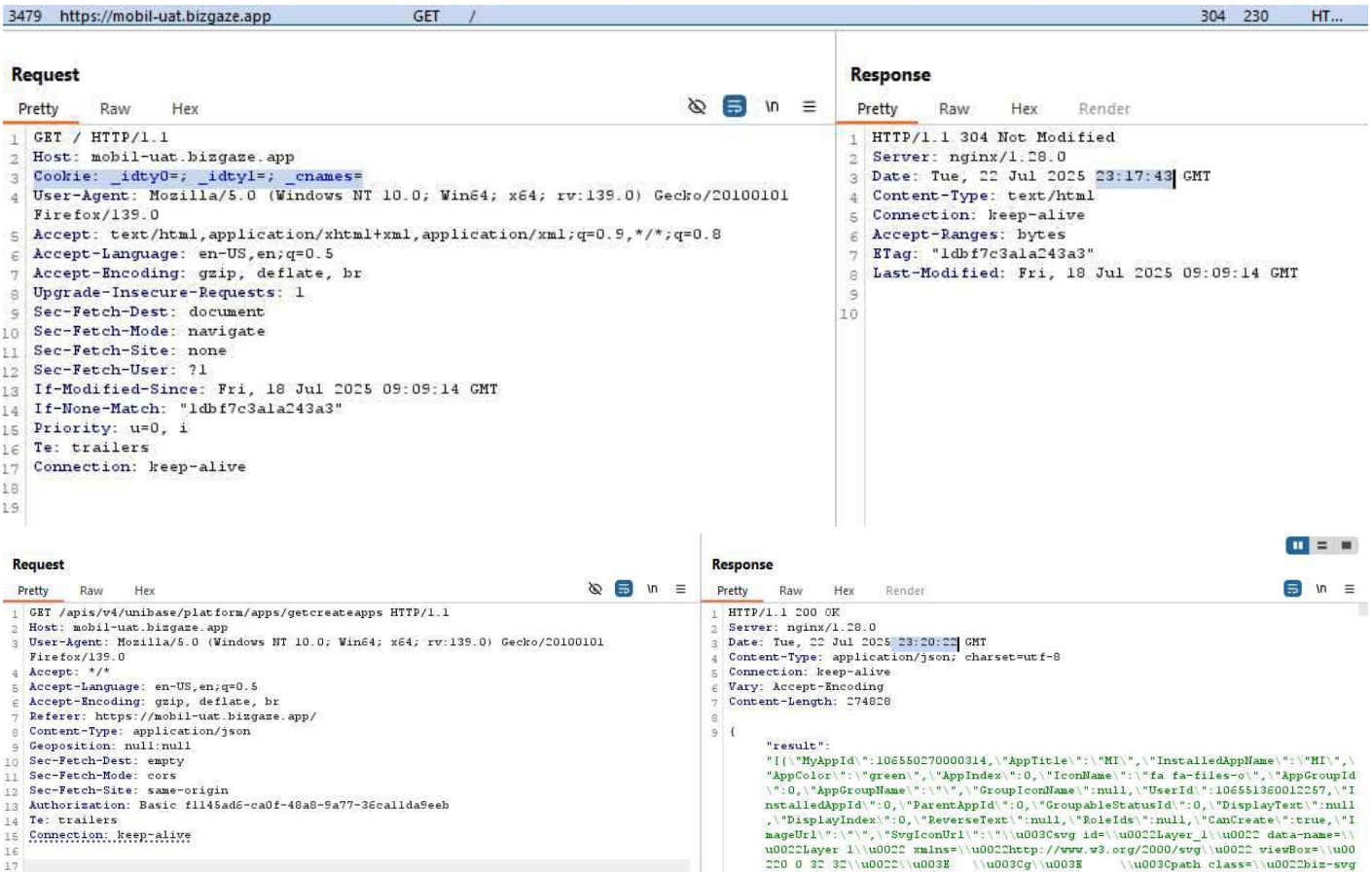
URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application fails to appropriately invalidate the user session upon logout, allowing replay of requests after logging out.

Valid request:

After submitting a logout request, the previous authorization token remains valid:



Implication:

An attacker may be able to execute privilege commands and interact with the web application or API even after the user has been logged out leading to privilege escalation, data manipulation or sensitive information disclosure.

Recommendation:

We recommend expiring the Bearer token issued to the previously logged-in user once the logout process has been initiated. This ensures that any user who might have obtained the Authorization Bearer token cannot be reused even after logging out of the web application.

Additional Tester Input:

Reference:

N/A
Defect ID: 67340

4. Outdated Software Components | Score: Medium

Description:

In assessing the application, outdated software components were found to be in use.

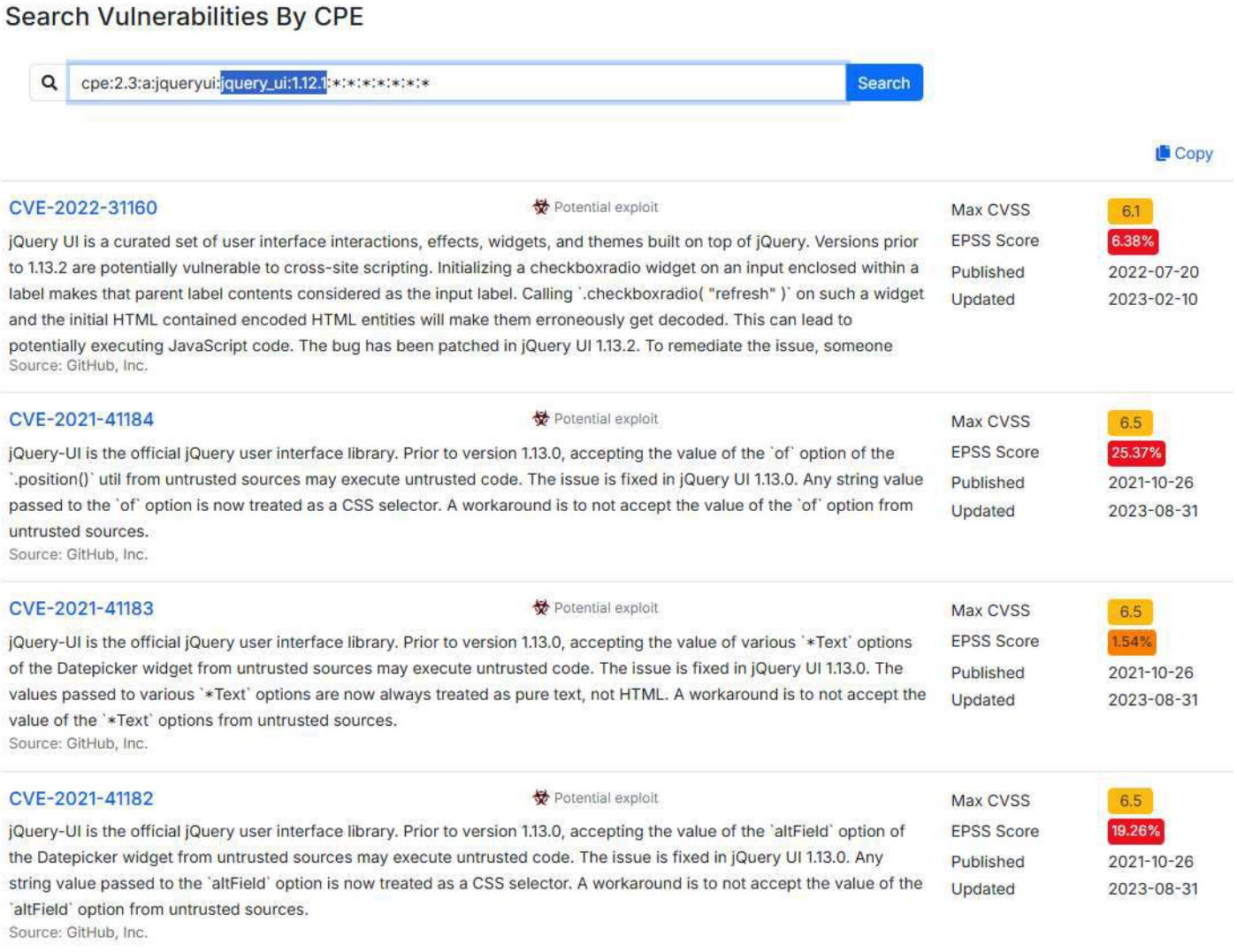
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The application uses an outdated version of JQuery UI. This version has known security vulnerabilities that may be leveraged in an attack against the application.

JQuery UI - v1.12.1



Implication:

Operating insecure and outdated versions of software components leave the affected host vulnerable to attacks to a potential threat using new vulnerabilities. Successful exploitation of some of these vulnerabilities would lead to a compromise of the system, applications, the data that is collected, and the administrative accounts that control it.

Recommendation:

Ensure missing security updates are installed on these hosts immediately. Update any older software components in use to the most recent available version to limit any security exposures. If the product has reached end of life or has been discontinued, consider migrating to an alternative technology that is supported.

Additional Tester Input:

5. Session Management - Session Token in URL | Score: Medium

Description:

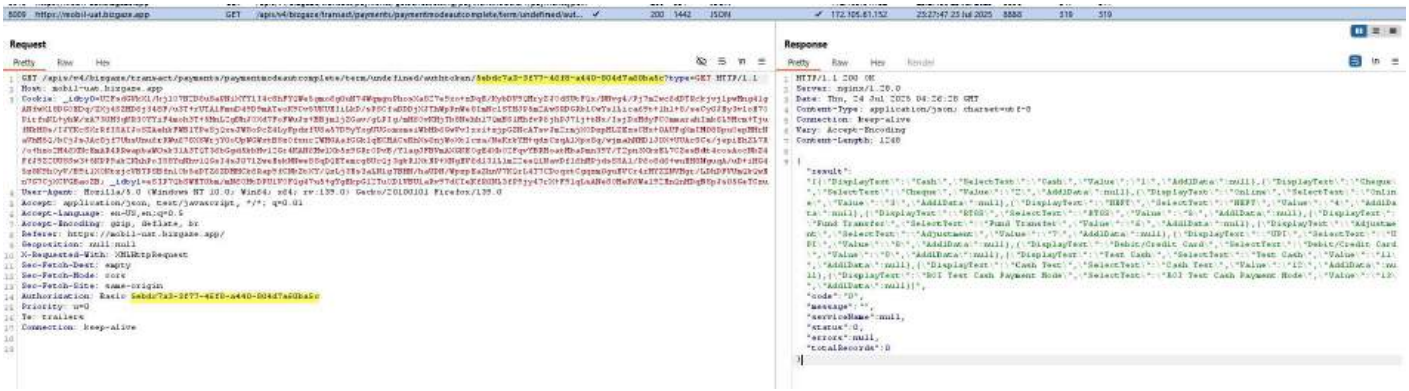
It was discovered that session tokens, which are used to maintain state, are included within the URL of each request

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The authorization token is passed in the URL of the following endpoint. URLs are at risk of being disclosed as they can be stored in browser history, server logs, proxy devices, etc.



For instance, the token was captured due to improper cache configuration.

Implication:

Sensitive information within URLs may be logged in various locations, including the user's browser, the web server, and any forward or reverse proxy servers between the two endpoints. URLs may also be displayed on-screen, bookmarked or emailed around by users. They may be disclosed to third parties via the Referer header when any off-site links are followed. Placing session tokens into the URL increases the risk that they will be captured by an attacker.

Recommendation:

The application should use an alternative mechanism for transmitting session tokens, such as HTTP cookies for browsers or headers for thick clients. When using cookies for session tokens ensure the cookies are protected with the HttpOnly flag to prevent JavaScript access to the cookie. If the entire browser session occurs over TLS the Secure cookie flag prevents leakage of the sensitive cookie.

https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

Additional Tester Input:

Reference:

CWE: 384
Defect ID: 67347

6. Cross-Site Scripting (Stored) | Score: Medium

Description:

Cross-Site Scripting (XSS) attacks occur when data enters a Web application through an untrusted source, most frequently a web request, and that data is included in dynamic content that is sent to a web user without being validated for malicious code.

Stored Cross-Site Scripting occurs when the malicious payload is stored and retained on the application server (generally a database) and retrieved and delivered to subsequent users who visit the affected page. As a result, Stored XSS are often used as worms to propagate between different users.

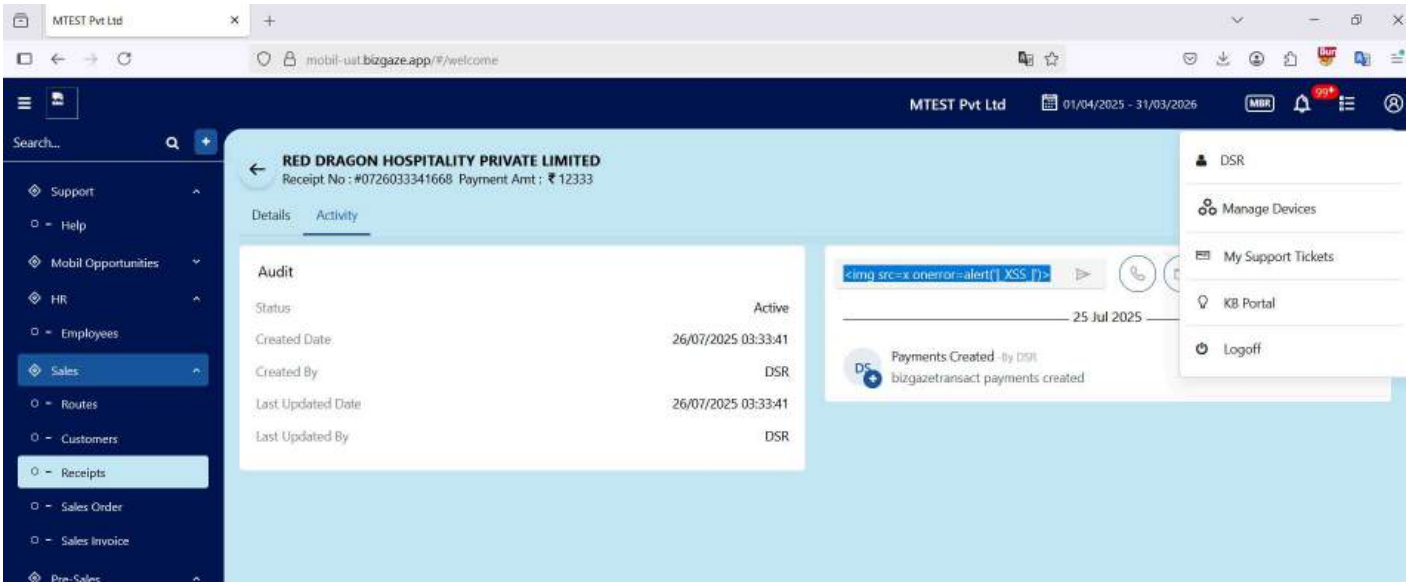
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

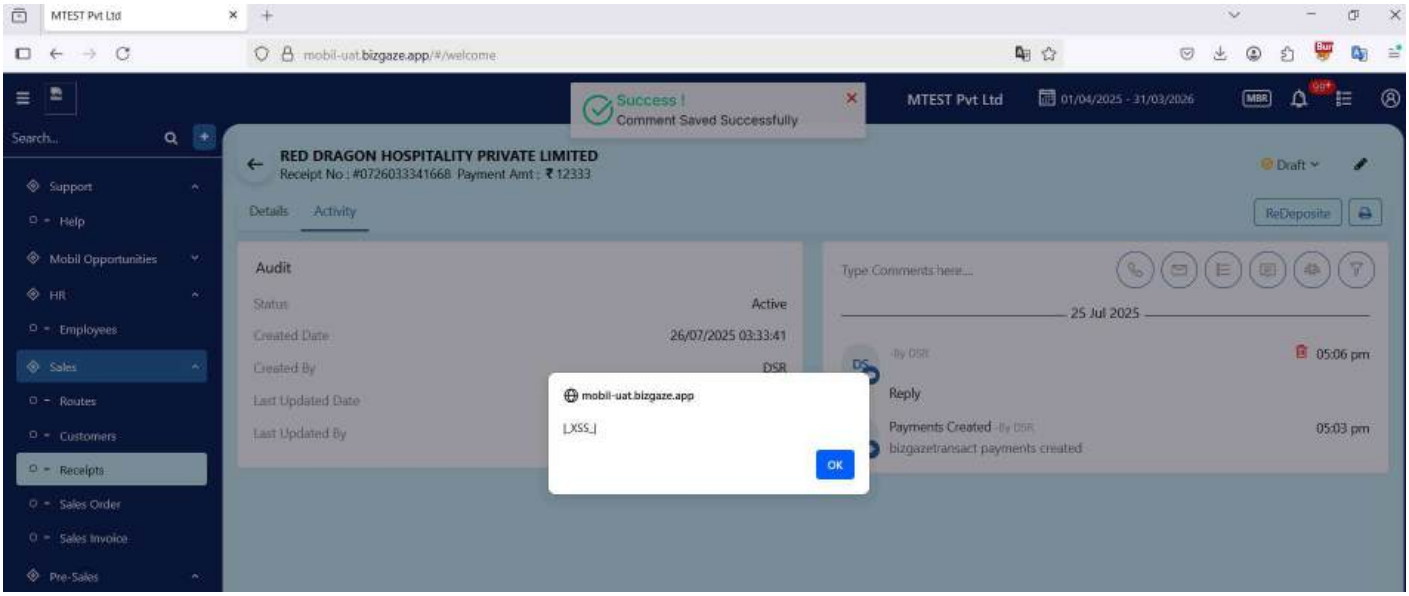
Details:

The web application is vulnerable to Stored Cross-Site Scripting due to the lack of input sanitization throughout the application. This can be misused in order to add malicious javascript that is triggered when the resources are opened by users.

