Bavacai

Technical Analysis — Cti Report

REVERSE-ENGINEERED REPORT

RansomLook · ransomlook.io
File last modified: 2026-05-06

Sample SHA-256: `86b4d075d5bd0c49cbb21fd43935789b6612a2165273cc158dd0607b68941d04`

Executive Summary

Family	BAVACAI (self-identified via `.BAVACAI` extension; internal name `locker_win_x64_encrypter`)
Platform	Windows x64 (PE32+, MSVC C++17)
Encryption	RSA-2048 PKCS#1 (legacy CryptoAPI) wrapping ChaCha20 (DJB original 8-byte nonce, no MAC)
Sophistication (Operational)	**Moderate** — curated kill list for MSSQL/Hyper-V, Restart Manager file unlock, multi-thread pool with priority queue for high-value extensions, optional network-share-only mode
Sophistication (Cryptographic)	**Adequate but unauthenticated** — three-tier key hierarchy executed correctly (master / victim / per-file) but with no MAC over file content
Engineering quality	**Amateur** — 3 typos in recovery commands (silent failures), 2 broken kill-list entries, verbose debug logs leak class names + PDB path
Exfiltration	**None observed** — no socket/HTTP-POST/WinHTTP/Ws2_32 imports; the 72-hour leak claim in the note is **not backed by code in this build**
Recovery feasibility (operator-side)	Trivial — every encrypted file carries a self-contained 1544-byte footer recoverable with the master private RSA-2048 key
Recovery feasibility (victim-side)	**Zero** — without the master private key, neither inline footer nor registry blob is reachable
Sample SHA-256	`86b4d075d5bd0c49cbb21fd43935789b6612a2165273cc158dd0607b68941d04`

Notable Observations

1. Configuration is encrypted with plaintext key

The 3119-byte JSON configuration (PE resource id 101, type `SETTINGS`) is ChaCha20-encrypted using a key stored as a **plaintext ASCII string** in `.rdata`:

```
Key   = "TRUMPTRUMPTRUMPTRUMPTRUMPTRUMPTRUMP"[:32]
Nonce = "PUTLERPUTLER"[:8]
```

Both strings are visible to `strings | grep` and are passed by reference (not copied) at `0x14000511D` (`Context::LoadSettings`). Any defender intercepting the binary can recover the full configuration — including the master public key, the Tor onion, the Tox ID, the email contact, and the verbatim ransom note — without running the binary.

2. Recovery-inhibition shell list contains 3 typos that silently fail

Command (verbatim from `preRunCommands`)	Status
`vssadmin.exe Delete Shadows /All /Quiet`	✓ effective
`wbadmin delete backup -keepVersion:0 -quiet`	✓ effective
`wbadmin DELETE SYSTEMSTATEBACKUP`	✓ effective
`wbadmin DELETE SYSTEMSTABACKUP -deleteOldest`	✗ typo (`SYSTEMSTABACKUP`) — wbadmin error, no-op
`wmic.exe SHADOWCOPY /nointeractive`	✗ missing `delete` verb — listing only, no destruction
`bcdedit.exe /set {default} recoverynabled No`	✗ typo (`recoverynabled`) — bcdedit error, no change
`bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures`	✓ effective

Net effect: VSS deletion and bcdedit boot-status override take effect; system-state backup deletion and Recovery Environment disablement do not. Backups created with `wbadmin` SystemStateBackup remain intact post-attack.

3. Process kill list contains 2 broken entries — PostgreSQL and SQLWriter survive

```
taskkill -f -im sql writer.exe       ← unquoted name with embedded space; taskkill rejects
taskkill -f -impostgres.exe          ← '-im' joined with 'postgres.exe'; taskkill rejects
```

PostgreSQL (`postgres.exe`) and SQLWriter (`sqlwriter.exe`) **continue running** through the encryption pass and continue to hold open file handles. Restart Manager (`Util_RestartManagerUnlock`) compensates partially by force-releasing handles before per-file open, but the kill list bug means these processes can keep writing during the encryption window.

