Everest

Technical Analysis — Windows

REVERSE-ENGINEERED REPORT

RansomLook · ransomlook.io
File last modified: 2026-05-15

Sample SHA-256: `1df92bf4c967297d8a39fc3f619a56702ee96d5cf9196b8e1d5b3654746c6514`

EVEREST Ransomware — Full Analysis

1. Sample Identification

Field	Value
Family	EVEREST
SHA-256	`1df92bf4c967297d8a39fc3f619a56702ee96d5cf9196b8e1d5b3654746c6514`
MD5	`0dd70c334507188714bae0af7229b379`
Type	PE32 .NET assembly (Intel i386), Windows console
Size	116 224 bytes (114 KB)
Language	C# / .NET Framework 4.0.30319
Original name	`hlntqyun.exe`
Compile timestamp	2022-07-02T18:04:11 UTC (raw `0x62C0889B`) — .NET timestamps are unreliable, treat as suspect
Image base	`0x00400000`
Sections	3 — `.text` (110 KB, code+IL), `.rsrc` (1 KB), `.reloc` (12 B)
TypeDefs	58 (all in global namespace)
MethodDefs	398
EntryPoint	`iWvpiwZPYibt::QXD6VTHE0TXI` (md=224, RVA `0x00004584`)
Note salutation	Hard-coded victim name in `<Module>.smethod_5` (decoded ransom note begins with "Dear <name>,")

The sample is a **fully-managed .NET 4.0 ransomware** protected with **ConfuserEx 1.x** (or compatible fork). DIE heuristic flags `Modified managed EP + Int confusion + Short names + Bad .cctor format`. The original `ConfusedByAttribute` watermark has been stripped, but the IL fingerprint (Int confusion via `Math.Cos` / `Math.Sin` / `Math.Sqrt` arithmetic, anonymous renamed types, modified module `.cctor`, encrypted UserStrings heap) is unambiguous.

The sample was deobfuscated with **de4dot-cex** (multiple successive passes) producing `sample-cleaned-final.exe` (94 208 bytes, SHA-256 `b30b7911b256daeceb631dec923f83c0ef17f34eb46452d5ab32aab2540be361`). Symbol renaming is not reversible (ConfuserEx does not retain a mapping), but strings and control-flow are restored. ConfuserEx's `constants` preset encrypts the UserString heap; runtime resolution happens via `<Module>.m(string)` (a GZip+Base64+lookup decryptor) called by ~210 zero-arg getter methods. Static decryption was achieved by loading the cleaned assembly via .NET reflection (`Assembly.Load(byte[])`) and invoking `<Module>.m()` on each getter's return value (`Object` boxing the index string), producing a complete `mdtoken → cleartext` mapping (214 strings).

For the analysis pass, the cleaned binary was processed with a private tool which (1) decrypts the strings via reflection, (2) rewrites the IL of every call-site to inline the cleartext as `ldstr` literals, eliminating all `<Module>.m(<Module>.smethod_X())` indirections, and (3) renames classes / methods / fields from a per-family JSON mapping. The `EverestRansomware.exe` file referenced in this report is the output of this pipeline.

Imports / Referenced Assemblies (15)

DLL / Assembly	Purpose
`mscorlib`, `System`, `System.Core`	runtime, LINQ, parallel tasks
`System.Xml`, `System.Xaml`, `WindowsBase`, `PresentationCore`, `PresentationFramework`	unused (dnSpy artifact)
`System.ServiceProcess`	ServiceController (stop AV/backup services)
`System.Management`	WMI (Win32_Share, Win32_NetworkAdapterConfiguration, Win32_LogicalDisk, Win32_Processor, Win32_OperatingSystem, Win32_NetworkConnection, Win32_MappedLogicalDisk, Win32_ShadowCopy)
`System.Windows.Forms`, `System.Drawing`	wallpaper bitmap generation, NotifyIcon balloon
`kernel32.dll` (P/Invoke)	LoadLibrary, GetProcAddress
`user32.dll` (P/Invoke)	SystemParametersInfo (wallpaper)

Native APIs are resolved **dynamically** via `LoadLibrary` / `GetProcAddress` (class `zPyJQYTgymZw`) — the only declared P/Invokes in the assembly are `kernel32!LoadLibrary`, `kernel32!GetProcAddress`, `user32!SystemParametersInfo`. The runtime-resolved DLLs are: `kernel32`, `rstrtmgr`, `advapi32`, `Netapi32.dll`, `Mpr.dll`, `Srclient`. This dynamic resolution evades static IAT-based detection.