For instance, the comments field is used to insert the javascript code. This is possible with all tested roles.



The XSS is triggered when a user accesses the resource.



Implication:

When exploited, XSS vulnerabilities have led to the harvesting of Personally Identifiable Information (PII) including user credentials, data exfiltration, Distributed Denial of Service attacks, bypassing some anti-CSRF measures, network probe of private internet address space, and malware distribution through exploitation of browser and browser plugin vulnerabilities. The browser itself can be further compromised, falling under complete control of the attacker. Several ready-made exploit frameworks, such as BeEF or XSSF exist for this purpose, allowing the attacker to abuse the browser in a number of ways including delivering browser-specific exploits, logging keystrokes, viewing browser history, or using the browser as a proxy to attack systems inside the corporate firewall.

An unknowing user can have their data compromised from a vulnerable site by merely browsing an attacker's web site that embeds the vulnerable site in a hidden iframe, or by clicking a malicious hyperlink such as from an email.

Recommendation:

Recommendations include implementing secure programming techniques that ensure proper filtration of user-supplied data, and encoding all user supplied data to prevent inserted scripts being sent to end users in a format that can be executed. Cross-Site Scripting attacks can be avoided by carefully validating all input and properly encoding all output. When validating user input, verify that it matches the strictest definition of valid input possible. For example, if a certain parameter is supposed to be a number, attempt to convert it to a numeric data type in your programming language. For validating string patterns uses Regexes to validate the string pattern match and enforce whitelist rules.

In some instances, the application must allow the user to supply free-form data using a large range of printable characters. In such cases, ensure that control characters are properly encoded to neutralize the potential for script injection. The HTML Document contains three contexts that provide an opportunity for script execution, that is, CSS, Script, and HTML Body. Each context has its own set of associated control characters, that when encountered, force the browser into JavaScript tokenization and execution. Consider which context the user's input will be echoed back in the response and encode the output for that context.

For a complete reference refer to:
https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

OWASP ESAPI Java Encoder Framework:
https://www.owasp.org/index.php/OWASP_Java_Encoder_Project

Additional Tester Input:

Reference:

CWE: 79, 80, 82, 83, 87, 116, 692, 811
Defect ID: 67348

7. Session Management - Insufficient Session Expiration | Score: Medium

Description:

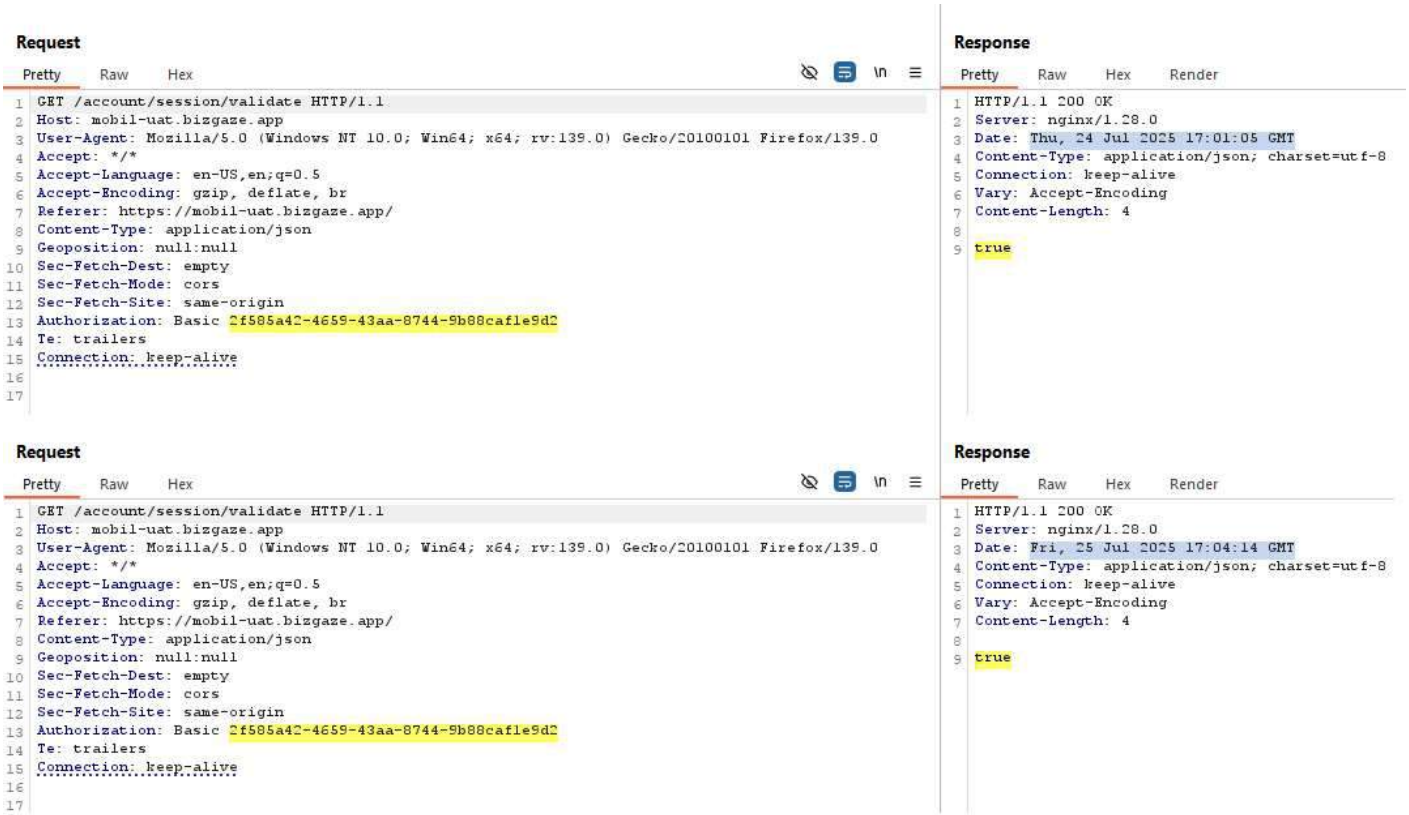
Insufficient Session Expiration occurs when a Web application permits an attacker to reuse old session credentials or session IDs for authorization. Insufficient Session Expiration increases a Web site's exposure to attacks that steal or reuse user's session identifiers.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The session token did not expire after more than one day. This increases the window of opportunity for an attacker to gain access to the token and the session.



Implication:

Session expiration is comprised of two timeout types: inactivity and absolute. An absolute timeout is defined by the total amount of time a session can be valid without re-authentication and an inactivity timeout is the amount of idle time allowed before the session is invalidated. The lack of proper session expiration may increase the likelihood of success of certain attacks. A long expiration time increases an attacker's chance of successfully guessing a valid session ID. The longer the expiration time, the more concurrent open sessions will exist at any given time. The larger the pool of sessions, the more likely it will be for an attacker to guess one at random. Although a short session inactivity timeout does not help if a token is immediately used, the short timeout helps to ensure that the token is harder to capture while it is still valid.

Recommendation:

The web application should invalidate a session after a predefined idle time has passed (a timeout).

Additional Tester Input:

Reference:

CWE: 613
Defect ID: 67351

8. Insecure Direct Object Reference | Score: Medium

Description:

IDOR occurs when exposes a reference to an internal implementation object to a user-supplied input without sufficient validation and direct access to the object requested is provided.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application allows user identifiers to be modified in requests without proper access control. This may expose sensitive data from other users, including privileged accounts.

The distributor`s authorization token and user ID are shown below.

Request

```
POST /Account/ValidateAndLogin HTTP/1.1
Host: mobil-uat.bizgaze.app
Cookie: _idty0=; _idtyl=; _cnames=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 94
Referer: https://mobil-uat.bizgaze.app/
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "UserName":"hacosmev@exxonmobil.com",
  "Password":"Password",
  "UnibaseId":"",
  "RememberMe":false
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 20:24:35 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 3252

{
  "status":1,
  "sessions":null,
  "currentSession":{
    "name":"DSR",
    "userId":106551360011665,
    "tenantId":503,
    "parentUserId":0,
    "languageId":0,
    "dateFormat":null,
    "timeZone":0,
    "sourceId":0,
    "unibaseId":"1744198fd7f74f3ab467251db4a10c33",
    "sessionId":"33527ef0-7d81-45fe-b106-46191676f2c4",
    "isAdmin":false,
    "latd":null,
    "lgId":null,
    "secDataEnabled":false,
    "ipAddress":null,
    "deviceId":105030230000003,
    "isRestrictedUser":false,
    "photoUrl":null,
    "appUrl":null,
    "partitionId":0,
    "partitionFromDate":"0001-01-01T00:00:00",
```

For instance, the following endpoint is accessed using the previous IDs, providing access to the distributor information.

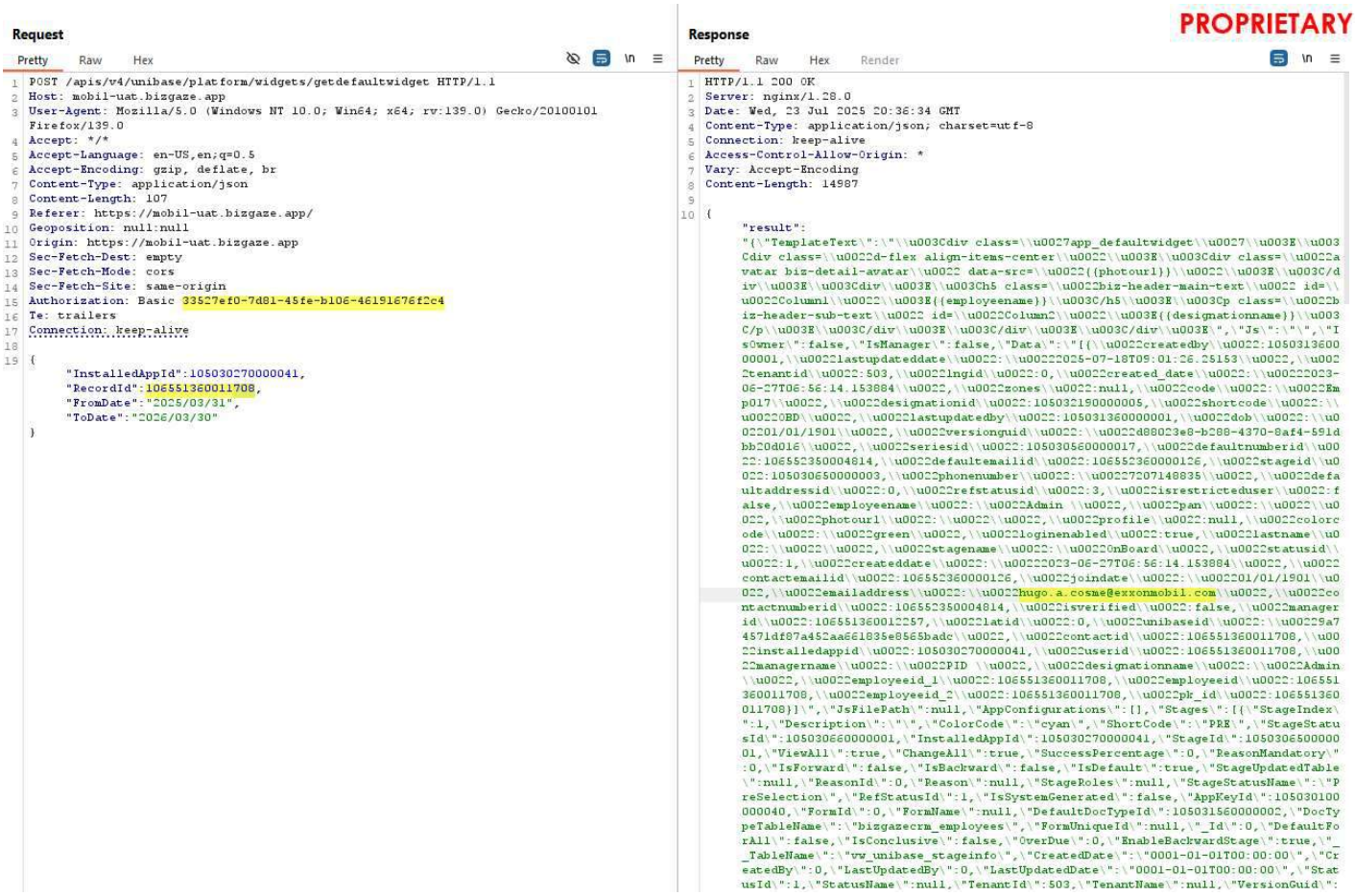
Request

```
POST /apis/v4/unibase/platform/widgets/getdefaultwidget HTTP/1.1
Host: mobil-uat.bizgaze.app
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 107
Referer: https://mobil-uat.bizgaze.app/
Geoposition: null:null
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Authorization: Basic 33527ef0-7d81-45fe-b106-46191676f2c4
Te: trailers
Connection: keep-alive

{
    "InstalledAppId":105030270000041,
    "RecordId":106551360011665,
    "FromDate":"2025/03/31",
    "ToDate":"2026/03/30"
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 20:34:47 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 14997

{
    "result":
    "{\"TemplateText\":\"\\u003Cdiv class=\\u0027app_defaultwidget\\u0027\\u003E\\u003
    Cdiv class=\\u0022d-flex align-items-center\\u0022\\u003E\\u003Cdiv class=\\u0022a
    vatar biz-detail-avatar\\u0022 data-src=\\u0022{{photourl}}\\u0022\\u003E\\u003C/d
    iv\\u003E\\u003Cdiv\\u003E\\u003Ch5 class=\\u0022biz-header-main-text\\u0022 id=\\
    u0022Column1\\u0022\\u003E{{employeename}}\\u003C/h5\\u003E\\u003Cp class=\\u0022b
    iz-header-sub-text\\u0022 id=\\u0022Column2\\u0022\\u003E{{designationname}}\\u003
    C/p\\u003E\\u003C/div\\u003E\\u003C/div\\u003E\\u003C/div\\u003E\",\"Js\":\"\",\"I
    sOwner\":true,\"IsManager\":false,\"Data\":\"[{\\u0022createdby\\u0022:10503136000
    0001,\\u0022lastupdateddate\\u0022:\\u0022\\u0022025-07-18T09:02:51.4513Z\\u0022,\\u0022
    tenantid\\u0022:503,\\u0022lngid\\u0022:0,\\u0022created_date\\u0022:\\u0022\\u0022023-0
    5-23T09:43:31.865817\\u0022,\\u0022zones\\u0022:null,\\u0022code\\u0022:\\u0022Emp
    009\\u0022,\\u0022designationid\\u0022:105032190000012,\\u0022shortcode\\u0022:\\u
    00220BD\\u0022,\\u0022lastupdatedby\\u0022:105031360000001,\\u0022dob\\u0022:\\u00
    2201/01/1901\\u0022,\\u0022versionguid\\u0022:\\u0022a81ac628-082d-4557-bc1d-0c6f9
    3491916\\u0022,\\u0022seriesid\\u0022:105030560000017,\\u0022defaultnumberid\\u002
    2:106552350004783,\\u0022defaultemailid\\u0022:106552360000113,\\u0022stageid\\u00
    22:105030650000003,\\u0022phonenumber\\u0022:\\u00228179723275\\u0022,\\u0022defau
    ltaddressid\\u0022:0,\\u0022refstatusid\\u0022:3,\\u0022isrestricteduser\\u0022:fa
    lse,\\u0022employeename\\u0022:\\u0022DSR \\u0022,\\u0022pan\\u0022:\\u0022\\u0022
    ,\\u0022photourl\\u0022:\\u0022\\u0022,\\u0022profile\\u0022:null,\\u0022colorcode
    \\u0022:\\u0022green\\u0022,\\u0022loginenabled\\u0022:true,\\u0022lastname\\u0022
    :\\u0022\\u0022,\\u0022stagename\\u0022:\\u0022OnBoard\\u0022,\\u0022statusid\\u00
    22:1,\\u0022createddate\\u0022:\\u0022\\u0022023-05-23T09:43:31.865817\\u0022,\\u0022con
    tactemailid\\u0022:106552360000113,\\u0022joindate\\u0022:\\u0022201/01/1901\\u0022
    ,\\u0022emailaddress\\u0022:\\u0022hacosmev@exxonmobil.com\\u0022,\\u0022contactnu
    mberid\\u0022:106552350004783,\\u0022isverified\\u0022:false,\\u0022managerid\\u00
    22:106551360011708,\\u0022latid\\u0022:0,\\u0022unibaseid\\u0022:\\u0022\\u0022744198fd7
    f74f3ab467251db4a10c33\\u0022,\\u0022contactid\\u0022:106551360011665,\\u0022insta
    lledappid\\u0022:105030270000041,\\u0022userid\\u0022:106551360011665,\\u0022manag
```

Using the same authorization token and modifying the last three digits of the "RecordID" parameter allows access to other users` data, including data from the administrator`s account.