4. Verbose debug logs leak internals

Every error path logs the **fully-qualified C++ class::method name** to stdout:

```
Crypto::RSA::GenerateKeys, Crypto::RSA::Encrypt, Context::InitializeKeys,
Context::InitializePCInfo, Context::LoadSettings, EncryptFile, CryptFile,
Utils::GetResource, Utils::GetFileSize, CreateAndSetBackgroundImage,
CreateNote
```

Combined with the embedded PDB path:

```
J:\edu\niggerProject\bin\Publish\locker_win_x64_encrypter.pdb
```

Static reverse engineering, signature generation, and behaviour fingerprinting are all simplified.

5. The exfiltration claim in the ransom note is not backed by code

The note says:

> We've also retrieved files from your servers, which will be published in 72 hours if you don't contact us.

But the binary contains:

- No `WS2_32`, `WinHTTP`, `WinINet`, `WinSock` imports
- No socket / connect / send / SSL / TLS code path
- No FTP/SFTP/SMB-upload / rclone / archive-and-stage logic
- No HTTPS POST, no WebDAV, no MEGA/Anonfiles/etc. URL anywhere

The only outbound HTTP call in the entire binary is `URLDownloadToFileW("https://api.ipify.org", …)` — a one-shot **GET** to fetch the victim's external IPv4 address for the per-victim ID. There is no code path that writes victim data to the network. The leak threat is **bluff in this build**.

6. PPID-spoofed self-relaunch breaks parent-child telemetry

The binary uses `InitializeProcThreadAttributeList` + `UpdateProcThreadAttribute(PROC_THREAD_ATTRIBUTE_PARENT_PROCESS=0x20000)` to spawn a child `cmd.exe` whose recorded parent in EDR/Sysmon is **`explorer.exe`** (located via `GetShellWindow()` + `GetWindowThreadProcessId`) instead of the malware itself. The single use of this routine is for the **self-relaunch** with `-network -skip_misc` — the admin process forks a copy of itself to encrypt mapped network drives, with the child appearing as if it were spawned by the user from explorer.

Detection implication: Sysmon Event ID 1 (process create) for the network-drive child will report `ParentImage=C:\Windows\explorer.exe` and `ParentCommandLine=` unrelated to the malware. EDR rules that pivot on parent-child chain (e.g. `cmd.exe → vssadmin.exe` or `<unsigned exe> → cmd.exe`) miss this branch entirely. The mitigating data point is the child's `Image` (full path of the running malware) and its `CommandLine` (which contains `-network -skip_misc` — a unique signature substring).

This is **T1134.004 Access Token Manipulation: Parent PID Spoofing** — the only token-manipulation technique present in this build.

7. Per-victim ID is reproducible from machine fingerprint

`[IDENTIFIER]` in the ransom note is `base64(RSA(masterPubkey, PCInfo_blob))` where `PCInfo_blob` is a plaintext UTF-8 multi-line string built by `Util_GetPCInfo` (`0x140033040`):

```
IP: <api.ipify.org response>
PC-Name: <GetComputerNameA>
Domain: <GetComputerNameExA(ComputerNameDnsDomain)>
CPU: <HKLM\HARDWARE\DESCRIPTION\System\CentralProcessor\0\ProcessorNameString>
RAM: <GlobalMemoryStatusEx.ullTotalPhys / 1048576> MB
Disks:
\\.\PhysicalDrive0
\\.\PhysicalDrive1
...
```

There is **no random salt and no timestamp**. RSA-2048 PKCS#1 v1.5 is non-deterministic (random padding) so the base64 ciphertext is fresh per run, but the underlying plaintext digest is reproducible per machine. The operator can therefore re-derive a victim's identity for triage purposes from public domain knowledge of the corporate IP allocation, hostname conventions, and rough hardware specs — and de-duplicate victims contacting from different IPs over time.