---

2. Infrastructure

Field	Value
Onion blog	`ransomocmou6mnbquqz44ewosbkjk3o5qjsl3orawojexfook2j7esad.onion`
Email	`everestaf@onionmail.org`
Twitter	mentioned in note ("Or read about our group in Twitter") — handle not embedded
Note filename	`EVERESTRANSOMWARE.txt`
Extension	`.everest`
Mutex	`Global\7efc73f7-fda1-42d1-a4c5-8f1670bd08a5` (single-instance)
Wallpaper file	`%TEMP%\[WALLNAME].bmp` (literal `[WALLNAME]` placeholder — apparent build-template residue)
Balloon title	"All your files are encrypted, please contact with us!"
Balloon text	"Find note file on desktop!"
User-Agent string (embedded, dead code)	`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134` — present in `<Module>.LeMnn5Q3tdTG` and used by `Helpers.DownloadUrl` (`WebClient.DownloadData`). `Helpers.DownloadUrl` is **only called by itself** (retry recursion); no other method in this binary invokes it.
Registry persistence (victim ID)	`HKCU\Software\AppName` ← stores RSA-wrapped seed under value `PublicKey`
Payment	Negotiation only (no automated payment portal mentioned)

The decoded note string in `<Module>.smethod_5` begins with a "Dear <name>," salutation where the name is hard-coded in the binary. The presence of a literal name in the encrypted strings heap, alongside an embedded RSA-1024 public key (see §5), means this binary was assembled with content specific to a single deployment.

---

3. Ransom Note

Filename and Drop Locations

`EVERESTRANSOMWARE.txt` is dropped twice: 1. Primary: `<CommonApplicationData>\EVERESTRANSOMWARE.txt` (= `C:\ProgramData\EVERESTRANSOMWARE.txt`) 2. Backup: `<Desktop>\EVERESTRANSOMWARE.txt` (only if not already present) 3. Per-directory: a copy is dropped in every directory traversed during encryption (skipping `\Users\<name>\Desktop`)

After encryption is complete, `Process.Start("notepad.exe", note_path)` opens the desktop copy.

Content (extracted by reflection-based string resolver, `<Module>.smethod_5` decoded; victim salutation redacted)

```
Dear [REDACTED_VICTIM],
Greetings from the Everest team. Your systems have been attacked, the files are encrypted. You
can read about us in our blog (Tor browser needed)
Blog : ransomocmou6mnbquqz44ewosbkjk3o5qjsl3orawojexfook2j7esad.onion
Or read about our group in Twitter
Also, our team was able to bypass your "Dataprotection" as any other your protection software and
more than 1 Terabyte of internal files were exfiltrated to our servers, which we can confirm with
great joy and ease
The list contains financial documents, internal orders, KYC information(documents,photos...),
trusted representatives personal info
Client risk levels,loans, debt and client data. Various financial documentation, backups , etc.
etc.
The information was collected both from personal PCs and from centralized storage locations.
If an agreement is reached with us, this information will never be published and the problem will
disappear as if it never happened, otherwise it will be posted on our blog and darknet. Which
will lead to even greater financial and reputational losses on your part.
Also you will get
1.Attack logbook (months of experience with your company) with full list of vulnerabilities and
bypass methods
2.Advices how to singifically improve your security and avoid such attacks in the future
3.We will delete all files from your company
4.We will attack your company no more
Yours trully Everest Team
Email to contact: everestaf@onionmail.org
```

Note structure at runtime

The dropped note appends the per-victim RSA-wrapped seed under a `Your key:` marker:

```
<base64 ransom note text>\r\n
Your key: \r\n
<base64-encoded RSA-1024-encrypted seed>\r\n
```

Notable typos (signature artifacts): **"trully"** (instead of "truly"), **"singifically"** (instead of "significantly"), spaced comma "backups , etc.". These are stable across Everest campaigns.