The ID-based access was found to be used in multiple areas of the application.

Implication:

May compromise all the data that can be referenced by the parameter. Unless object references are unpredictable, it's easy for an attacker to access all available data of that type.

Recommendation:

Use per user or session indirect object references. This prevents attackers from directly targeting unauthorized resources. For example, instead of using the resource's database key, a drop down list of six resources authorized for the current user could use the numbers 1 to 6 to indicate which value the user selected. The application has to map the per-user indirect reference back to the actual database key on the server.
Check access. Each use of a direct object reference from an untrusted source must include an access control check to ensure the user is authorized for the requested object.

Additional Tester Input:

Reference:

CWE-639
Defect ID: 67353

9. Lack of Access Authorization | Score: Medium

Description:

While assessing the application, a vulnerability was found relating to the mismanagement of permissions, privileges and other security features that are used to perform access control.

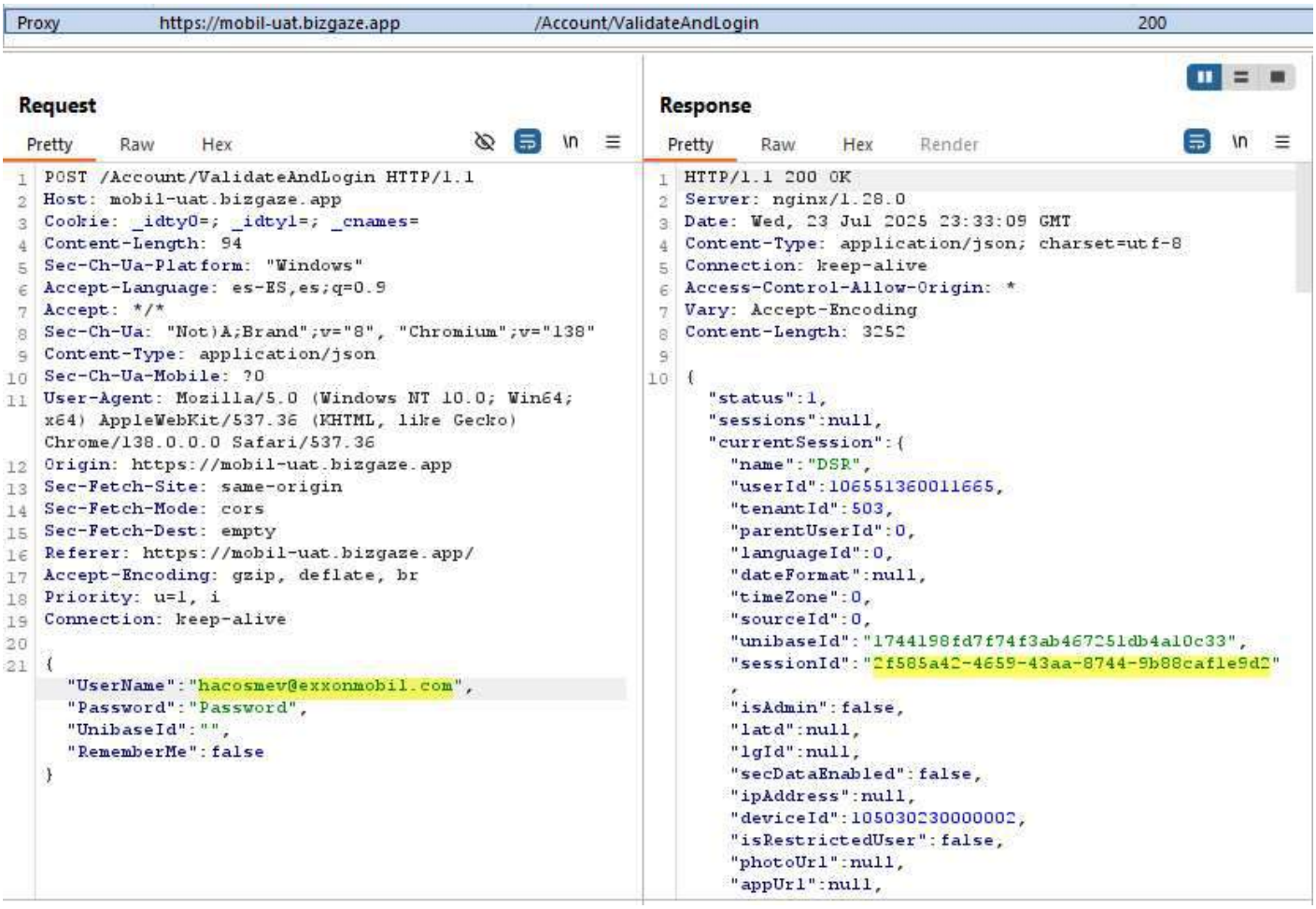
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

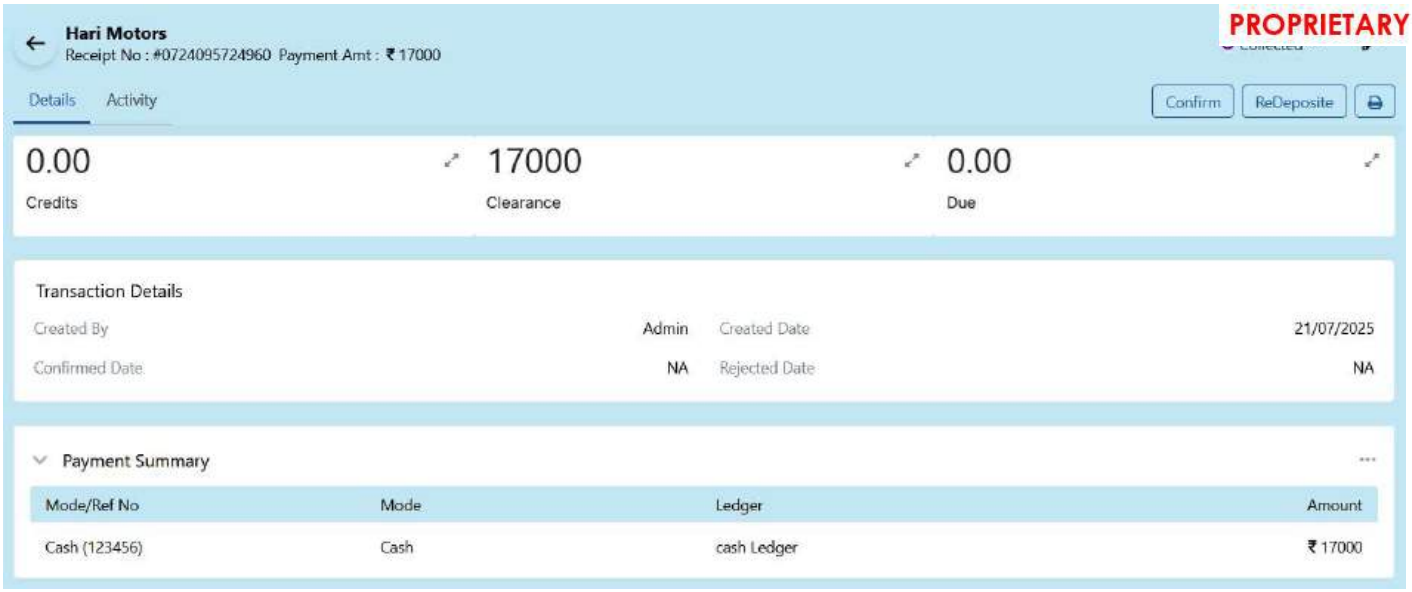
Details:

The web application does not enforce proper access authorization controls, allowing restricted actions with a low-privileged token.

The following screenshot shows the authorization token of a distributor.



For instance, it is possible to modify the amount of a receipt. The following screenshot shows the view of an existing receipt.



Using the distributor token, it is possible to change the total amount adjusting the highlighted parameters.

Request

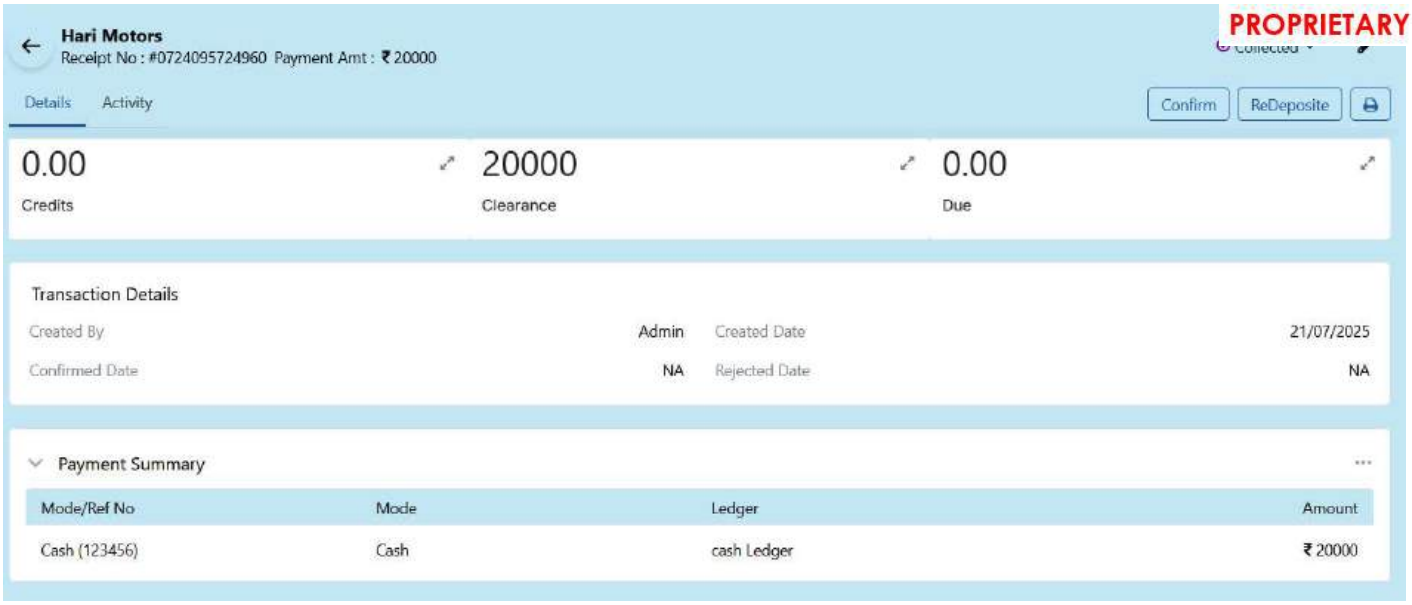
Pretty Raw Hex

```
POST /apis/v4/unibase/platform/forms/savedynform/105031650000307 HTTP/1.1
Host: mobil-uat.bizgaze.app
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101 Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 1804
Referer: https://mobil-uat.bizgaze.app/
Geoposition: null:null
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Authorization: Basic cf585a42-4659-43aa-8744-9b88cafle9d2
Priority: u=0
Te: trailers
Connection: keep-alive

{"lobid_lobname":"0","paymentaccountid_ledgername":"0","currencyid_currencyname":"0",
"contactid_orgcontactname":"0","paymentid":"106553290002390","journalid":"106553290002390","journalno":
"#0724095724960","journaltypeid":"4","decimalpoints":"","refno":"123456","firstledgerid":"106552770000445",
"secondledgerid":"105032770000001","issystemgenerated":"False","journalstatusid":"1","branchid":
"105032230003413","paymenttypeid":"1","tempjournalno":"#0724095724960","advanceamount":"16000",
"paymentstatusid":"1","stageid":"105030650000030","totalamount":"16000","paymentamount":"16000",
"journal_paymentmodeid":"1","refdate":"01/01/1900 00:00:00","seriesid":"105030560000151","paymentmodeid":"1",
"journal_seriesid":"105030560000151","payment_contactid":"106551360011971","payment_orgcontactid":
"106551360011971","remoteid":"0","templateid":"105032850000009","chequeno":"","bankid":"","bankrefno":"",
"bankrefdate":"01/01/1900 00:00:00","bankbranch":"","duedays":"0","journalallocid":"106553320002903",
"alloc_journalid":"106553290002390","code":"#0724095724960","conversionratedifference":"1","operatortype":"*"
,"refid":"0","discountaccountid":"","discountamount":"0","bouncecharges":"0","contactid":"106551360011972",
"journaldate":"2025/07/21","conversionrate":"1","currencyid":"0","PaymentEntries":[{"PaymentModeId":1,"RefNo"
:"123456","LedgerId":105032770000001,"RefDate":"2025-07-22","ChequeNo":"","ChequeDate":"2025-07-24","BankId":
0,"BankBranch":"","PaymentAmount":20000,"JournalEntryId":0,"DiscountAccountId":0,"PenalityAccountId":0,
"TDSAccountId":0,"ForeingExchangeAccount":0}],"paymentaccountid":"105032770000001","lobid":"105032260000001",
"Wallets":[{"JournalAllocId":0,"LedgerId":106551360011972,"Credit":20000,"RefAllocId":0,"AllocType":3,
"AllocTypeId":3}],"Dues":[],"notes":"testing description 01"}
```