Sample Fingerprint

SHA-256	`86b4d075d5bd0c49cbb21fd43935789b6612a2165273cc158dd0607b68941d04`
MD5	`7ae9c8b779ce3114199f4fd6335f681d`
Size	752 640 bytes
Toolchain	MSVC v19 / Windows SDK, C++17, nlohmann::json v3.11.3
Image base	`0x140000000`
Cryptographic library	None (legacy CryptoAPI for RSA + custom inlined ChaCha20)
String obfuscation	None (only 1 ChaCha20-encrypted PE resource with plaintext key)
Mutex	None
Packer	None

Killchain at a Glance

```
[1]  Stage gate       None — runs unconditionally, no environment check, no mutex
[2]  Settings load    Decrypt PE resource 101 ('SETTINGS') with ChaCha20
                       (key="TRUMP…"x7[:32], nonce="PUTLERPU"); parse JSON
[3]  Skip-path seed   Append CSIDL APPDATA / LOCAL_APPDATA / COMMON_APPDATA /
                       WINDOWS / SYSTEM (via SHGetFolderPathW) to runtime skipPaths
[4]  Key persistence  HKLM\SOFTWARE\PAIDMEMES\{PUBLIC,PRIVATE}; if absent, generate
                       fresh victim RSA-2048 keypair, RSA-wrap private with master
                       pubkey (5×256=1280 B), save base64 in registry
[5]  Victim ID gen    Util_GetPCInfo:
                         IP:  <api.ipify.org GET>
                         PC-Name: <GetComputerNameA>
                         Domain: <GetComputerNameExA(DnsDomain)>
                         CPU: <HKLM\HARDWARE\…\ProcessorNameString>
                         RAM: <GlobalMemoryStatusEx ullTotalPhys MB>
                         Disks: <\\.\PhysicalDrive0..N>
                        → RSA-wrap with master pubkey → base64
                        → string-replace [IDENTIFIER] in note template
[6]  Hide UI          ShowWindow(GetConsoleWindow(), SW_HIDE) if hideConsole=true
[7]  Hotkey thread    VK_INSERT toggle visibility / VK_HOME restart explorer
                       (operator UX, not anti-analysis)
[8]  preRunCommands   30 shell commands wrapped via SysWOW64\cmd.exe → sysnative\cmd.exe
                       (WoW64 hop, no PPID spoof):
                         - 13 taskkill (2 broken: 'sql writer.exe', '-impostgres.exe')
                         - 7 net stop MSSQL service variants
                         - vssadmin delete shadows  (effective)
                         - 3 wbadmin delete  (1 typo SYSTEMSTABACKUP, 2 effective)
                         - wmic shadowcopy  (broken — missing 'delete' verb)
                         - 2 bcdedit  (1 typo recoverynabled, 1 effective)
[9]  Volume reach     FindFirstVolumeW / FindNextVolumeW over all volumes including
                       \\?\Volume{GUID}\ entries; QueryDosDeviceW validates;
                       SetVolumeMountPointW mounts unmounted ones at temp letters
                       → reaches EFI / recovery / BitLocker-suspended partitions
[10] Privilege check  OpenProcessToken(TOKEN_QUERY) + GetTokenInformation(TokenElevation)
                       → r15 = TokenIsElevated (NOT informational — gates target)
[11] Target selection
       elevated  → Drive_EnumCipher (GetLogicalDrives + filter REMOVABLE/FIXED/RAMDISK)
                    + spawn child <own_path> -network -skip_misc with PPID=explorer.exe
                      (Util_RunProcess via UpdateProcThreadAttribute(PARENT_PROCESS))
       not elev  → Drive_EnumNetworkConnections (WNetGetConnectionW mapped drives)
[12] CreateNote       WHATS_HAPPEND.txt at every drive root and every directory
[13] DirWalker        Recursive FindFirstFileW; skip if StrStrIW(path, skipPath) matches
[14] EncryptFile      Per file:
                       - CreateFileW (with Restart Manager fallback for locked files)
                       - 40 random bytes (32 key + 8 nonce) via CryptGenRandom
                       - RSA-encrypt with victim public key → 256 B
                       - ChaCha20 XOR over Range A [0, min(size, 601 MiB)] +
                         Range B [size−601 MiB, size−265 MiB] if size > 1.2 GiB
                       - Append 1544-byte footer (1280 + 256 + 8)
                       - MoveFileW → <orig>.BAVACAI
[15] Pool drain       Wait for 32-thread encryption pool to flush
[16] Recycle empty    SHEmptyRecycleBinW per drive (removeRecycle=true)
[17] Wallpaper        SystemParametersInfoW(SPI_SETDESKWALLPAPER)
                       — disabled in this config (backgroundImage=false)
[18] postRunCommands  Empty in this config
[19] Note open        ShellExecute on note (openRequirementsOnFinish=true)
```

No self-deletion. The binary remains on disk after run.