---

4. Execution Flow (`iWvpiwZPYibt::QXD6VTHE0TXI` @ RVA `0x00004584`)

```
1.  Acquire single-instance Mutex Global\7efc73f7-fda1-42d1-a4c5-8f1670bd08a5
    If already held → kill self via jYiNxYzr4HjQ()  (Process.Kill | Environment.Exit(0))

2.  LoadLibrary("kernel32") → cache handle in zPyJQYTgymZw.intptr_0

3.  CIS-exclusion geo-fencing
    - Compare InstalledUICulture.Name AND CurrentCulture.Name (lower) against:
        hy-AM, az-Cyrl-AZ, Cy-az-AZ, Lt-az-AZ, be-BY, kk-KZ, ky-KZ (×2),
        tt-RU, ba-RU, sah-RU, ru-RU, tg-Cyrl-TJ, uz-Cyrl-UZ, uk-UA, ka-GE
    - Compare CurrentCulture.LCID against:
        1049 (RU-RU), 1067 (hy), 2092 (az-Cyrl), 1068 (az-Lat), 1059 (be),
        1079 (ka), 1087 (kk), 1064 (tg), 1090 (tt), 2115/1091 (uz), 1058 (uk)
    - On any match → exit silently

4.  Spawn 3 background threads (IsBackground=true)
    a. Thread iYqREGyyukhw.aVcAKp19tNAg (Normal priority)
       Loop 4s: kill RE/sniffer tools matching ProcessName or MainWindowTitle
       (dnspy, x64dbg, ida64, wireshark, fiddler, pe-sieve, MegaDumper, …)
    b. Thread avlgfDFtmIlr.PPXFI8z6qAGd (AboveNormal priority)
       - One-shot: sc.exe config <name> start=disabled for SQLTELEMETRY/SQLWriter/
         SstpSvc/MBAMService etc.; ServiceController.Stop for services starting
         with "sophos"
       - Loop 15s: Parallel.ForEach services → stop if name matches AV/backup
         list (~100 services) or contains "backup" / "sql"
       - Same loop: kill processes matching kill list (~50 procs)
    c. Thread wqqiwcJkIwbq.DjhJetlmaw3J (Normal priority)
       Loop 2.5s: kill any Process whose PrivateMemorySize64 > 250 MB
       unless name in {chrome, opera, msedge, firefox, iexplore, explorer,
       winlogon, SearchIndexer, wininit, SearchApp, SearchUI, powershell}

5.  Synchronous initialization helpers (executed on the main thread):
    - IdSMieJfrIOy.cB()       — Anti-Raccine (delete IFEO entries, kill Raccine
                                processes, schtasks /DELETE Raccine Rules Updater)
    - uxTwxHBatbAA.smethod_0()— SetThreadExecutionState(0x80000003) (anti-sleep)
    - KZsyzkgZlDQw.UB()       — Process self-protection: insert AccessDenied ACE
                                for WorldSid (Everyone) into own DACL
    - hyqyuDHMHPRC.aB()       — Backup file wipe: del /s /f /q on ~80 patterns
                                across drives c-h (*.VHD, *.bac, *.bak, *.wbcat,
                                *.bkf, Backup*.*, *.set, *.win, *.dsk)
    - jjOjKFMSDrCb.BB()       — VSS deletion: powershell Get-CimInstance
                                Win32_ShadowCopy | Remove-CimInstance
                                (vssadmin Delete Shadows /all /quiet on XP)
    - WHtZqsICMCns.dd()       — System Restore Points wipe: dyn-load
                                Srclient!SRRemoveRestorePoint, enum WMI
                                \\.\root\default\systemrestore, remove all

6.  OS-conditional Defender CFA bypass
    If iWvpiwZPYibt.gj8BnGv30aeJ.Contains("10")  // OS info from WMI Caption
       powershell.exe Set-MpPreference -EnableControlledFolderAccess Disabled (wait)

7.  Recycle bin destruction
    cmd.exe /c rd /s /q %SYSTEMDRIVE%\$Recycle.bin       (no wait)
    cmd.exe /c rd /s /q D:\$Recycle.bin                  (no wait)

8.  NYPSjWpoCyzg.T6m93N1SE77Z()
    - sc config Dnscache/fdPHost/FDResPub/SSDPSRV/upnphost start= auto
    - ServiceController.Start for those (network discovery prep)