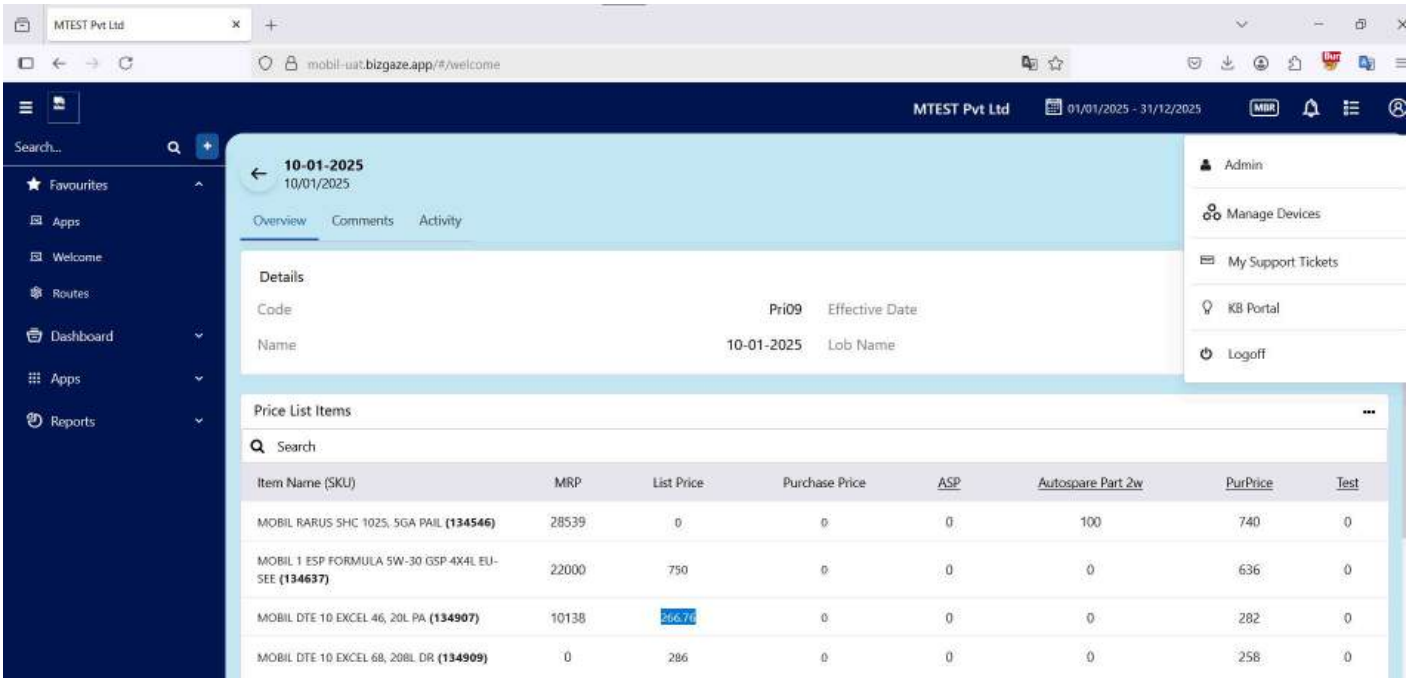
Response

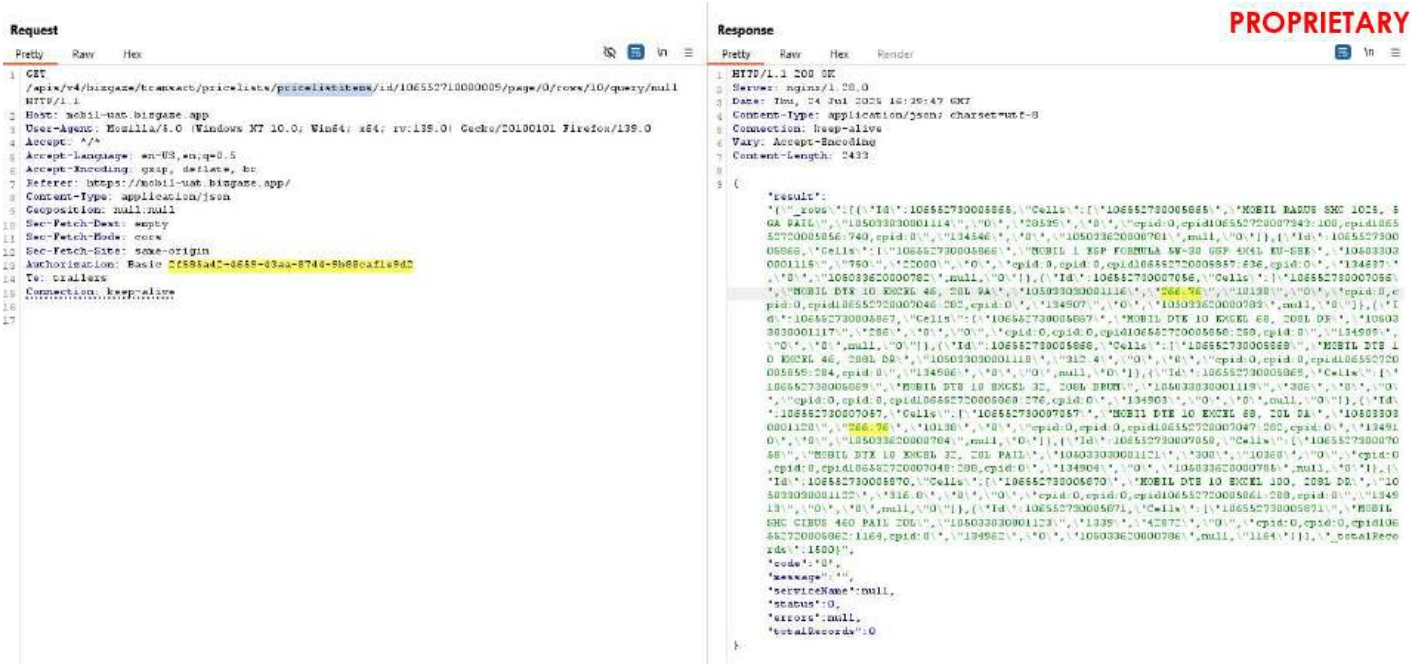
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 24 Jul 2025 05:46:16 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 139