Lateral Movement

None. BAVACAI does not propagate. There is no:

- ADSI / LDAP enumeration
- SMB share enumeration (`NetShareEnum` , `WNetEnumResource` — only `WNetGetConnectionW` for already-mapped drives)
- WMI / WinRM / DCOM / PsExec / scheduled-task / service-install primitive
- Python/PowerShell/CMD lateral runner
- USB / removable media replication
- Embedded credential dumper / Mimikatz / SMB credential testing

The `-network` CLI flag merely scopes encryption to drives already mapped by the user (`WNetGetConnectionW`); it does not establish new SMB connections, does not authenticate, does not enumerate shares.

Detection & Response

Priority IOCs

Type	Value
File extension	`.BAVACAI` (uppercase, appended after original)
Ransom note	`WHATS_HAPPEND.txt` (typo "HAPPEND")
Footer signature	last 8 bytes = `06 00 00 00 00 00 00 00` (constant per file)
Footer size	1544 bytes appended at end of encrypted file
Tor file server	`t33zoj4qwv455fog7qnb2azi5xcdxkixughmmduzbw2rtdgryqfbh6id.onion`
Email	`nhuvgh@outlook.com`
Tox (qtox) ID	`7C564920870C0D33535D2012ECDDE389FE25BAF7AF427DD584EE39C04AF8CF024F8BFA93D8DB`
Outbound HTTP	`https://api.ipify.org` (single GET per run)
Registry root	`HKLM\SOFTWARE\PAIDMEMES`
Registry values	`PUBLIC` , `PRIVATE` (REG_BINARY base64) — only two values written
External-IP file	`<11-char-name>.crypt` in working dir (transient artefact)
PDB path	`J:\edu\niggerProject\bin\Publish\locker_win_x64_encrypter.pdb`
Distinctive `.rdata` strings	`TRUMPTRUMPTRUMPTRUMPTRUMPTRUMPTRUMP` , `PUTLERPUTLER` , `SOFTWARE\PAIDMEMES` , `[IDENTIFIER]` , `chiperDrives` , `skipPathes`

Critical behavioural pattern: admin/non-admin asymmetric encryption

The privilege check at `0x140020aec` does not gate execution but **selects the encryption target**:

Token state	Encryption scope	Drive enumerator
Elevated (admin)	Local drives — REMOVABLE (USB), FIXED (HDD/SSD), RAMDISK	`Drive_EnumCipher` (`0x140039F30`)
Non-elevated	Mapped network drives only	`Drive_EnumNetworkConnections` (`0x14003A160` , via `WNetGetConnectionW`)

When admin, the binary additionally **self-relaunches** with `-network -skip_misc` and **PPID-spoofed parent = explorer.exe** (see notable observation #6) so that network drives are processed in parallel by a child process that is not visibly a descendant of the admin malware.

Beyond `GetLogicalDrives()` , the binary also enumerates **non-lettered volumes** via `FindFirstVolumeW` / `FindNextVolumeW` and mounts them at temporary letters using `SetVolumeMountPointW` — covering hidden recovery partitions, EFI System Partitions, and BitLocker-suspended volumes that are not visible to standard tools.