```

```
 9.  Firewall + SMB1 enablement
     netsh advfirewall firewall set rule group="Network Discovery" new enable=Yes (wait)
     netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=Yes (wait)
     powershell Enable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol (no wait)

10.  XXGcXDwCBmxY.ub() + .CCjtuCBT7qzI()
     reg ADD HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
        /v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1
     reg ADD same path /v EnableLinkedConnections /t REG_DWORD /d 1
     reg ADD HKLM\SYSTEM\CurrentControlSet\Control\FileSystem
        /v LongPathsEnabled /t REG_DWORD /d 1

11.  NquuvOEOwNeN.uWL7cOwG50y5() — mount unlettered volumes
     - Enum free letters [A-Z] minus existing logical drives
     - mountvol.exe → parse stdout, take all "\\?\Volume{...}\" lines
     - mountvol.exe <free_letter>: <volume_guid>
     → mounts EFI/recovery/hidden partitions on free letters for encryption

12.  LcKDzxOgRJco.FB() — ACL grants
     icacls "C:*" /grant Everyone:F /T /C /Q
     icacls "D:*" /grant Everyone:F /T /C /Q
     icacls "Z:*" /grant Everyone:F /T /C /Q

13.  fscjvXdLHuJt.rc() — LAN propagation engine (see §10)
     - Wake-On-LAN every ARP'ed host (UDP magic packet to broadcast :7,:9)
     - net view + NetDfsEnum + WNet recursive enum + WMI Win32_Share + WMI
       Win32_NetworkConnection/MappedLogicalDisk
     - net.exe use \\<unc> for each discovered share
     - Recursive directory count for dedup
     - Append all targets to uJBRudfoWFSi.jBZdLoBjJEgb
     - Save target list to C:\ProgramData\<MD5(ProcessorId+VolumeSerial)>

14.  Generate per-victim seed and wrap with RSA
     o3mpYXKn8zl6 = XXGcXDwCBmxY.Vb()             // 32 ASCII printable bytes
                                                    // via System.Random (NOT RNG-secure)
     lw4n5k9vFpSx = ACWpKFzdfHch.smethod_0(        // RSA-1024-PKCS1v1.5 encrypt
        ASCII.GetString(o3mpYXKn8zl6))             //   then Base64 encode

15.  Persistence registry write
     reg ADD HKCU\Software\AppName /v PublicKey /d <RSA-wrapped-seed-base64>

16.  Build ransom note string
     Fb = ransom_note_text + NewLine
     Fb = Fb + "Your key: " + NewLine + lw4n5k9vFpSx + NewLine

17.  Drop note (primary + backup)
     hb(C:\ProgramData\EVERESTRANSOMWARE.txt, Fb)
     If not exists: hb(<Desktop>\EVERESTRANSOMWARE.txt, Fb)

18.  uJBRudfoWFSi.Jb() — encryption main (see §5)
     - Derive AES-128 Key + IV via PBKDF2(seed, salt=\x01..\x08, iter=1000)
     - Parallel.ForEach drives: walk → encrypt every eligible file
     - Per-directory note drop

19.  XXGcXDwCBmxY.OwMvDWEPTi4W(title, text) — NotifyIcon balloon tip
     "All your files are encrypted, please contact with us!" / "Find note file on desktop!"
```

```
20.  QmTYuuYAWWkY.qNvEjwNtgDZ3() — wallpaper change
     - Generate fullscreen .bmp (DarkBlue, white Tahoma 16pt) with full note
     - Save to %TEMP%\[WALLNAME].bmp
     - SystemParametersInfo(SPI_SETDESKWALLPAPER, 0, path, SPIF_UPDATEINIFILE|SPIF_SENDCHANGE)
     - reg HKCU\Control Panel\Desktop\WallPaper = path

21.  Process.Start("notepad.exe", <Desktop>\EVERESTRANSOMWARE.txt)

22.  XXGcXDwCBmxY.kahdFWq2bR0P() — self-delete
     cmd.exe /C ping 127.0.0.7 -n 3 > Nul
                & fsutil file setZeroData offset=0 length=524288 "<self_path>"
                & Del /f /q "<self_path>"
     Process.GetCurrentProcess().Kill()
```