{
    "result":"106553290002390",
    "code":"0",
    "message":"Receipt Saved Successfully",
    "serviceName":null,
    "status":0,
    "errors":null,
    "totalRecords":0
}
```

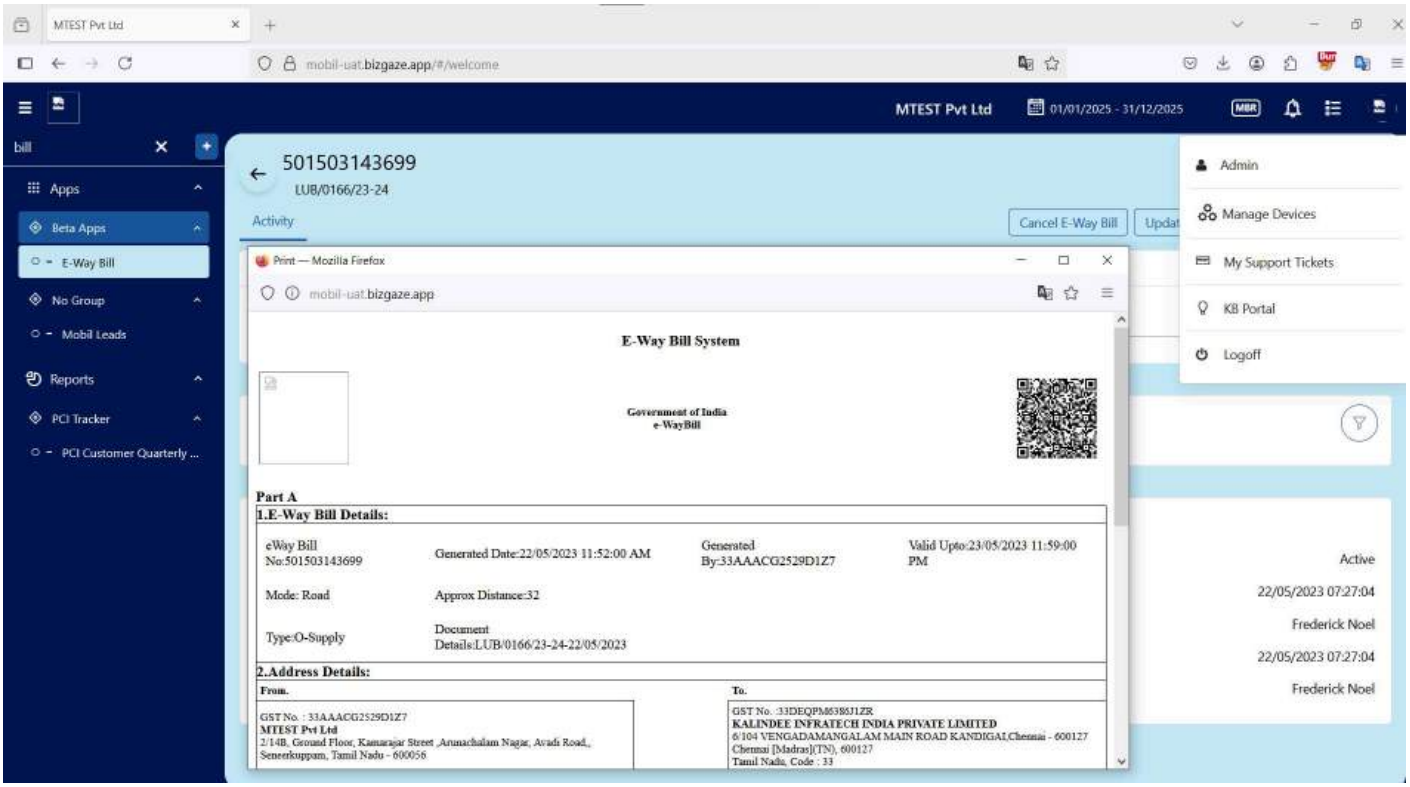


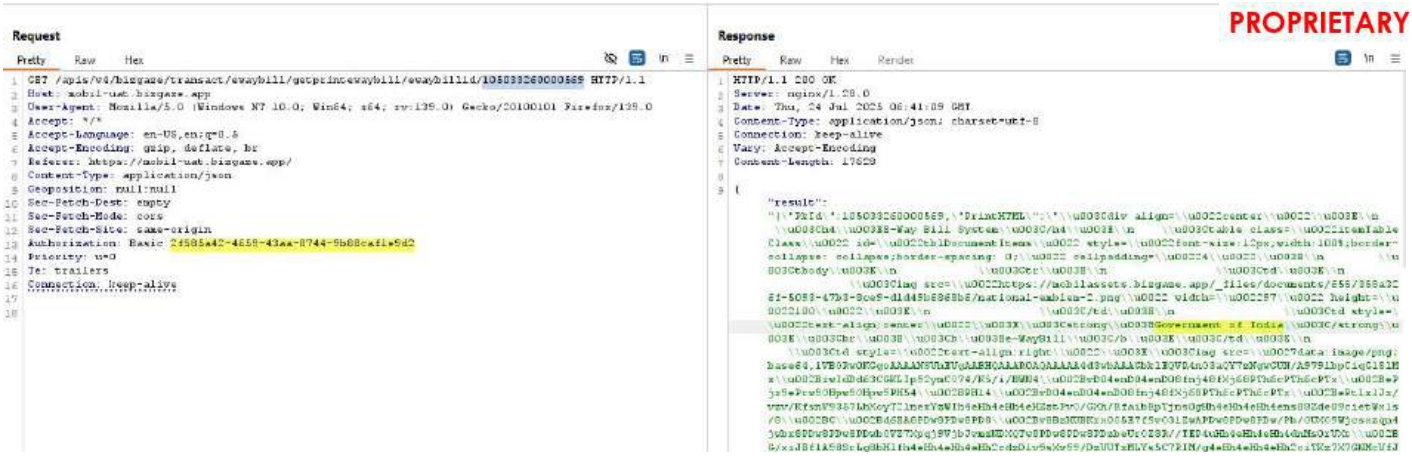
In addition, the price list can be accessed using the distributor`s token.





It is also possible to access document information from administrators in the same way.





This behavior is present throughout the application.

Implication:

If misconfigurations and mismanagement of access controls are present in an application, the risk of violating segregation of duties and allowing unauthorized access to restricted access is increased.

Recommendation:

Follow the principle of least privileges when assigning access rights within the application.

Additional Tester Input:

Reference:

CWE: 264
Defect ID: 67355

10. Bypass Client-side Protection Mechanism | Score: Medium

Description:

When the server relies on protection mechanisms placed on the client side, an attacker can modify the client-side behavior to bypass the protection mechanisms resulting in potentially unexpected interactions between the client and server. The consequences will vary, depending on what the mechanisms are trying to protect.

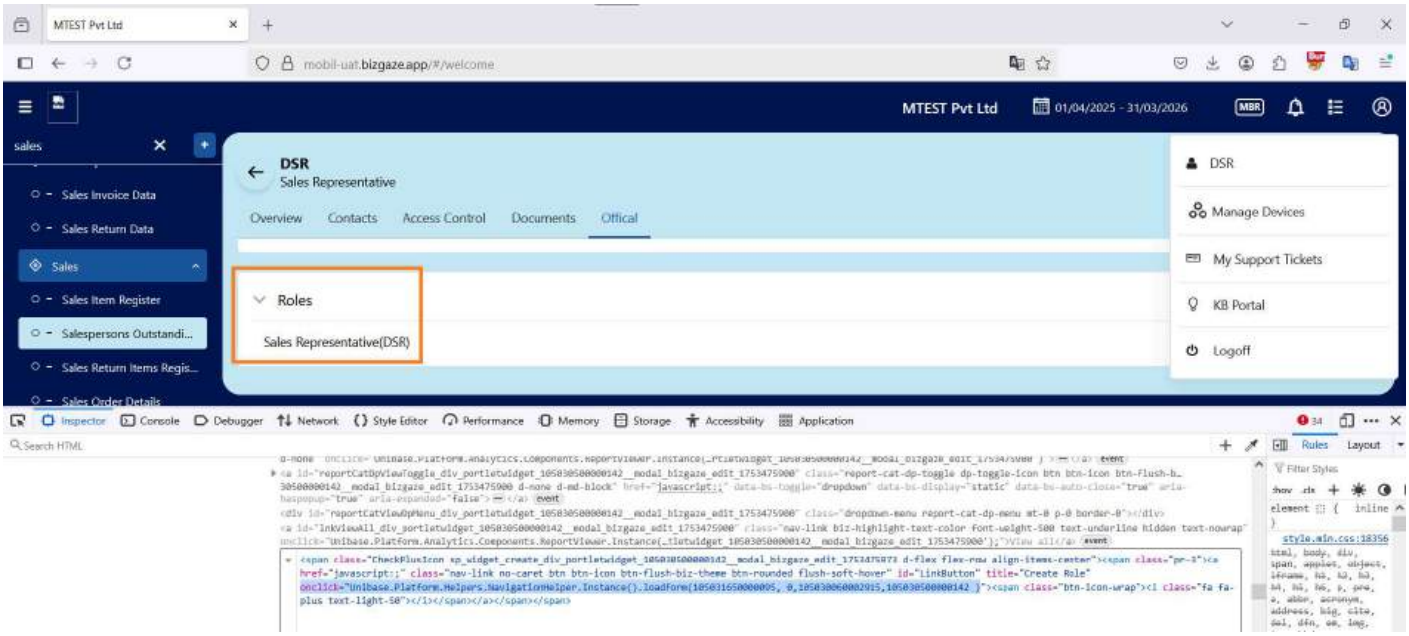
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

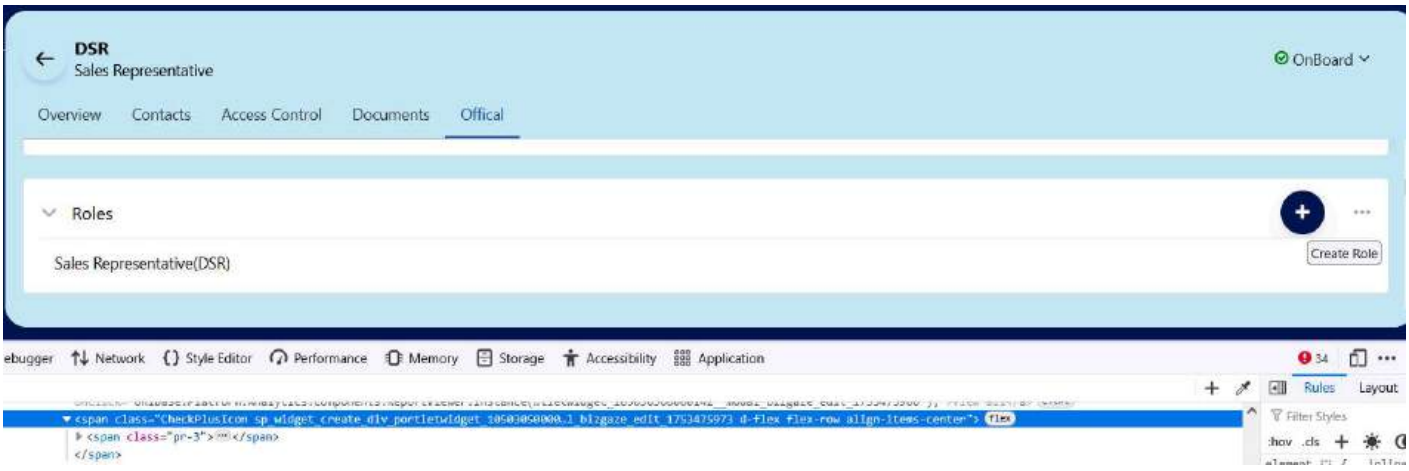
Details:

Distributors are able to bypass client side protections and use javascript handlers to trigger restricted actions. The risk is mitigated by the fact that the form identifiers are not so easy to guess.

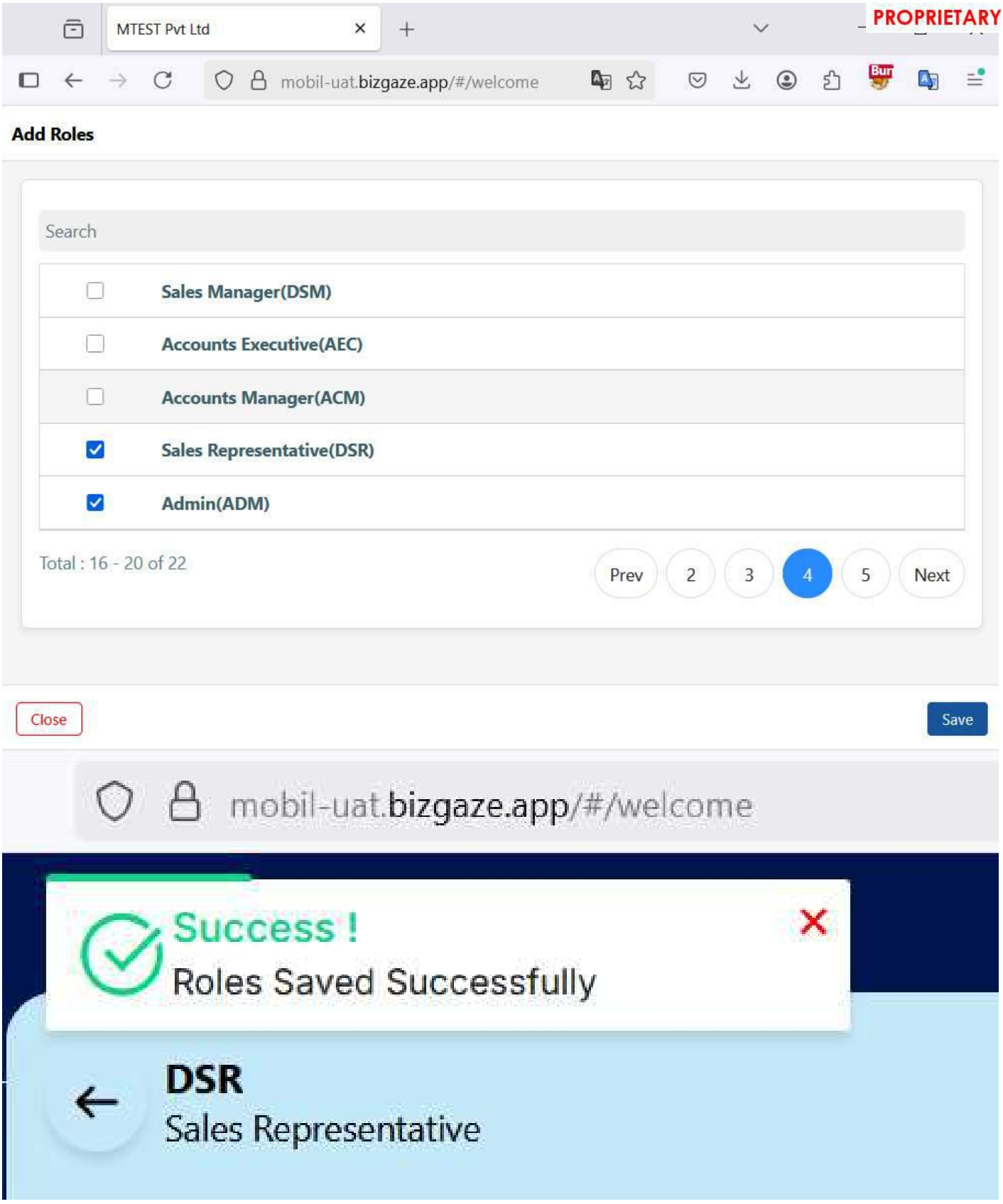
For instance, it is possible to escalate privileges from a distributor account. The original role is shown below.



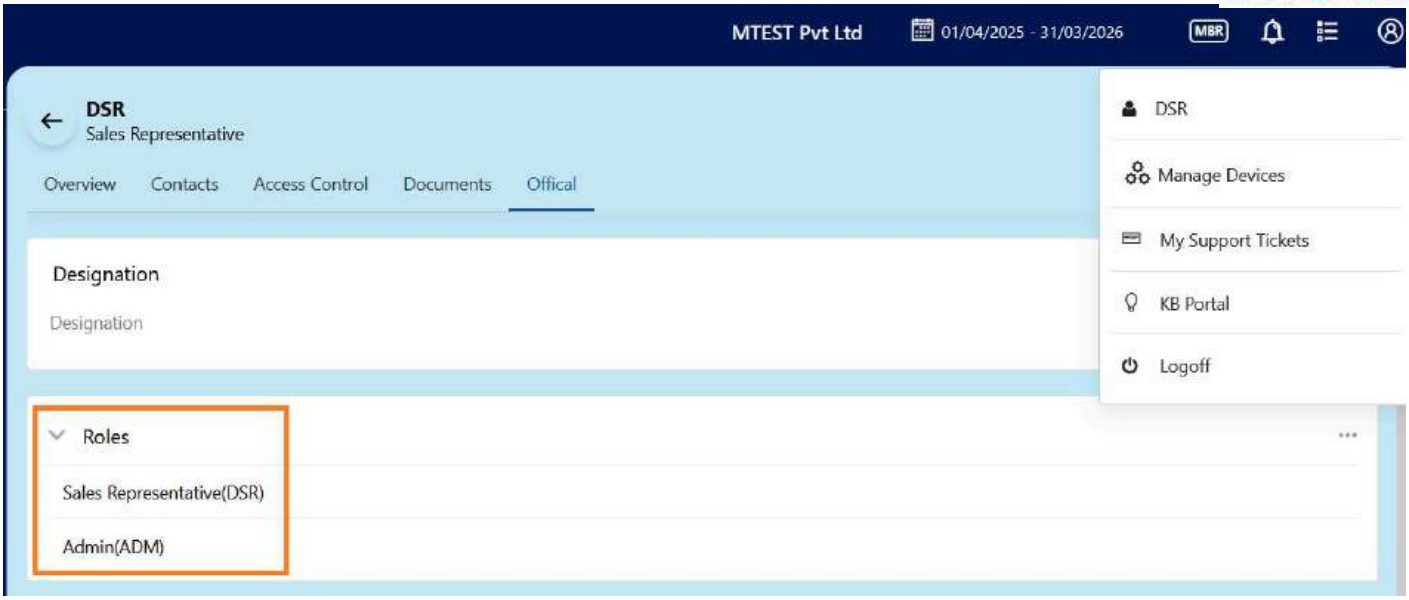
Adding the previous HTML code allows a button to be added for creating a new role.



It is then possible to continue using the functionality.



The administrator role is successfully added, granting elevated privileges.



Implication:

Attacker can intercept the request and modify the value to bypass the client-side control.

Recommendation:

Implement the security controls on the server-side as well.

Additional Tester Input:

Reference:

CWE-602
Defect ID: 67356

11. Information Disclosure - Server Version Headers | Score: Low

Description:

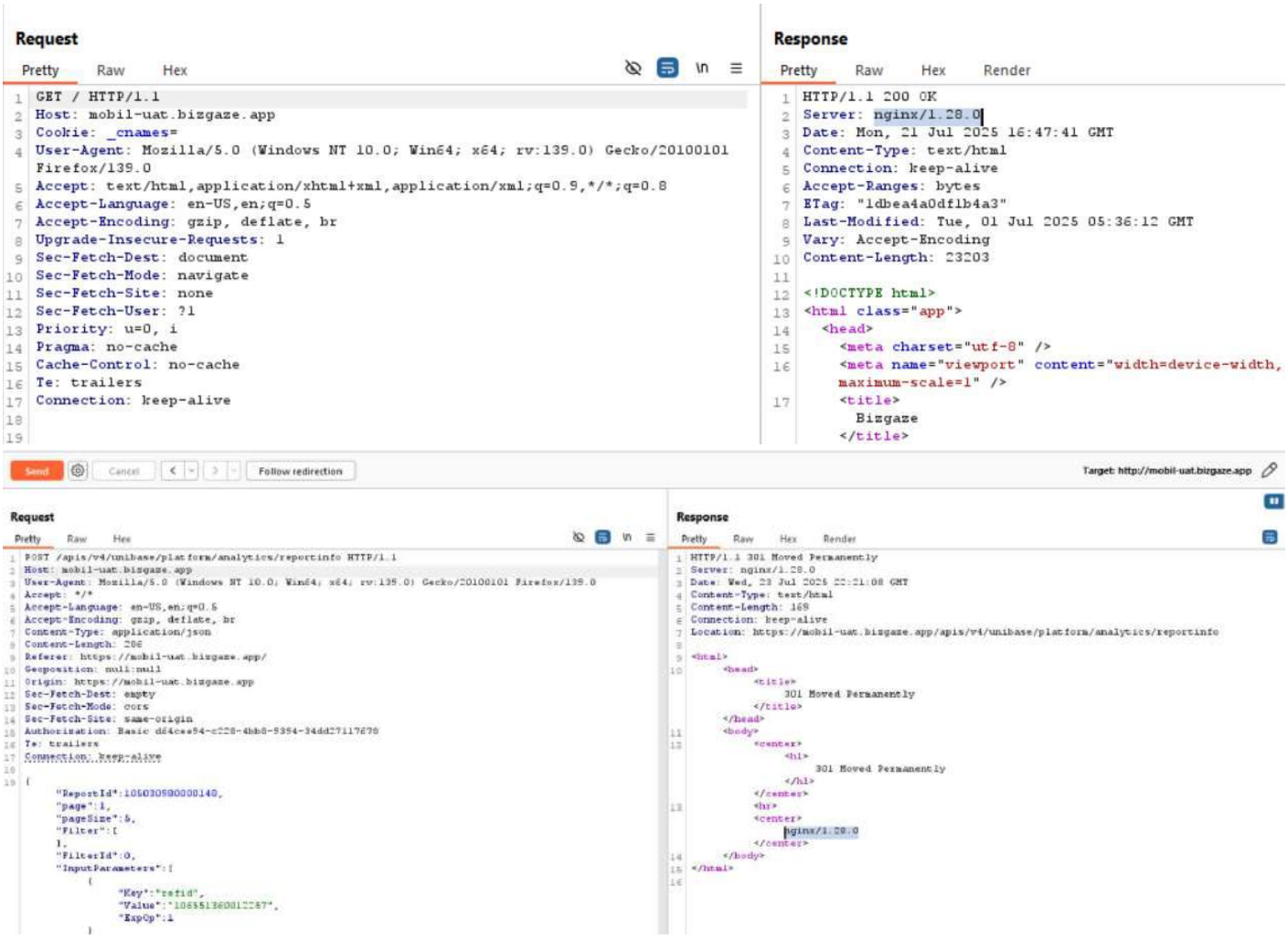
The HTTP Response header unnecessarily discloses version information about the server, middleware components and/or application platform framework.

Instance:

URL	Host Name	IP Address	Version
https://mobiltest.bizgaze.app	-	-	-

Details:

The web application return responses that include server headers. These headers can offer an attacker insight into what technologies and versions are in use by the application.



Implication:

Version disclosures provide an opportunity for an attacker to tailor attacks based on known attack vectors.

Recommendation:

Version headers in the HTTP response are commonly part of the default configuration of applications and servers. Modify the configuration to remove this information.

Additional Tester Input:

Reference:

CWE: 215
Defect ID: 67331

12. Insufficient Origin Validation | Score: Low

Description:

An Origin validation error occurs when applications do not properly authorize an origin of request. As background reference, JavaScript executing in a web browser in one HTTP document, www.example.com, cannot access the document on www.example2.com as the domains are different and violates the browsers Same Origin Policy.

https://en.wikipedia.org/wiki/Same-origin_policy

Modern web browsers commonly provide native extensions to the JavaScript Document Object Model (DOM) in the form of Cross-Origin Resource Sharing (CORS) and Web Sockets that have their Origin or Cross-Domain Security Policies. When these cross-domain network APIs are utilized, the browser includes the HTTP Origin Request Header to indicate to the application the Origin of the request. Incumbent upon the application is the performance of Origin validation to determine if the Origin of the cross-domain request is authorized to access the application resources.

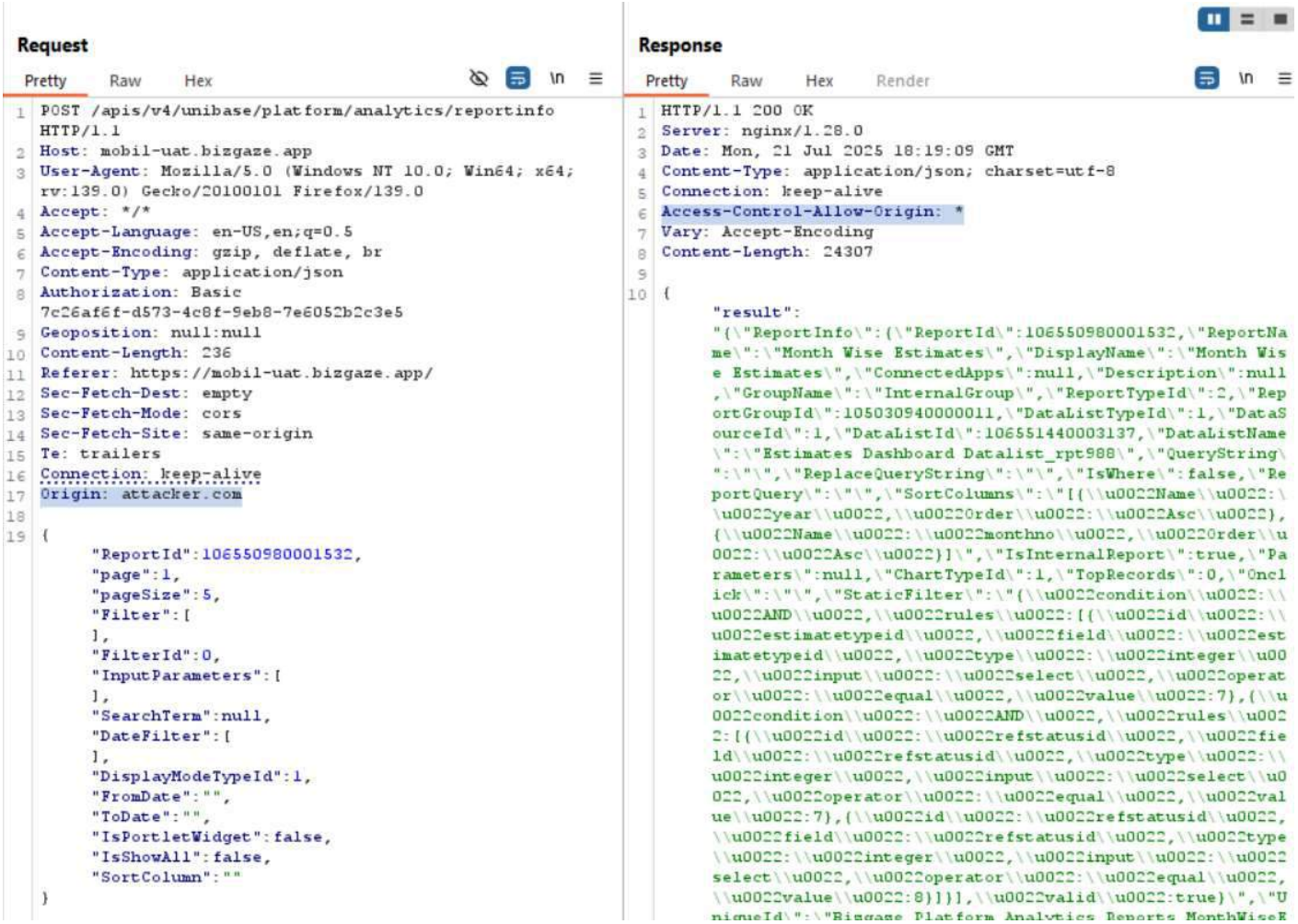
Insufficient, or broken, validation of the Origin Request Header provides no restrictions on cross-domain requests. Applications leveraging the capabilities of CORS and Web Sockets must enforce validation checks on the value of the Origin header to minimize the threat of attack from malicious web applications that a user may inadvertently visit.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application accepts requests from third party origins. This increases the risk of various attacks including information disclosure to malicious sites.



Request

Pretty Raw Hex