High-signal behaviours

- `vssadmin.exe Delete Shadows /All /Quiet` from a non-administrative parent (cmd.exe)
- `bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures` from cmd.exe
- `bcdedit.exe /set {default} recoverynabled No` (note the **typo** in the command line — pivot indicator)
- `wbadmin DELETE SYSTEMSTABACKUP -deleteOldest` (note the **typo** `SYSTEMSTABACKUP` — pivot indicator)
- `wmic.exe SHADOWCOPY /nointeractive` without verb — pivot indicator
- 30 child `cmd.exe` processes spawned within seconds of parent process start
- `taskkill -f -im` sweep of: `sqlbrowser.exe` , `sqlserv.exe` , `msmdsrv.exe` , `MsDtsSrvr.exe` , `sqlceip.exe` , `fdlauncher.exe` , `Ssms.exe` , `SQLAGENT.EXE` , `fdhost.exe` , `ReportingServicesService.exe` , `msftesql.exe` , `pg_ctl.exe`
- `taskkill -f -impostgres.exe` (malformed — pivot indicator)
- `net stop` of: `MSSQLServerADHelper100` , `MSSQL$ISARS` , `MSSQL$MSFW` , `SQLAgent$ISARS` , `SQLAgent$MSFW` , `SQLBrowser` , `REportServer$ISARS` , `SQLWriter`
- `RegCreateKeyExW HKLM\SOFTWARE\PAIDMEMES` followed by `RegSetValueExW` with name `PUBLIC` or `PRIVATE` (REG_BINARY)
- `URLDownloadToFileW` to `https://api.ipify.org` from a process that subsequently writes encrypted files
- Two related processes with **same Image path** running concurrently, one with a normal command line and one with `-network -skip_misc` , where the second has `ParentImage=explorer.exe` (PPID-spoofed self-relaunch)
- `FindFirstVolumeW` followed by `SetVolumeMountPointW` from the same process — mount-point manipulation to reach hidden volumes
- `RegGetValueW` against `HKLM\SOFTWARE\PAIDMEMES\PUBLIC` and `\PRIVATE` (REG_BINARY)
- File rename storm: `MoveFileW` from `<original>` to `<original>.BAVACAI` at high cadence (32 concurrent threads default)
- Restart Manager session opened (`RmStartSession` → `RmShutdown(RmForceShutdown=4)`) to release locked files — visible as `RestartManager` ETW provider events
- `SHEmptyRecycleBinW` invoked on every drive at end of run

Hunting queries (Sigma-style sketch)

```
title: BAVACAI — bcdedit Recovery Disable Typo
description: Detects the unique BAVACAI typo "recoverynabled" in bcdedit
detection:
  selection:
    EventID: 1                  # Sysmon ProcessCreate
    Image|endswith: '\bcdedit.exe'
    CommandLine|contains: 'recoverynabled'
  condition: selection
```

```
title: BAVACAI — Registry PAIDMEMES Persistence
detection:
  selection:
    EventID: 13                 # Sysmon RegistryEvent
    TargetObject|contains: '\SOFTWARE\PAIDMEMES\'
  condition: selection
```

```
title: BAVACAI — Wbadmin Typo SYSTEMSTABACKUP
detection:
  selection:
    EventID: 1
    Image|endswith: '\wbadmin.exe'
    CommandLine|contains: 'SYSTEMSTABACKUP'
  condition: selection
```

```
title: BAVACAI — File Extension Mass Rename
detection:
  selection:
    EventID: 11                 # Sysmon FileCreate
    TargetFilename|endswith: '.BAVACAI'
  timeframe: 5m
  condition: selection | count() > 100
```

```
title: BAVACAI — api.ipify.org from Encryptor Process
detection:
  selection:
    EventID: 22                 # Sysmon DnsQuery
    QueryName: 'api.ipify.org'
  filter:
    Image|endswith:
      - '\firefox.exe'
      - '\chrome.exe'
      - '\edge.exe'
      - '\powershell.exe'
  condition: selection and not filter
```

```
title: BAVACAI — PPID-Spoofed Self-Relaunch
description: Detects the admin → network-drive child fork with PPID spoofed to explorer.exe
detection:
  selection:
    EventID: 1                  # Sysmon ProcessCreate
    ParentImage|endswith: '\explorer.exe'
    CommandLine|contains:
      - ' -network -skip_misc'
      - ' -skip_misc -network'
  condition: selection
```