Thread Architecture Summary

Thread	Loop	Job
Main	once	Setup → keygen → drop note → encryption → wallpaper → self-delete
`iYqREGyyukhw`	4 s	Anti-analysis killer (RE tools, sniffers)
`avlgfDFtmIlr`	15 s	Service stopper + process killer (AV/backup/DBs)
`wqqiwcJkIwbq`	2.5 s	Memory pig killer (>250 MB heap, except whitelist)

The 3 worker threads run **for the entire lifetime of the process**, killing newly-spawned defenders or analysis tools while the main thread encrypts.

---

5. Encryption System

Key Exchange

Parameter	Value
Algorithm	RSA, **PKCS#1 v1.5 padding** (`Encrypt(data, false)`)
Key size	**1 024 bits** (declared as 4096 in `RSACryptoServiceProvider(4096)` constructor, **but immediately overwritten** by `FromXmlString(...)` with the 128-byte modulus from `<Module>.smethod_19()`)
Implementation	`System.Security.Cryptography.RSACryptoServiceProvider`
Embedded attacker public key (XML format, base64-decoded from `xb`):	

```
<RSAKeyValue>
  <Modulus>uByCMcH5MwQ1wXW3pKGb4eP8YVb3+d0Vrg243VI2f9jj00hM4CmRilFu+CrnNo/kZ/eMLOW13T/
5RW1ono1c4uOtm6zDW/
S229nc8eDjIEyqw9A6McuoVYYdW+lkm3u15pYG2uZBrv3MnAkQRllguCxnvvv+VT8K5rAHemyqVfc=</Modulus>
  <Exponent>AQAB</Exponent>
</RSAKeyValue>
```

The Base64 modulus decodes to 128 bytes = 1024 bits. This is the public key embedded in this specific sample. Other samples in this family carry distinct keys.

Symmetric Cipher

Parameter	Value
Algorithm	AES
Mode	**CBC** (small files, full encryption) / **CBC NoPadding** (large files, intermittent)
Key size	**128 bits** (declared `KeySize=256` in object initializer, but overridden by `Key=16 bytes` setter)
Padding	PKCS#7 (full mode) / None (intermittent mode, blocks aligned 32 B)
IV size	16 bytes
Per-file key	**NO** — same Key+IV for every file in a single run
Provider	`System.Security.Cryptography.AesCryptoServiceProvider`

Key Derivation

```
// uJBRudfoWFSi.smethod_1(byte[] seed)
var rfc = new Rfc2898DeriveBytes(seed, salt = {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08},
                                 iter = 1000);
iWvpiwZPYibt.dafMgM4BRhQ8 = rfc.GetBytes(16);   // AES Key
iWvpiwZPYibt.JaoUr2NWRrUQ = rfc.GetBytes(16);   // AES IV
```

PBKDF2-HMAC-SHA1, **fixed 8-byte salt** `01 02 03 04 05 06 07 08`, **only 1000 iterations** — both are below modern crypto standards.

Per-Victim Seed Generation (`XXGcXDwCBmxY.Vb()`)

```
var sb = new SecureString();
for (int i = 0; i < 32; ) {
    char c = (char) random.Next(33, 127);   // [!] System.Random — not crypto-secure
    if (c != ' ' && c != '\\') {
        sb.AppendChar(c);
        i++;
    }
}
return Encoding.ASCII.GetBytes(sb.ToString());  // 32 ASCII printable bytes
```

The seed is **32 printable ASCII characters** (range 33–126, excluding space and backslash) generated by the **`System.Random`** PRNG, which is **NOT cryptographically secure**. Seeded by default from `Environment.TickCount` (low-entropy boot time wall clock).

Crypto Stack Weaknesses (recap)

#	Weakness	Severity
1	RSA-1024 (declared 4096 — misleading constructor)	**High** — factorable in theory
2	PKCS#1 v1.5 (no OAEP)	Medium — Bleichenbacher-vulnerable
3	`System.Random` for seed (not RNG-secure)	**High** — predictable from boot time
4	Static PBKDF2 salt `01..08`	**High** — defeats salt purpose
5	1000 PBKDF2 iterations	Low — modern recommend ≥ 600 000
6	AES-128 (declared 256 — misleading initializer)	Low — still secure
7	Same Key+IV for **all files in one run** (CBC mode)	Medium — identical plaintexts → identical ciphertexts (information leak)

As deployed, `System.Random` is seeded from `Environment.TickCount` at first instantiation, and PBKDF2 uses a static salt with 1000 iterations. These are observed properties of the binary; their cryptographic implications are not assessed here.

File Encryption Process

Small files (≤ 10 MB) — full AES-CBC PKCS#7

`uJBRudfoWFSi.smethod_0(srcPath, dstPath)`:

1. Open `src` (existing file) FileMode.Open
2. Open `dst` (= `src + ".everest"`) FileMode.Create, FileShare.Read
3. Wrap `dst` in `CryptoStream(aes.CreateEncryptor(), Write)` with PaddingMode.PKCS7
4. Loop `srcStream.ReadByte()` → `cryptoStream.WriteByte((byte)num)` until EOF (byte-by-byte)
5. Close streams
6. **Secure-delete original**: `XXGcXDwCBmxY.wb(srcPath)` (overwrite chunks of 512 random bytes via `RNGCryptoServiceProvider`, SetLength(0), set timestamps to 2037-01-01, File.Delete)

After encryption: `<file>.everest` is the encrypted output.