```
1  POST /apis/v4/unibase/platform/analytics/reportinfo
   HTTP/1.1
2  Host: mobil-uat.bizgaze.app
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
   rv:139.0) Gecko/20100101 Firefox/139.0
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/json
8  Authorization: Basic
   7c26af6f-d573-4c8f-9eb8-7e6052b2c3e5
9  Geoposition: null:null
10 Content-Length: 236
11 Referer: https://mobil-uat.bizgaze.app/
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16 Connection: keep-alive
17 Origin: attacker.com
18
19 {
       "ReportId":106550980001532,
       "page":1,
       "pageSize":5,
       "Filter":[
       ],
       "FilterId":0,
       "InputParameters":[
       ],
       "SearchTerm":null,
       "DateFilter":[
       ],
       "DisplayModeTypeId":1,
       "FromDate":"",
       "ToDate":"",
       "IsPortletWidget":false,
       "IsShowAll":false,
       "SortColumn":""
   }
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: nginx/1.28.0
3  Date: Mon, 21 Jul 2025 18:19:09 GMT
4  Content-Type: application/json; charset=utf-8
5  Connection: keep-alive
6  Access-Control-Allow-Origin: *
7  Vary: Accept-Encoding
8  Content-Length: 24307
9
10 {
       "result":
       "{\"ReportInfo\":{\"ReportId\":106550980001532,\"ReportNa
       me\":\"Month Wise Estimates\",\"DisplayName\":\"Month Wis
       e Estimates\",\"ConnectedApps\":null,\"Description\":null
       ,\"GroupName\":\"InternalGroup\",\"ReportTypeId\":2,\"Rep
       ortGroupId\":105030940000011,\"DataListTypeId\":1,\"DataS
       ourceId\":1,\"DataListId\":106551440003137,\"DataListName
       \":\"Estimates Dashboard Datalist_rpt988\",\"QueryString\
       ":\"\",\"ReplaceQueryString\":\"\",\"IsWhere\":false,\"Re
       portQuery\":\"\",\"SortColumns\":\"[{\\u0022Name\\u0022:\
       \u0022year\\u0022,\\u00220rder\\u0022:\\u0022Asc\\u0022},
       {\\u0022Name\\u0022:\\u0022monthno\\u0022,\\u00220rder\\u
       0022:\\u0022Asc\\u0022}]\",\"IsInternalReport\":true,\"Pa
       rameters\":null,\"ChartTypeId\":1,\"TopRecords\":0,\"Oncl
       ick\":\"\",\"StaticFilter\":\"{\\u0022condition\\u0022:\\
       u0022AND\\u0022,\\u0022rules\\u0022:[{\\u0022id\\u0022:\\
       u0022estimatetypeid\\u0022,\\u0022field\\u0022:\\u0022est
       imatetypeid\\u0022,\\u0022type\\u0022:\\u0022integer\\u00
       22,\\u0022input\\u0022:\\u0022select\\u0022,\\u0022operat
       or\\u0022:\\u0022equal\\u0022,\\u0022value\\u0022:7},{\\u
       0022condition\\u0022:\\u0022AND\\u0022,\\u0022rules\\u002
       2:[{\\u0022id\\u0022:\\u0022refstatusid\\u0022,\\u0022fie
       ld\\u0022:\\u0022refstatusid\\u0022,\\u0022type\\u0022:\\
       u0022integer\\u0022,\\u0022input\\u0022:\\u0022select\\u0
       022,\\u0022operator\\u0022:\\u0022equal\\u0022,\\u0022val
       ue\\u0022:7},{\\u0022id\\u0022:\\u0022refstatusid\\u0022,
       \\u0022field\\u0022:\\u0022refstatusid\\u0022,\\u0022type
       \\u0022:\\u0022integer\\u0022,\\u0022input\\u0022:\\u0022
       select\\u0022,\\u0022operator\\u0022:\\u0022equal\\u0022,
       \\u0022value\\u0022:8}]}],\\u0022valid\\u0022:true}\",\"U
       niqueId\":\"Bizgaze Platform Analytics Reports MonthWiseE
```

Implication:

Rogue web pages and malicious domains may force a visiting user to interact with the application in unauthorized ways. This can lead to the exploitation of known vulnerabilities, Cross-Site Request Forgery, denial of service attacks, loss of confidentiality and integrity of data.

Recommendation:

Validate the Origin header before authorizing access to the web resource. The following is an example of the Origin Request Header:

Origin: http://www.example.com

Ensure the entire Origin value of the request header is validated using an absolute string match or Regex pattern match delimiters such as ^ for start of string and $ for end of string. For example, Request.Headers["Origin"] == "http://www.example.com"

Or

```
Regex regex = new Regex(@"^http://www.example.com/");
if (regex.IsMatch(Request.Headers["Origin"].ToLower()))
// process request;
```

Insufficient pattern matching can result in a bypass of the validation logic by attackers from a domain such as http://www.example.com.evilsite.com.

Additionally for CORS, upon validation, the application must provide directives to the browser to allow/deny access to these resources via the Access-Control-Allow-Origin response header. Other granular response headers may be required: https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS#The_HTTP_response_headers

Additional Tester Input:

Reference:

CWE: 346
Defect ID: 67338

13. Missing HTTP Security Response Headers | Score: Low

Description:

HTTP response headers which are used to enhance the security posture of the application were not used.

The Strict-Transport-Security HTTP header is used to instruct the browser to only access a web application over a secure connection and for how long to remember this restriction (twelve months is recommended), thereby forcing continued use of a secure connection. (Note that web browsers will only honor this header when delivered over a trusted, secure connection.) This header cannot completely defend against man-in-the-middle attacks, but providing that the user has previously visited the site without outside interference, it can be useful in defending against an attack in which an attacker establishes an encrypted connection to the application and presents an unencrypted fraudulent service to the user, as the user's browser will know not to use the unencrypted service. This type of attack has become more prevalent and has received widespread media attention following the publishing of the easy-to-use SSLStrip attack tool.

The X-Content-Type-Options HTTP header can be used to prevent web browsers from using content sniffing to discover a file's MIME type. This header, when set, can help protect against Cross-Site Scripting attacks.

The Cache-Control HTTP header provides control over how pages can be cached either by proxies or a user's browser. Using this response header can provide enhanced privacy by ensuring that sensitive content is not cached in a user's browsers or intermediary proxy, where it could potentially be recovered by an attacker.

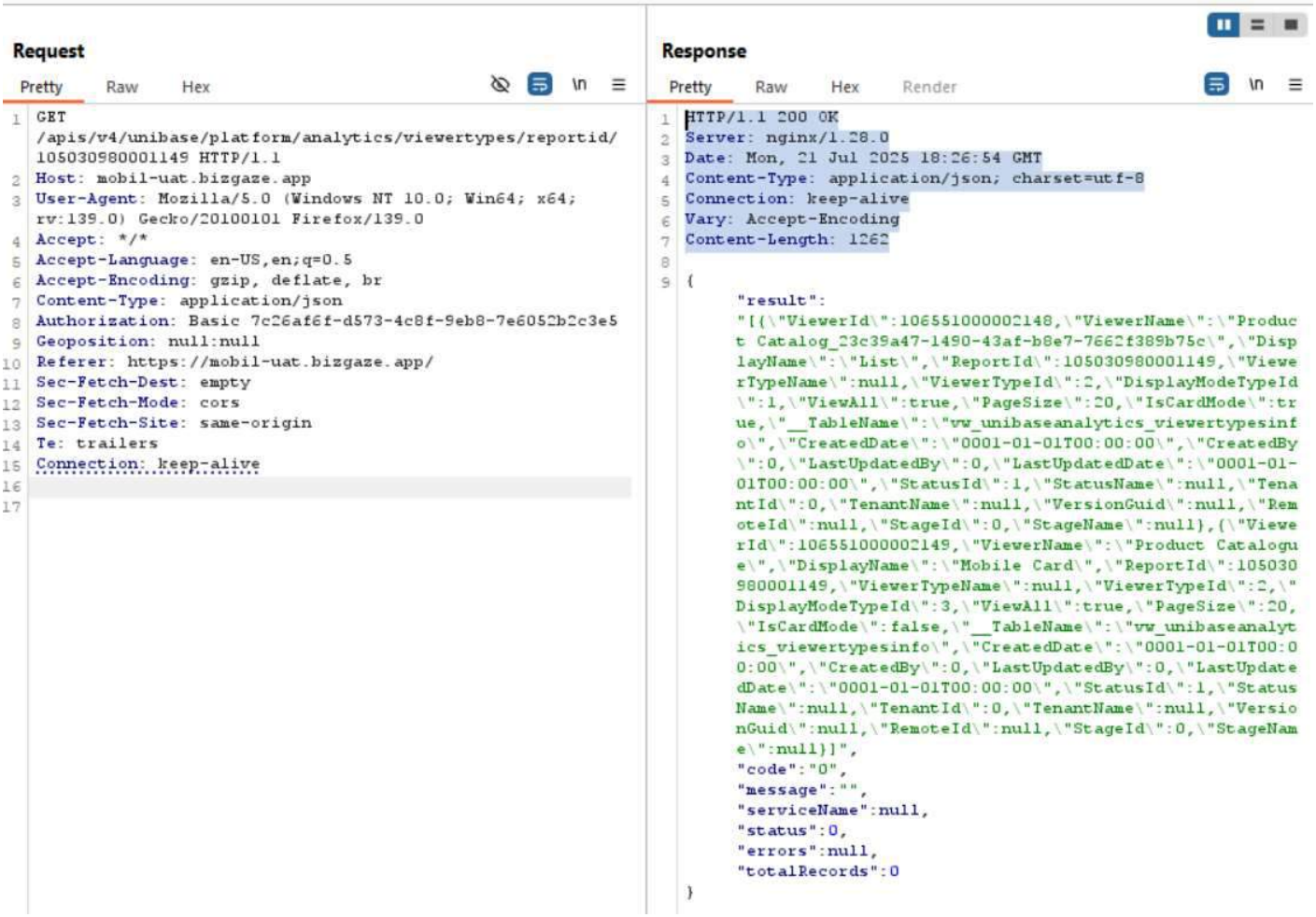
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

Responses of the application are missing best practice security headers:

1) X-Content-Type-Options
2) Strict-Transport-Security
3) Cache-control
4) X-Frame-Options



Request

Pretty Raw Hex

```
GET
/apis/v4/unibase/platform/analytics/viewertypes/reportid/
105030980001149 HTTP/1.1
Host: mobil-uat.bizgaze.app
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:139.0) Gecko/20100101 Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Basic 7c26af6f-d573-4c8f-9eb8-7e6052b2c3e5
Geoposition: null:null
Referer: https://mobil-uat.bizgaze.app/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Mon, 21 Jul 2025 18:26:54 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Vary: Accept-Encoding
Content-Length: 1262

{
    "result":
    "[{\"ViewerId\":106551000002148,\"ViewerName\":\"Produc
    t Catalog_23c39a47-1490-43af-b8e7-7662f389b75c\",\"Disp
    layName\":\"List\",\"ReportId\":105030980001149,\"Viewe
    rTypeName\":null,\"ViewerTypeId\":2,\"DisplayModeTypeId
    \":1,\"ViewAll\":true,\"PageSize\":20,\"IsCardMode\":tr
    ue,\"__TableName\":\"vw_unibaseanalytics_viewertypesinf
    o\",\"CreatedDate\":\"0001-01-01T00:00:00\",\"CreatedBy
    \":0,\"LastUpdatedBy\":0,\"LastUpdatedDate\":\"0001-01-
    01T00:00:00\",\"StatusId\":1,\"StatusName\":null,\"Tena
    ntId\":0,\"TenantName\":null,\"VersionGuid\":null,\"Rem
    oteId\":null,\"StageId\":0,\"StageName\":null},{\"Viewe
    rId\":106551000002149,\"ViewerName\":\"Product Catalogu
    e\",\"DisplayName\":\"Mobile Card\",\"ReportId\":105030
    980001149,\"ViewerTypeName\":null,\"ViewerTypeId\":2,\"
    DisplayModeTypeId\":3,\"ViewAll\":true,\"PageSize\":20,
    \"IsCardMode\":false,\"__TableName\":\"vw_unibaseanalyt
    ics_viewertypesinfo\",\"CreatedDate\":\"0001-01-01T00:0
    0:00\",\"CreatedBy\":0,\"LastUpdatedBy\":0,\"LastUpdate
    dDate\":\"0001-01-01T00:00:00\",\"StatusId\":1,\"Status
    Name\":null,\"TenantId\":0,\"TenantName\":null,\"Versio
    nGuid\":null,\"RemoteId\":null,\"StageId\":0,\"StageNam
    e\":null}]",
    "code":"0",
    "message":"",
    "serviceName":null,
    "status":0,
    "errors":null,
    "totalRecords":0
}
```

Implication:

The recommended HTTP security response headers provide a means for browsers to apply a stronger security policy per application. As such, the application should set these security response headers to direct browsers to apply the proper security settings for the duration of the application session.

Recommendation:

Set the following HTTP headers:
* Strict-Transport-Security: max-age=31536000; includeSubDomains
* X-Content-Type-Options: nosniff
* Cache-control: no-store, no-cache
* X-Frame-Options: SAMEORIGIN

Additional Tester Input:

Reference:

CWE: 200
Defect ID: 67339

14. HttpOnly Cookie Flag Not Set | Score: Low

Description:

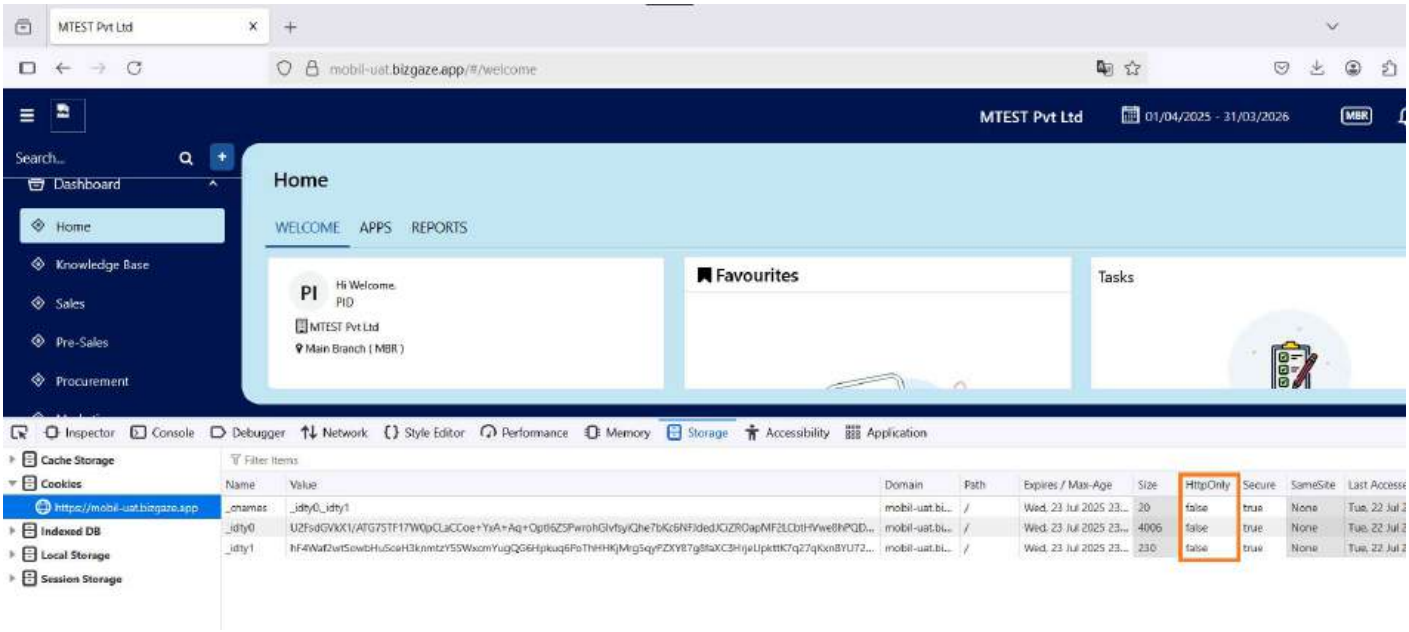
The HttpOnly keyword is a security feature to mitigate the possibility of a successful Cross-Site Scripting attack by not allowing cookies with the HttpOnly attribute to be accessed via client-side scripts.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application uses cookies that do not have the HttpOnly attribute set. This allows the cookies to be accessed by client-side scripts, exposing it to scripting attacks.



Implication:

Without the HttpOnly keyword set, the possibility exists that an attacker could leverage a Cross-Site Scripting vulnerability to read, manipulate or steal any data contained within the cookie. This could potentially allow an attack to steal a value such as a sessionID and impersonate a user within the application.

Recommendation:

Recommendations include adopting a development policy that includes the utilization of HttpOnly cookies and performing other actions such as ensuring proper filtration of user-supplied data, utilizing client-side validation of user supplied data, and encoding all user supplied data to prevent inserted scripts being sent to end users in a format that can be executed.

Additional Tester Input:

Reference:

CWE: 284
Defect ID: 67341

15. Cookie Attribute - SameSite Attribute Missing or Misconfigured | Score: Low

Description:

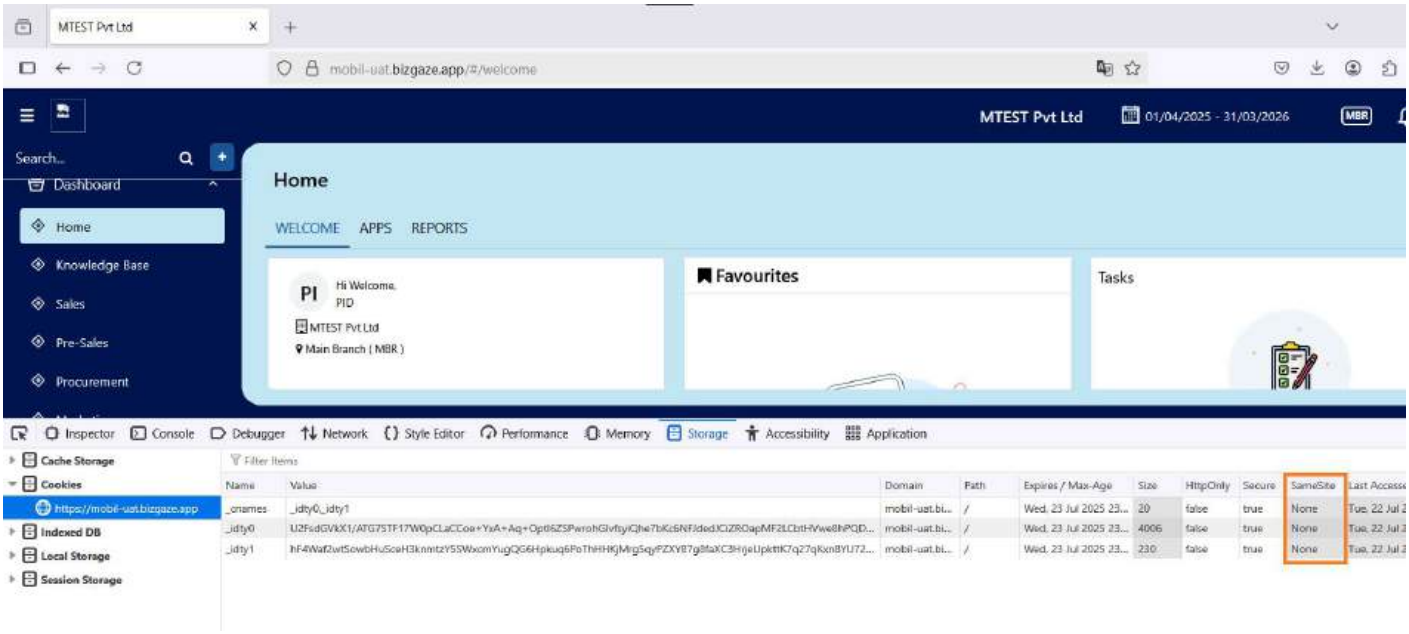
The SameSite attribute restricts the browser from sending cookies in certain cross-site requests. A value of "strict" or "lax" should be used for any session cookie to restrict it from being sent in unauthorized cross-site request. The SameSite attribute restricts the browser from sending cookies in certain cross-site requests. This can provide protection against cross-origin information leakage and Cross-Site Request Forgery (CSRF) attacks. When set to "strict", the cookie will be sent with same-site requests only. When set to the default value of "lax", the cookie will be withheld on cross-site sub-requests (i.e. load images, iframes), but will be sent on any top-level navigation to a URL from an external site (e.g. following a link). In the absence of the cookie attribute, browser vendors have implemented a default behavior of SameSite=lax. However, as browser vendors may change the default behavior in future iterations we recommend, as is customary, to explicitly set the value of this attribute for session cookies.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application uses cookies that do not have the SameSite attribute set properly. This attribute helps prevent cross-site request forgery (CSRF) attacks.



Implication:

SameSite cookie protections provide an additional Defense in Depth countermeasure against Cross-Site Request Forgery attacks. In the absence of this protection for session cookies, an application that may not have full coverage against CSRF attacks may be vulnerable to unauthorized transactions on behalf of a user visiting a malicious website. For example, an active session user visits http://attacker.org which contains malicious scripts that sends POST requests to https://finance.exxonmobil.com

Recommendation:

In most cases, upon user logon applications should set post-authenticated session cookies with SameSite=strict.

An additional guidance is provided for ExxonMobil “Cloud Ready” applications that integrate with the enterprise Identity Providers (IdP), such as, Azure Active Directory (AAD), Auth0, and SAP Identity. These applications may find it preferable to establish two session cookies one for pre-authentication status and the other for post-authentication status. A SameSite=lax cookie may be used during pre-authentication to track the anonymous user during the redirections to and from the IdP. Once the user has been authenticated and identified a post-authentication cookie should be set with SameSite=strict to provide the intended security protections of this cookie attribute.

As an advisement, there are rare occasions, applications are developed to be hosted within an iframe of larger company web portal application. In those cases, the cookie should be set with SameSite=None to ensure proper functioning of the application.

In all cases, proper Cross-Site Request Forgery countermeasures must be implemented to reduce the likelihood of this attack in light of current, and future, browser capabilities that may lead to exploitation of this vulnerability.

Additional Tester Input:

Reference:

CWE: 1275
Defect ID: 67342

16. Username Enumeration (Error Responses) | Score: Low

Description:

A different error is displayed to the end user when an invalid username is presented as opposed to a correct username with an incorrect password.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

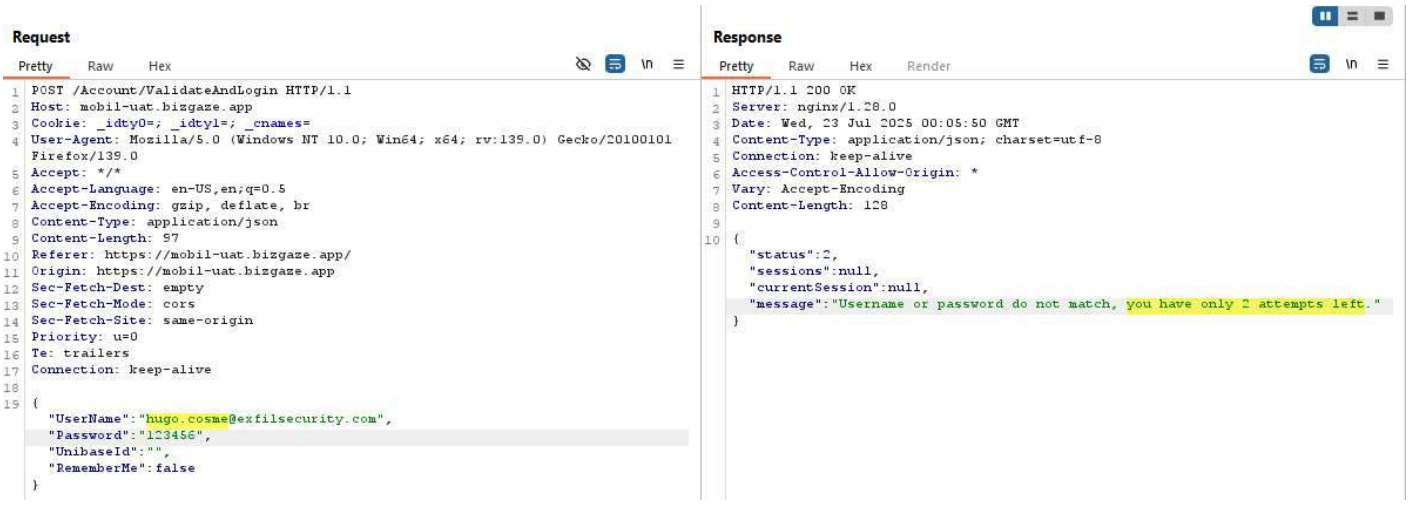
Details:

Different responses are provided for requests with valid and invalid emails. This can help an attacker identify valid users, facilitating phishing attacks.

Login endpoint:

- Invalid username:

- Valid username:



Password recovery endpoint:

- Invalid email:

Request