Response priorities

1. **Containment**: isolate hosts where `.BAVACAI` files appear or where `HKLM\SOFTWARE\PAIDMEMES` exists. The binary does not propagate, so containment is host-local — no AD-wide hunt needed for lateral artefacts.
2. **Backup verification**: System State Backup (`wbadmin SystemStateBackup`) is **not destroyed** in this build (typo); shadow copies and `wbadmin delete backup` are. Check `wbadmin get versions` and image-level backups before payment consideration.
3. **Process kill list survivors**: PostgreSQL (`postgres.exe`) and SQLWriter (`sqlwriter.exe`) survive `preRunCommands` due to malformed taskkill arguments — encrypted files written by these processes during the run may be partially recoverable from PostgreSQL WAL or SQL Server's Volume Shadow Copy if VSS deletion failed.
4. **Registry forensics**: `HKLM\SOFTWARE\PAIDMEMES\PRIVATE` contains the victim's RSA private key encrypted with the master public key (1280 B base64). Preserve this value alongside disk image. If the master private key ever surfaces (operator arrest, leak, dump), this single value enables decryption of every encrypted file on the system without needing per-file footers.
5. **Memory acquisition**: pointless. The victim's RSA private key is not held in memory after decryption — it is stored on disk (registry) and not re-imported until next run. Memory carving will not yield additional keying material.
6. **Do not pay**: payment relies on the operator's discretion. The cryptographic protocol does allow operator-side decryption (every file's footer is self-contained) but: (a) the leak threat is bluff in this build — there is no exfiltration to leverage, (b) the operator's infrastructure is single-handle (one Tor server, one outlook.com mailbox, one Tox) suggesting a small actor with no consistent reputation track record, (c) the embedded ID `nigger…` in the PDB path raises material reputational risk for any organisation paying.
7. **Decrypt without payment**: only feasible if the master private RSA-2048 key is leaked or recovered from operator infrastructure. Until then, files are non-recoverable in practice.
8. **Hunt for staged-but-not-yet-encrypted hosts**: machines where `HKLM\SOFTWARE\PAIDMEMES\PUBLIC` and `\PRIVATE` exist but no `.BAVACAI` files are present yet may be **staged with keys generated but encryption pending or interrupted** — kill the parent process and remove the registry keys before any subsequent run can resume with the same victim keypair.

Threat Positioning

BAVACAI reads as the work of a **single developer** producing a **commodity-grade Windows ransomware** with curated targeting for **MSSQL / Hyper-V / SQL Server estate** environments and no aspiration to lateral movement. The operational tradecraft visible in the configuration — `threadPoolPriorityExtensions = [".sql", ".bak", ".VHDX"]`, kill list specifically covering MSSQL Forefront/ISA Server instance names (`ISARS`, `MSFW`), Restart Manager file unlock — suggests deliberate selection of mid-market backup/database targets. The lack of AD enumeration, SMB lateral, and exfiltration suggests the operator deploys via **interactive RDP** or **manual hands-on-keyboard staging** after initial access, not through a worm-like loader.

The engineering quality is **markedly amateur**:

- The PDB path `J:\edu\niggerProject\bin\Publish\` exposes both the developer's drive layout, an internal slur as folder name, and the stage of the build pipeline (`Publish` = MSVC release publish target — typical of solo MSVC users on a personal machine, not a hardened build server).
- Three of the seven recovery-inhibition shell commands have typos that silently fail. The typos are recognisable as English-non-native (`SYSTEMSTABACKUP`, `recoverynabled`) and make excellent pivot indicators for hunting.
- Two of the thirteen `taskkill` entries are syntactically invalid (`sql writer.exe` with an unquoted space; `-impostgres.exe` with the switch and target glued).
- JSON config keys contain consistent typos (`skipPathes` instead of `skipPaths`; `chiperDrives` instead of `cipherDrives`).
- Verbose debug log lines leak class names and a Putin/Hitler portmanteau (`PUTLERPUTLER`) and Trump-themed marker (`TRUMPTRUMP…`) that are clearly the developer's idea of in-jokes — not operational steganography.
- The registry root `SOFTWARE\PAIDMEMES` is itself an in-joke, not an obfuscated tradecraft choice.

The cryptographic engineering, in contrast, is **competent if textbook**: a three-tier key hierarchy (master / victim / per-file), a self-sufficient per-file recovery envelope, RSA-2048 PKCS#1 v1.5 (not vulnerable in this single-block-per-message usage), ChaCha20 with proper sigma constant, no key-reuse, no nonce-reuse, no IV management bugs. The only crypto weakness is **lack of authentication** — files can be tampered with and the malware will not detect it — but that has no bearing on victim recovery.

The mismatch between **clean cryptographic theory** and **leaked debug strings + typos throughout** is consistent with a developer who learned crypto from a textbook (`expand 32-byte k` ChaCha20 sigma reference is canonical Bernstein) but learned shell scripting from forum copy-paste (the wbadmin/bcdedit typos are easy-to-introduce when transcribing from a CMD prompt). This is **not a state-aligned actor** and **not an established RaaS affiliate** — it reads as a solo developer iteration of a commodity locker, possibly an **affiliate of a larger family** distributing a personalised build.