Large files (> 10 MB) — intermittent AES-CBC NoPadding

`vvljguSShBVu.pM7i4uPaot1X(filePath, threshold_MB=10)`:

1. `File.Move(src, src + ".everest")` (rename in place)
2. Open `<src>.everest` FileMode.Open ReadWrite, FileShare.None
3. Compute `chunkSize = round(thresholdMB × fileSize / 200 / 32) × 32` (aligned to 32 B = 2× AES block)
4. Compute `step = fileSize / 6`
5. **First half**: encrypt 2 chunks at offsets `step×0`, `step×2` (i = 0, 2)
6. **Second half**: encrypt 4 chunks at offsets `fileSize - chunkSize - step×j` (j = 0, 1, 2, 3)
7. Each chunk: read → `aes.CreateEncryptor()` (PaddingMode.None, CipherMode.CBC) → write back at same offset

Total: ~6 chunks × `chunkSize` bytes encrypted in-place, the rest of the file untouched. Sufficient to **destroy** structured files (databases, VMDKs, archives) by corrupting headers/footers/metadata while encrypting only a small fraction.

Encrypted File Format

```
[ AES-CBC-PKCS7(plaintext) ]                  ← full mode (≤10 MB), full file is ciphertext
                                                 Original is securely deleted (separate file).

[ original_data_chunk_1 ]
[ AES-CBC-NoPadding(chunk_2) ]                ← intermittent (>10 MB)
[ original_data_chunk_3 ]
[ AES-CBC-NoPadding(chunk_4) ]
...
                                                 Original is renamed in-place to .everest
                                                 File length unchanged.
```

There is **no header**, **no magic byte**, **no per-file IV**, **no footer with key blob**. The only indicator of encryption is the `.everest` extension and (for large files) randomized content in chunk regions.

The per-victim AES seed is recoverable by the attacker via the RSA-wrapped value stored in: - The ransom note (`Your key: <base64>`) - The registry (`HKCU\Software\AppName\PublicKey`)

---

6. File Targeting

Encryption Mode Selection (per file)

```
if (FHS6J9WwaF2I[0] != "[auto]")     → use custom drive list (build-time configurable)
else                                  → DriveInfo.GetDrives(), skip DriveType.Network (3)

if (ZDcvczx4V1j9[0] != "[FULL]")      → only encrypt files matching extension whitelist
else                                  → encrypt all (default — FULL mode)

This sample's defaults: drives = [auto], extensions = [FULL]   → encrypt everything.
```

Excluded Directories (substring match, case-insensitive, 16 entries)

Directory	Reason
`programdata`	OS data + own drop folder
`windows`	OS files
`program files`	installed software
`perflogs`	performance logs
`appdata`	user roaming/local app data (twice in list)
`application data`	XP equivalent
`internet explorer`	browser cache
`mozilla`, `google chrome`	browser caches
`tor browser`	Tor browser folder
`system volume information`	VSS metadata
`boot`	bootloader
`msocache`	Office install cache
`trend micro`, `sophos`	AV install paths (avoid alerting them while running)

Excluded Extensions (system files)

`dll`, `exe`, `sys`, `ini`, `dat`, `inf` — protects OS-critical binaries (the ransomware needs the system to keep booting until the user reads the note).

Excluded Files (basename match, case-insensitive)

`ntdetect.com`, `bootnxt`, `ntldr`, `Recycle.Bin`, `bootmgr`, `thumbs.db`, `ntuser.dat.log`, `boot.ini`, `bootsect.bak`, `autoexec.bat`, `iconcache.db`

Plus runtime exclusions: - Files ending in `.everest` (already encrypted) - Files starting with `$` (system metadata, Recycle, $MFT, …) - Files containing `EVERESTRANSOMWARE` (the dropped notes) - Files with `FileAttributes.System` set - Zero-byte files - Path containing `\Desktop\` (skip per-directory note drop only)

Eligible File Selection Logic (`uJBRudfoWFSi.uNAefB4P3hFR`)

A file is encrypted iff none of the following match: - Path contains an excluded directory keyword - Extension matches `dll/exe/sys/ini/dat/inf` - Extension is the running ransomware's own (`.everest`) - Filename contains `EVERESTRANSOMWARE` - Filename starts with `$` - File has System attribute - File is empty (Length == 0)

If `string_3` is non-empty (build option), a custom inclusion-extension list applies on top.

