26, April 2025
Reported By :: Zhaenx
Contact :: zhaenx.101@gmail.com

Title	***A01:2021 – Broken Access Control + A04:2021 – Insecure Design (OWASP Top 10)***
URL-Endpoint	***1. https://portal.gcore.com/accounts/profile/users***
Evidence	***BrokenAccessControl_PoC.mp4***
Severity	***Critical***

DESCRIPTION

I discovered a critical logic vulnerability in the Gcore invitation system that allows an attacker to silently "hijack/claim" any email address (even an unregistered one), forcing the victim into the attacker's organization without the victim's knowledge or consent. Worse yet, expired invitation links remain functional well beyond the stated 24-hour expiration, the backend system does not clear the expired invitation status, so it is still considered active, allowing unauthorized access long after the user believes the link is inactive.

STEP TO REPRODUCE

As Attacker:

1. Log in to your Gcore account as the attacker.
2. Edit the "account name" (this is done to make it look anonymous, when the invitation is sent to the victim's email)
3. Invite victim email (e.g. company CEO with specific email e.g. "joyannconvincing@ptct.net") - Note: This email was already invited 5 days ago, with a status of "not activated" — meaning it hasn't been used to register yet.

As Victim:

4. The Gcore system sends an invitation email, but the victim does not know who sent the invitation.
   This is where the victim starts to get suspicious and of course ignores the invitation email because they don't know the sender (anonymous).
   Additionally, the invitation clearly states that "the email will expire after 24 hours and account creation will be canceled." However, the invitation remains active well beyond that time frame, which indicates a critical flaw in your logic.
5. **The proof:** Now as a victim, try to click on the "confirm email" invitation, there the invitation has expired. (But only on the frontend)
6. But if the victim tries to register an account on the Gcore service normally, with his email "joyannconvincing@ptct.net",
   an error message appears:"Client with email joyannconvincing@ptct.net already exists.", Because the email has been taken over (claimed) by the attacker without confirmation/interaction from the victim.
7. Then the victim uses the "Forgot Password" feature to recover his email/account, and successfully logs in.
8. after the victim managed to get in but the impact was very detrimental
   - The victim is unaware of how they were added, and cannot claim ownership/admin of their own account.
   - The victim is trapped in the attacker's team, unless the attacker manually removes them, Etc...

IMPACT

1. **Unauthorized Account Takeover:** Attackers can "claim" anyone's email without any verification or interaction from the victim.
2. **Permanent Attachment to Attacker's Organization:** The victim is indirectly forced into the attacker's organization without any interaction or explicit permission.
3. **Denial of Service (DoS) to Target Email**: Legitimate users are prevented from registering or accessing their accounts unless they go through a password recovery process they did not request.
4. Attackers can send mass email invitations to specific email addresses (e.g., CEO Email), locking them out of registration and misusing ownership of their identities.
5. **Confusion & Reputation Damage**: This vulnerability can be abused by certain parties and can damage your reputation, as users will feel unsafe and Victims of Confusion, trust is lost, as they do not know who the inviter is and why their account already exists.
6. Using this method, attackers can mass "book" important emails from an organization — and make them unable to register legitimately except through a recovery mechanism.

RECOMMENDATION

1. Once the email is verified and normal login is complete, the user should be disconnected from the inviting organization, unless they expressly agree to it.
2. **Invitation Revocation:** Implement a feature for users to revoke or manage invitations directly from the dashboard to prevent abuse.
3. **Expiration Enforcement:** Ensure that invitations are truly disabled once the expiration time is reached and that no invitation can block new user registration.
4. Add validation so that only confirmed invitees will have accounts/emails that are considered "existent".
5. Implement TTL enforcement on the backend to ensure expired invitations are truly not considered active.
6. Automatically delete expired invitations, both on the frontend and backend, to prevent abuse.

Impact + Mapping to OWASP Top 10:

CATEGORY	IMPACT	OWASP TOP 10
Account Takeover (ATO)	The victim's email can be claimed without confirmation, causing an account to be created without the victim's knowledge.	A01:2021 - Broken Access Control
Denial of Registration	Potential users cannot register because the system considers the email to have been used. (without confirmation of email ownership)	A01:2021 - Broken Access Control
Expired Invitation Still Active	The invite is considered expired in the UI, but the backend still receives and links the victim's email to the attacker team.	A04:2021 - Insecure Design
Forced Team Association	The victim is automatically logged into the attacker's organization without any interaction or explicit permission.	A01:2021 - Broken Access Control
Mass Invitation Abuse	Attackers can invite multiple emails from one domain to prevent mass registration.	A04:2021 - Insecure Design + Business Logic Flaw
Confusion & Platform Reputation Declining	The victim is confused, trust is lost, and UX is disturbed because they don't know who the inviter is and why the account already exists.	A05:2021 - Security Misconfiguration

Additional Notes:

- This is not just a technical bug, but a business logic weakness (Business Logic Abuse) that can be used for sabotage, access theft, and preventing legitimate user registration.
- Can be used for targeted attacks against large organizations (e.g. CEO or business partners), with high reputation and trust risks.

Evidence and Impact:

- Invite expired in appearance (UI) but still active in the backend.
- The victim never confirms anything, but the account has been created and associated with the attacker.
- The victim cannot register a new account, because the system assumes the email is already in use.
- The attacker can do this to many emails to prevent other potential users/clients from registering.
- The collaboration model is vulnerable to abuse. Imagine if this happened to a company's executive email (CEO/CFO) — the result could be a ruined internal reputation or a potential insider threat.
- If users ignore the email assuming it will "expire", they could become unwitting victims of exploitation at a later date.
- Even after the victim resets password and logs in, they do not have full control over the account (e.g. losing admin access, not being able to leave the attacker's organization, etc...).

Severity Justification (Severity Reason - Critical):

1. **Account Takeover (Without Victim Interaction)**
   - Attackers can create accounts in the victim's name by simply inviting the victim's email.
   - There is no active validation from the victim (zero-click ATO).
   - After the password is reset, the victim remains in the attacker's organization.
2. **Broken Access Control**
   - The victim never gave permission to join the attacker's team, but the system automatically associated them.
   - The victim has no way to break the connection independently (locked inside attacker's org).
3. **Denial of Service for Registration**
   - Attackers can invite many legitimate emails (e.g. domains from certain companies) to prevent legitimate users from registering their own accounts.
   - This is a form of abuse that can hinder the onboarding of new users (sabutage).
4. **Logic Flaw & Insecure Design**
   - Expired invites are still active logically in the backend.
   - There is no TTL enforcement mechanism or automatic revoke after the time runs out.
   - Frontend-backend synchronization is very weak and can be manipulated for abuse.

Conclusion:

This bug meets the criteria to be categorized as Critical, as it directly impacts:

1. Confidentiality (access to other people's accounts),
2. Integrity (accounts created without authorization),
3. Availability (preventing legitimate user registration),