MITRE ATT&CK

ID	Technique
T1486	Data Encrypted for Impact
T1490	Inhibit System Recovery (`vssadmin Delete Shadows` , `wbadmin delete backup` , `bcdedit bootstatuspolicy ignoreallfailures`)
T1489	Service Stop (`net stop` of MSSQL service variants from `preRunCommands`)
T1057	Process Discovery (implicit via fixed `taskkill -f -im` list, no enumeration)
T1083	File and Directory Discovery (`FindFirstFileW` recursion in `DirWalker`)
T1135	Network Share Discovery (`WNetGetConnectionW` only — already-mapped drives)
T1071.001	Application Layer Protocol: Web (single GET to `api.ipify.org`)
T1480.001	Execution Guardrails: Environmental Keying (external IP in victim ID)
T1112	Modify Registry (`HKLM\SOFTWARE\PAIDMEMES` for victim key persistence)
T1027.013	Encrypted/Encoded Files: Encrypted Resources (ChaCha20 over PE resource id 101)
T1485	Data Destruction (`SHEmptyRecycleBinW` per drive)
T1564.003	Hide Artifacts: Hidden Window (`ShowWindow(SW_HIDE)` console)
T1059.003	Command and Scripting Interpreter: Windows Cmd (`preRunCommands` shell-out)
T1059	Command and Scripting Interpreter (Restart Manager via `RstrtMgr` API — out-of-band file unlock)
T1134.004	Access Token Manipulation: Parent PID Spoofing (`UpdateProcThreadAttribute(PROC_THREAD_ATTRIBUTE_PARENT_PROCESS)` with `explorer.exe` for self-relaunched network-drive child)
T1082	System Information Discovery (hostname, DNS domain, CPU model, RAM, physical disks)
T1016	System Network Configuration Discovery (`api.ipify.org` external IP)

Notably absent: T1561. disk-structure-wipe (no MBR/UEFI overwrite, no MFT destruction), T1547. autostart persistence (no Run keys, no Winlogon, no SafeBoot), T1543.003 service install, T1021. lateral movement (no SMB / WMI / WinRM / DCOM / PsExec), T1091 USB replication, T1499. DDoS, T1056.001 keylogging, T1562.001 disable security tools (no Defender disable, no AMSI bypass, no ETW patch). T1485 (Data Destruction) is present only as recycle-bin emptying at end of run, not as bulk data wiping.

Verdict

Axis	Score
Operational tradecraft	Moderate — curated for MSSQL/Hyper-V, no lateral
Cryptographic engineering	Competent textbook execution, no MAC
Exfiltration capability	**Absent** (note claim is bluff)
Lateral movement capability	**None** (host-local only)
Persistence robustness	Low — registry-only, no service / Run / autostart
Recovery feasibility for the victim	**Zero** without master private key
Recovery feasibility for the operator	Trivial (footer self-contained)
Threat to enterprise environments	Moderate — destructive but contained per host
Distinctiveness for hunting	High (typos, registry root, footer signature)

BAVACAI should be treated as a **single-host destructive ransomware event** with no propagation risk and no exfiltration component in this build. The leak threat in the ransom note is unsupported by the binary's capabilities. The technical robustness of the cryptographic envelope means files are non-recoverable in practice without operator cooperation, but the operator-side recovery protocol does function correctly — there is no architectural impossibility to decryption (unlike VECT 2.0's lost-nonce flaw or IronChain 3.0's discarded-shift transformation).

The high concentration of typos, debug log leakage, and in-jokes (`PAIDMEMES` , `TRUMPTRUMP` , `PUTLERPUTLER` , the slur in the PDB path) makes BAVACAI **trivially fingerprintable** for static and behavioural detection. The five-second hunting sketch built around `recoverynabled` , `SYSTEMSTABACKUP` , and `HKLM\SOFTWARE\PAIDMEMES` should achieve high-precision identification of any deployment of this exact build.

Organisations should prioritise containment and offline backup recovery; payment is not an operationally sound path given the operator's single-handle infrastructure and the absence of any reputation-management mechanism.