LAN Propagation: Targets added at runtime

`fscjvXdLHuJt.rc()` populates `uJBRudfoWFSi.jBZdLoBjJEgb` with discovered SMB shares (UNC paths `\\<server>\<share>`), which are then walked exactly like local drives. Per-share dedup uses recursive directory count — if two UNC paths point to the same physical share via different mounts, only the first is encrypted.

7. Recovery Inhibition

Action	Implementation	Address (RVA)
VSS deletion (modern Windows)	`powershell.exe Get-CimInstance Win32_ShadowCopy | Remove-CimInstance`	`jjOjKFMSDrCb.BB()` `0x6664`
VSS deletion (XP fallback)	`vssadmin.exe resize shadowstorage /for=<X>: /on=<X>: /maxsize=401MB` (drives c-h, both maxsize=401MB and maxsize=unbounded) + `Delete Shadows /all /quiet`	same
System Restore Point wipe	`Srclient.dll!SRRemoveRestorePoint` (dyn-loaded) called for every WMI `\\.\root\default\systemrestore` instance, in reverse seqnum order	`WHtZqsICMCns.dd()` `0x8F3C`
Backup file deletion	`cmd /c del /s /f /q <pattern>` for ~80 patterns: `c:\*.VHD c:\*.bac c:\*.bak c:\*.wbcat c:\*.bkf c:\Backup*.* c:\backup*.* c:\*.set c:\*.win c:\*.dsk` (and same for d:, e:, f:, g:, h:)	`hyqyuDHMHPRC.aB()` `0x65E8`
Recycle Bin destruction	`cmd.exe /c rd /s /q %SYSTEMDRIVE%\$Recycle.bin` and `cmd.exe /c rd /s /q D:\$Recycle.bin`	EP `0x4584`
Restart Manager (free file locks)	dyn-load `rstrtmgr.dll` → `RmStartSession` + `RmRegisterResources` + `RmShutdown(force=1)` + `RmEndSession` for every locked file	`ODeLPhLRcCdj.Ic()` `0x72CC`
Anti-Raccine	Delete `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<X>` for X ∈ {vssadmin.exe, wbadmin.exe, bcdedit.exe, powershell.exe, diskshadow.exe, net.exe, taskkill.exe, wmic.exe}; delete `HKCU\…\Run\Raccine Tray`, `HKCU\SOFTWARE\Raccine`, `HKLM\…\EventLog\Application\Raccine`, `HKLM\SOFTWARE\Raccine`; `taskkill /F /IM Raccine.exe`; `taskkill /F /IM RaccineSettings.exe`; `schtasks /DELETE /TN "Raccine Rules Updater" /F`	`IdSMieJfrIOy.cB()` `0x66F8`

---

8. Targeted Services (~110)

Stopped via `ServiceController.Stop()` (substring or exact match in 100+-entry list)

McAfee: `mfefire`, `mfemms`, `mfevtp`, `McShield`, `McTaskManager`, `Antivirus`, `mfewc`, `McAfee`, `avpsus`, `DLPAgentService`, `DefWatch` (Symantec), `ccEvtMgr`, `ccSetMgr`, `SavRoam`, `RTVscan`

Trend Micro: `TmCCSF`, `tmlisten`, `ShMonitor`, `ntrtscan`

Sophos: `SAVAdminService`, `SAVService`, `SepMasterService`, `EhttpSrv`, `MMS`

Kaspersky: `KAVFS`, `KAVFSGT`, `kavfsslp`, `klnagent`, `klvssbridge64`

ESET: `ekrn`, `EPSecurityService`, `EPUpdateService`, `EsgShKernel`

MalwareBytes: `MBAMService`, `MBEndpointAgent`, `MBR Boot Service`

Acronis: `AcrSch2Svc`, `Acronis`, `CASAD2DWebSvc`, `CAARCUpdateSvc`

Veeam: `veeam`, `Veeam`, `VSNAPVSS`

QuickBooks: `QBFCService`, `QBIDPService`, `QBCFMonitorService`, `Intuit.QuickBooks.FCS`

Exchange: `msexchangeadtopology`, `msexchangeimap4`, `MSExchangeSRS`, `MSExchangeMTA`, `MSExchangeSA`, `MSExchangeIS`, `MSExchangeMGMT`, `MSExchangeES`, `ARSM`, `unistoresvc_1af40a`