```
POST /account/sendotp HTTP/1.1
Host: mobil-uat.bizgaze.app
Cookie: _idty0=; _idty1=; _cnames=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 254
Referer: https://mobil-uat.bizgaze.app/index.html
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "ContactOrEmail":"hugo.cosme+1@exfilsecurity.com",
  "LastName":"",
  "ContactNumber":"",
  "Email":"hugo.cosme+1@exfilsecurity.com",
  "TenantName":"",
  "IsSignup":false,
  "IsForgotPswd":false,
  "IsRegisterUser":true,
  "UnibaseId":"",
  "OtpId":0,
  "UserOtp":"",
  "TenantId":681
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 00:03:02 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 201

{
  "result":0,
  "code":"500",
  "message":"Email/Phonenumber does not exist in selected Tenant",
  "serviceName":null,
  "status":2,
  "errors":[
    "Email/Phonenumber does not exist in selected Tenant"
  ],
  "totalRecords":0
}
```

- Valid email:

Request

```
POST /account/sendotp HTTP/1.1
Host: mobil-uat.bizgaze.app
Cookie: _idty0=; _idty1=; _cnames=
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 250
Referer: https://mobil-uat.bizgaze.app/index.html
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers
Connection: keep-alive

{
  "ContactOrEmail":"hugo.cosme@exfilsecurity.com",
  "LastName":"",
  "ContactNumber":"",
  "Email":"hugo.cosme@exfilsecurity.com",
  "TenantName":"",
  "IsSignup":false,
  "IsForgotPswd":false,
  "IsRegisterUser":true,
  "UnibaseId":"",
  "OtpId":0,
  "UserOtp":"",
  "TenantId":681
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 00:03:14 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 187

{
  "result":0,
  "code":"500",
  "message":"Email or PhoneNumber is already registered!.",
  "serviceName":null,
  "status":2,
  "errors":[
    "Email or PhoneNumber is already registered!."
  ],
  "totalRecords":0
}
```

Implication:

By signaling to the end user whether the username is correct or not, you give an attacker the ability to discover all usernames in the system through trial and error.

Recommendation:

A generic error message, such as "Invalid username or password" should be displayed in both cases.

Additional Tester Input:

Reference:

CWE: 200
Defect ID: 67343

17. Weak Password Policy | Score: Low

Description:

A weak password policy makes it significantly easier for attackers to guess correct passwords on the domain. They can also make it so that attackers can guess passwords without being locked out, therefore allowing them to determine valid passwords on most accounts, given sufficient time.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The application`s server allows weak passwords such as "Password" or "Elephant" if the request is intercepted and modified. This could allow an attacker to brute force a user`s account.

Request

Pretty Raw Hex

```
POST /account/UpdatePassword HTTP/1.1
Host: mobil-uat.bizgaze.app
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101
Firefox/139.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 132
Referer: https://mobil-uat.bizgaze.app/
Geoposition: null:null
Origin: https://mobil-uat.bizgaze.app
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Authorization: Basic 7a0e2daa-a521-4205-ba73-878baa989fef
Priority: u=0
Te: trailers
Connection: keep-alive

{
    "Password":"Password",
    "UserName":"1744198fd7f74f3ab467251db4a10c33",
    "OtpId":0,
    "UserOtp":"",
    "OldPassword":"Elephant",
    "IsReset":true
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Wed, 23 Jul 2025 19:52:21 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Content-Length: 129

{
    "result":null,
    "code":"0",
    "message":"Password Updated Successfully",
    "serviceName":null,
    "status":0,
    "errors":null,
    "totalRecords":0
}
```

Implication:

An attacker with some patience will be able to guess valid passwords on the domain, giving them valid credentials for various services, from FTP accounts to domain administrators.

Recommendation:

Consider adding more password complexity, a minimum length of 8 characters, and a lockout policy.
As with all controls in the Process Control Network, this should be compared to the risk of the account in question.
Additionally, service accounts should be hardened using Fine-Grained password policies. They should be much longer (minimum 25 characters, in most instances), and can be set to not expire.

Additional Tester Input:

Reference:

NA
Defect ID: 67345

18. Session Management - Allows Concurrent Sessions | Score: Low

Description:

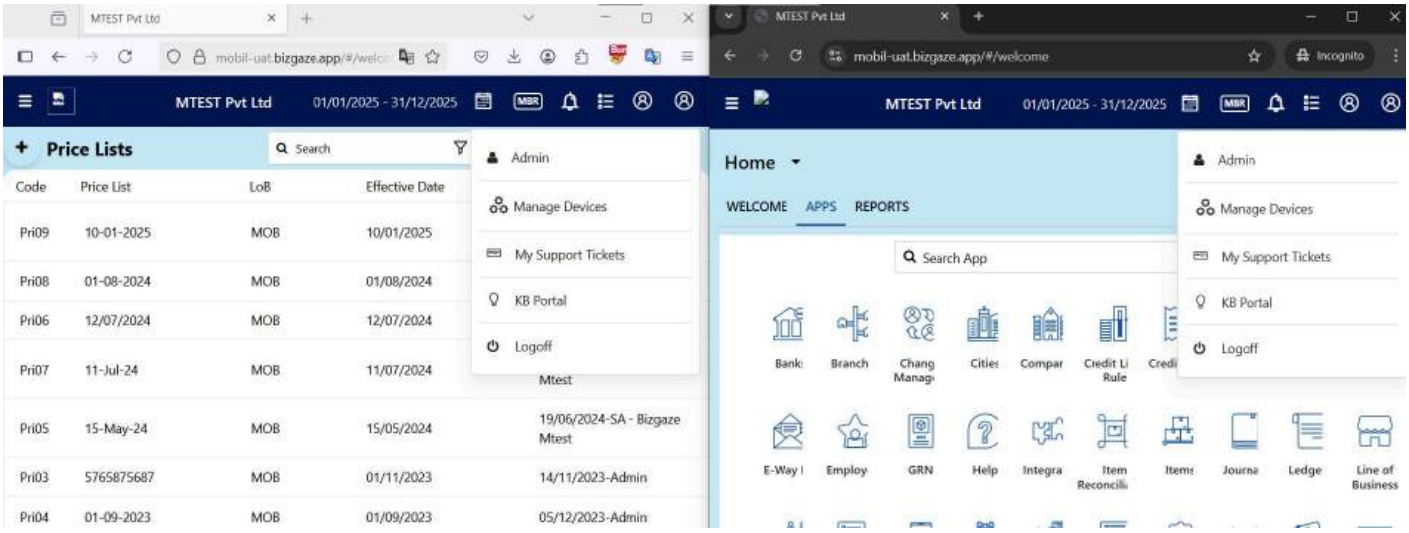
The application supports concurrent sessions, enabling an attacker who has compromised another user's credentials to make use of them without risk of detection.

Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The application allows the use of concurrent sessions across different browsers. If a user`s account is compromised, they will be unaware.



Implication:

An attacker who has compromised another user's credentials could continue to use their session undetected, increasing the likelihood of breaches in integrity and confidentiality.

Recommendation:

Consider invalidating current user sessions server-side upon subsequent user login. Notification can also be made to the terminated session along with pertinent information such as the IP address of the new session holder as well as contact information for the site's security administration.

Additional Tester Input:

Reference:

-
Defect ID: 67350

19. Frameable HTTP Response | Score: Low

Description:

A Frameable HTTP Response can allow an attacker to load the vulnerable application inside an unauthorized hidden HTML iframe tag on a malicious page to carry out various attack scenarios.

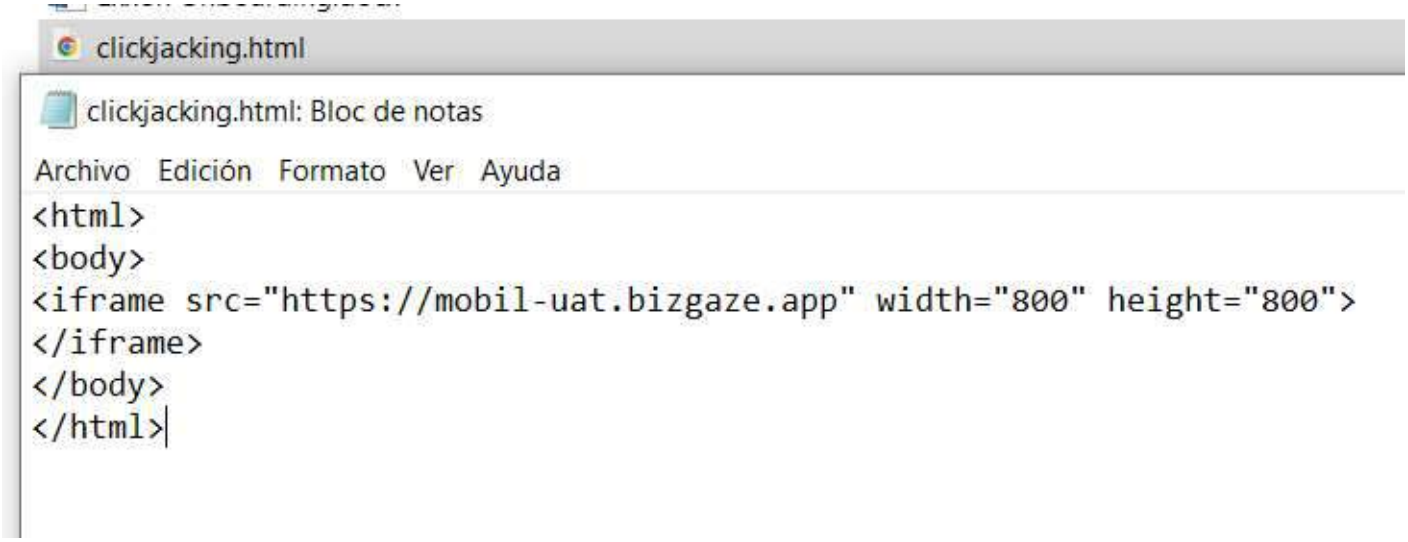
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

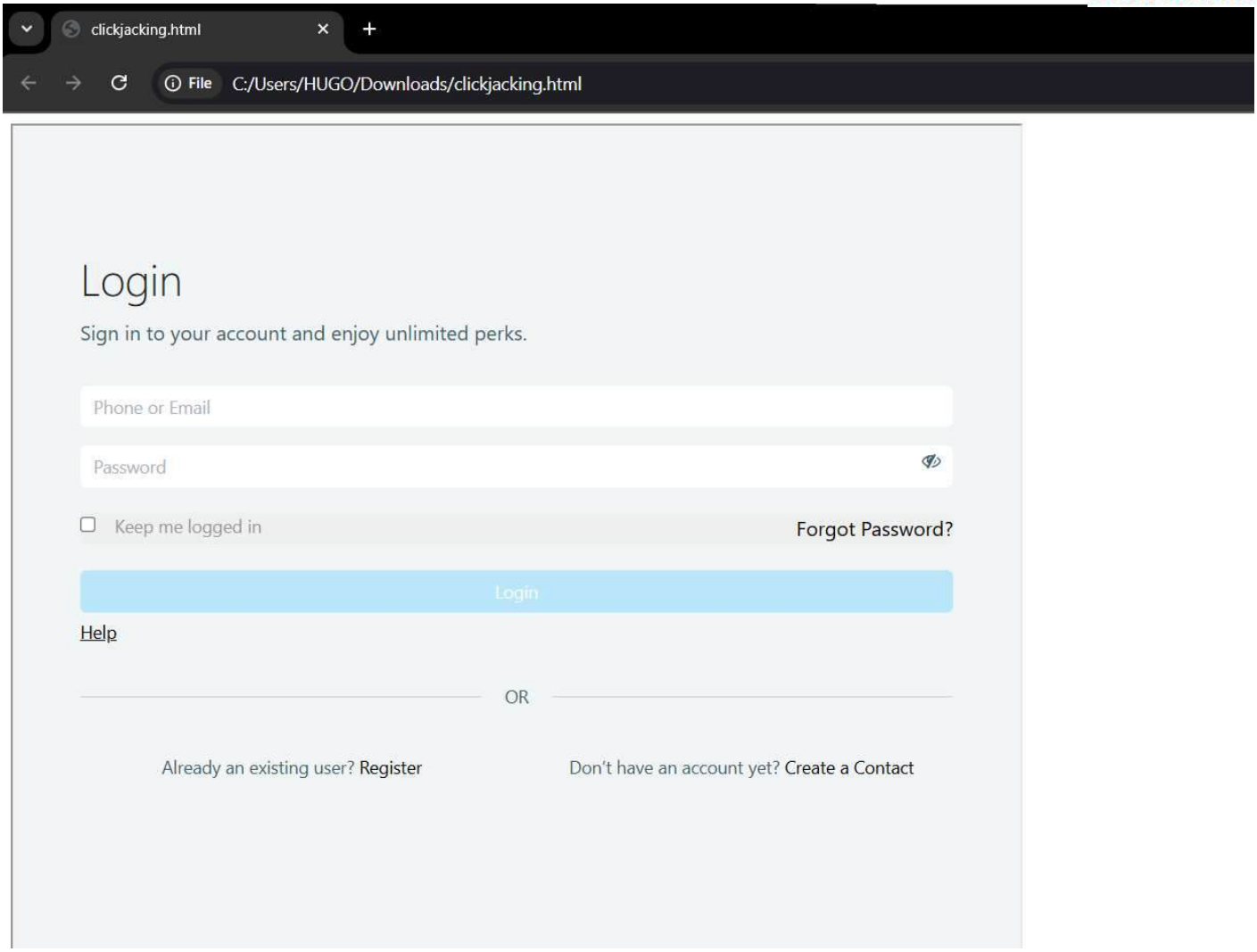
Details:

The web application can be loaded into an attackers frame, potentially enabling other attacks against the application.

- HTML test file:



- Site loaded within a frame:



Implication:

A Frameable HTTP Response weakness could allow an attacker to embed the vulnerable application inside an iframe. Exploitation of this weakness could result in:

1. Exploitation of XSS vulnerabilities in a target web site.
2. Unauthorized rendering of sensitive content.
3. Determine if a user is logged on to the website, to mount other attack scenarios.
4. Hijacking of user events such as keystrokes.
5. Theft of sensitive information through phishing attacks.
6. Execution of privileged functionality through combination with Cross-Site Request Forgery attacks.
7. Clickjacking attacks.

The framed target site may be obscured from the user through a hidden iframe. Alternatively, the frame's UI can be redressed with a CSS Overlay to entice the user to interact with the malicious site. The victim interaction events are actually triggered on the target site to carry out privileged functionality.

Recommendation:

HTTP Response Framing can be mitigated through browser policy directives using X-Frame-Options or Content-Security-Policy response headers. These headers have options that are not ubiquitously supported across all major browsers. Therefore, some guidance is offered below to determine which header, value, and options are appropriate for the given circumstance.

Restrict All Framing:
This is the most common case, in which applications do not rely on the use of parent/child iframes. In such a case, emitting the below HTTP response header and value is sufficient.

X-Frame-Options: DENY

Note: This DENY configuration for the X-Frame-Options header is supported by all modern versions of the major browsers (i.e., Microsoft Edge, Google Chrome, Mozilla Firefox).

Restrict Framing to Same Domain:
At times, an application may contain a child iframe from the same domain. For instance, http://www.example.org/main.html may contain an iframe source of http://www.example.com/frame.html. In such a case, the application must allow framing by the same domain (e.g., http://www.example.com) and reject other domains. Emitting the below HTTP response header will deliver the desired results.

X-Frame-Options: SAMEORIGIN

Note: This SAMEORIGIN configuration for X-Frame-Options header is supported by all modern versions of major browsers as well.

Alternatively, the Content-Security-Policy header allows framing by multiple domains and is much more effective in securing the application against framing chains. Support for wildcard subdomains are also supported.

Content-Security-Policy: frame-ancestors 'self' https://www.somesite.org; https://www.othersite.com;

ExxonMobil primarily supports Microsoft Edge and Google Chrome browsers. As such, when framing is not intended, `X-Frame-Options: DENY` is the most common configuration. This can be set in web server and application in configuration files.

Apache:
Header always append X-Frame-Options DENY

IIS and ASP.NET web.config:
</httpProtocol>
<customHeaders>
<add name="X-Frame-Options" value="DENY" />
</customHeaders>
</httpProtocol>

Additional Guidance:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors

Additional Tester Input:

Reference:

CWE: 79
Defect ID: 67352

20. Verbose Errors | Score: Low

Description:

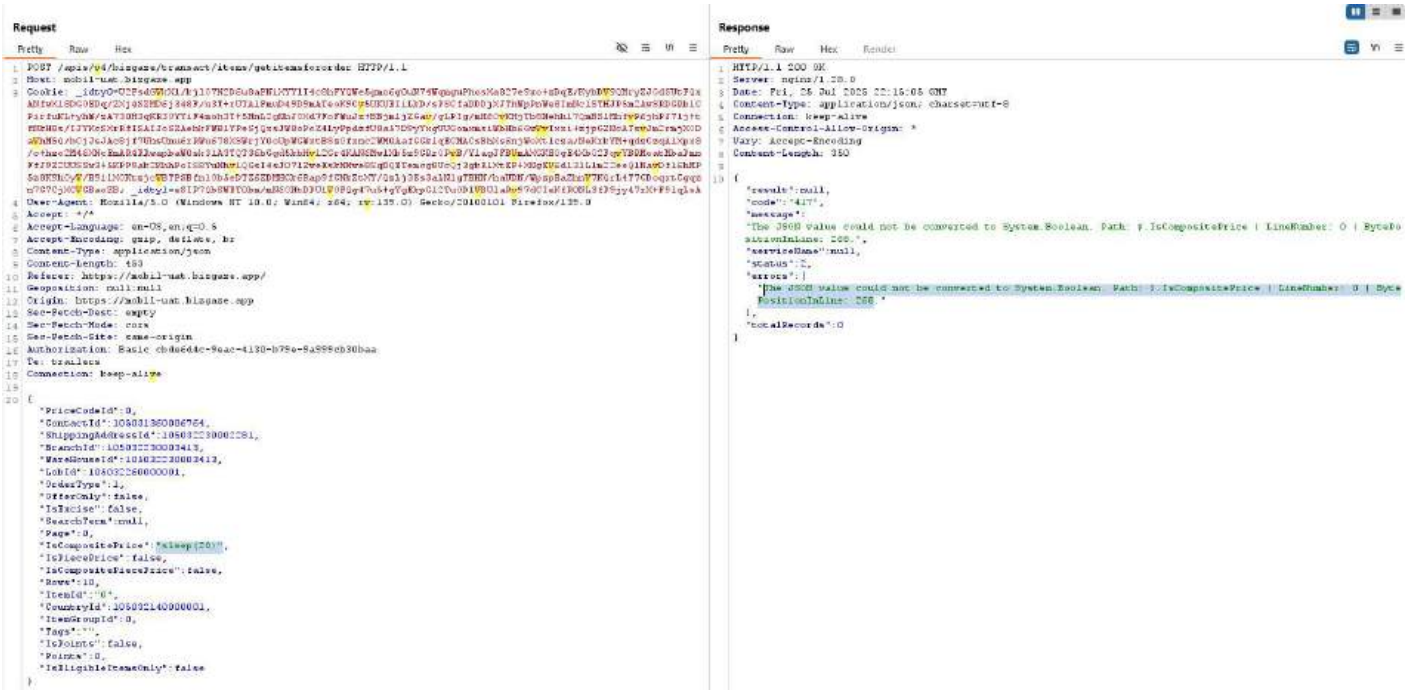
While evaluating the application, verbose error messages were encountered that included significant information regarding the system and component being tested.

Instance:

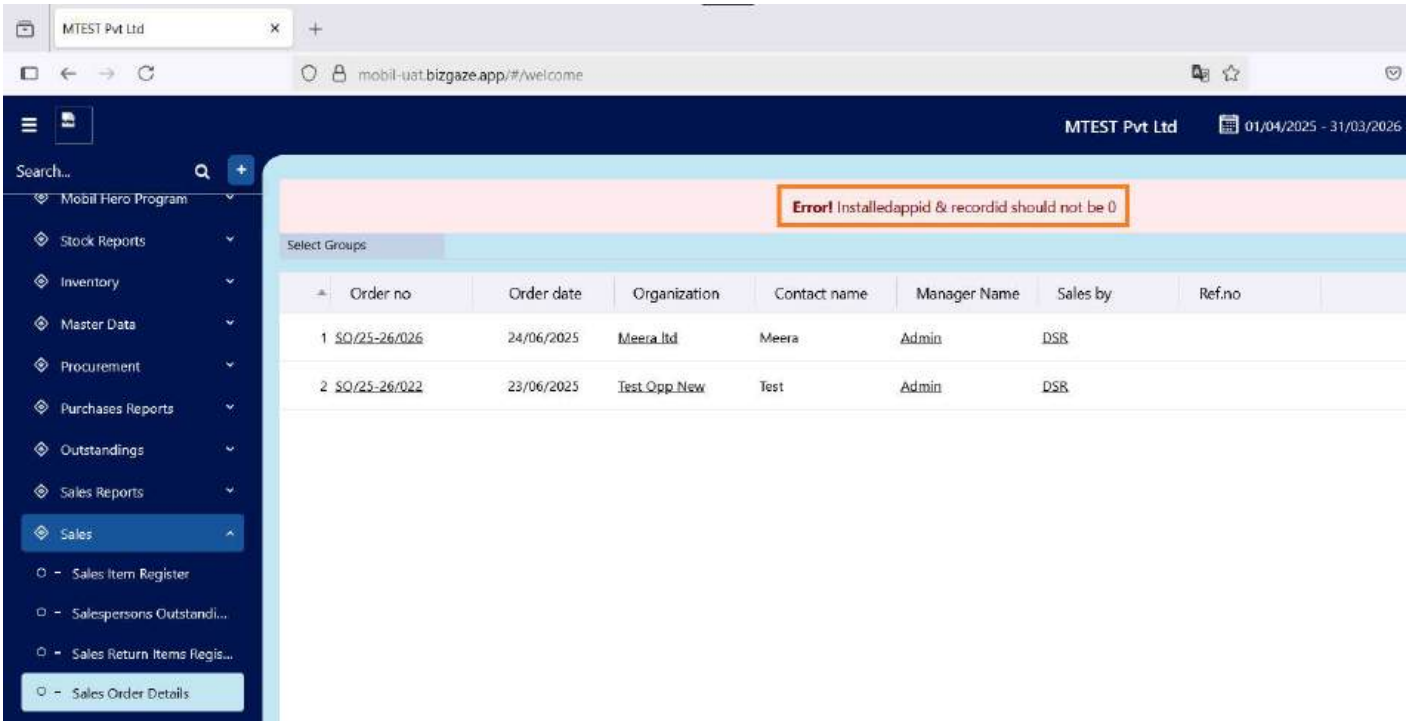
URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application returns verbose errors on some endpoints when handling unexpected input. The errors provide an attacker with insights into the application`s backend structure.



When selecting a group in the following section:



Implication:

The danger from presenting verbose errors to the end-user comes from providing a potential attacker with too much information about the system. Depending on the exact details in a verbose error message, it could be determined what conditions caused the error, as well as potentially information about the back-end database, and injection opportunities.

Recommendation:

Ensure that any error messages presented on screen do not include any sensitive information so as not to provide a potential attacker with more information than is needed. Further error details should be logged to a location available only to the support team, and not the general userbase.

Additional Tester Input:

Reference:

CWE: 200
Defect ID: 67357

21. Unrestricted File Upload | Score: Informational

Description:

The software allows files to be uploaded, however, the software does not have controls in place to prevent uploading arbitrary files.

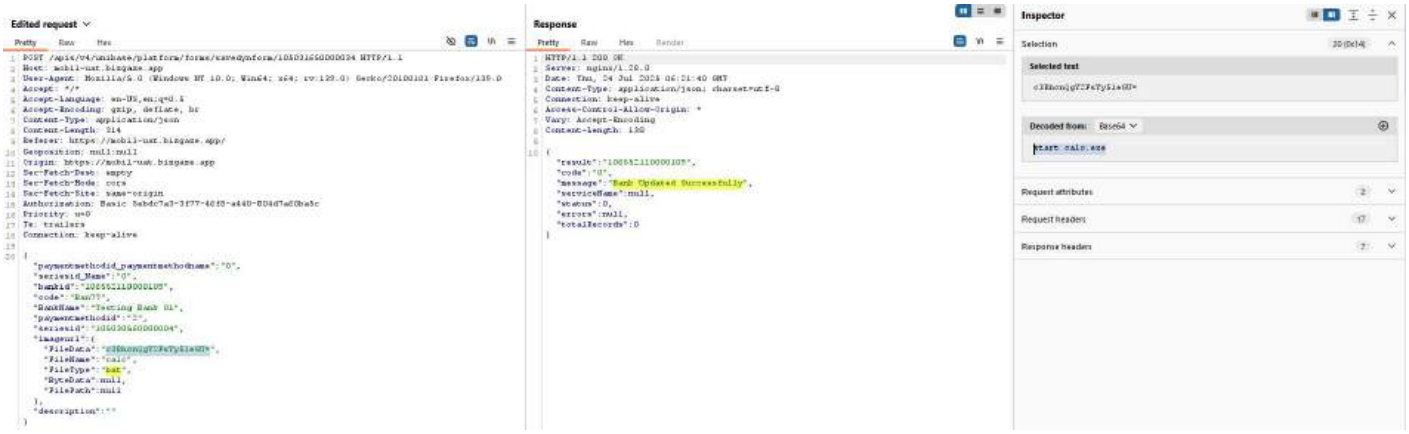
Instance:

URL	Host Name	IP Address	Version
https://mobil-uat.bizgaze.app	-	172.105.61.152	-

Details:

The web application does not restrict some types of files that can be uploaded. This could allow a user to upload a malicious file that might impact others if they handle the file.

For instance, the application is validating the ".bat" file type only on the client side, allowing bypass through request interception and modification. The risk is mitigated by the fact that access to the uploaded file could not be verified.



Implication:

Insufficient controls to protect file upload could lead to a variety of issues such as code execution in the context of the server or client, Cross-Site Scripting, denial of service attacks, and more.

Recommendation:

It is recommended that the software utilize controls to ensure that uploaded files are of the correct type. Some web applications utilize file type recognizers that check the type of the file before processing. For example, image uploading functionality may go through an image type recognizer to ensure that an actual image file is being uploaded. A whitelist for file extensions in combination with other protections is also recommended.

Additional Tester Input:

Reference:

-

Defect ID: 67349