MSSQL/SSAS/SSRS/SSIS: `MSOLAP$TPSAMA`, `MSOLAP$SYSTEM_BGC`, `MSOLAP$TPS`, `ReportServer$TPS`, `ReportServer$TPSAMA`, `ReportServer$SYSTEM_BGC`, `ReportServer`, `MsDtsServer110`, `MsDtsServer100`, `MsDtsServer`, `IisAdmin`, `SQLTELEMETRY`, `SQLTELEMETRY$ECWDB2`, `SQLWriter`, `SstpSvc`

Other: `vapiendpoint`, `Smcinst`, `SmcService`, `SntpService`, `svcGenericHost`, `swi_`, `TrueKey`, `TrueKeyScheduler`, `TrueKeyServiceHelper`, `WRSVC`, `OracleClientCache80`, `wbengine`, `RESvc`, `sacsvr`, `audioendpointbuilder`, `AVP`, `DCAgent`, `bedbg`, `EraserSvc11710`, `Enterprise Client Service`, `NetMsmqActivator`, `stc_raw_agent`, `Symantec System Recovery`, `UI0Detect`, `MSExchange`, `POP3Svc`, `SMTPSvc`, `Zoolz 2 Service`, `aphidmonitorservice`, `PDVFSService`, `FA_Scheduler`, `SDRSVC`, `ESHASRV`, `Intel(R) PROSet Monitoring`, `BMR Boot Service`, `YooIT`, `zhudongfangyu` (Qihoo 360), `MasterControl`, `klntsrv`, `SAVAGENT`

Substring also matched on `backup,sql` (catch-all for backup and SQL-related services with custom names).

Set to `start= disabled` via `sc.exe config` (one-shot startup):

`SQLTELEMETRY`, `SQLTELEMETRY$ECWDB2`, `SQLWriter`, `SstpSvc`, `MBAMService`

Started/re-enabled via `ServiceController.Start()`:

`Dnscache`, `fdPHost`, `FDResPub`, `SSDPSRV`, `upnphost` (also `sc config <name> start= auto`) — for SMB/SSDP discovery.

Services with name starting `sophos` (substring match in init):

Stopped explicitly during the AV stomp pass.

---

9. Targeted Processes (~50)

Killed via `Process.Kill` looped through `Process.GetProcesses()` (substring match in `jBpIYgcpDHj5` decoded list):

`mspub`, `mydesktopqos`, `mydesktopservice`, `mysqld`, `mysqld-nt`, `mysqld-opt`, `sqbcoreservice`, `firefoxconfig`, `agntsvc`, `thebat`, `thebat64`, `steam`, `encsvc`, `excel`, `CNTAoSMgr`, `sqlwriter`, `tbirdconfig`, `dbeng50`, `ocomm`, `infopath`, `mbamtray`, `zoolz`, `thunderbird`, `dbsnmp`, `xfssvccon`, `Ntrtscan`, `isqlplussvc`, `onenote`, `PccNTMon`, `msaccess`, `outlook`, `tmlisten`, `msftesql`, `powerpnt`, `visio`, `winword`, `wordpad`, `ocautoupds`, `ocssd`, `oracle`, `sqlagent`, `sqlbrowser`, `sqlservr`, `synctime`

Anti-analysis kill list (Thread #1, `iYqREGyyukhw`)

`tcpdump`, `HTTPNetworkSniffer`, `NetworkTrafficView`, `NetworkMiner`, `NoFuserEx`, `Universal_Fixer`, `UnConfuserEx`, `MegaDumper`, `pe-sieve`, `LordPE`, `protection_id`, `PEiD`, `CFF Explorer`, `RDG Packer Detector`, `ida64`, `dotpeek64`, `dotpeek`, `ilspy`, `de4dot`, `dnspy-x86`, `dnspy`, `x32dbg`, `x64dbg`, `ollydbg`, `Intercepter-NG`, `intercepter`, `http analyzer stand-alone`, `fiddler`, `effetech http sniffer`, `firesheep`, `IEWatch Professional`, `dumpcap`, `wireshark`, `wireshark portable`, `sysinternals tcpview`

Match on either `ProcessName.Contains(target)` or `MainWindowTitle.Contains(target)`. Process termination via `ODeLPhLRcCdj.Ic(file)` which uses Restart Manager `RmShutdown(RmForceShutdown=1)`.

Memory pig kill (Thread #3, `wqqiwcJkIwbq`)

Any process with `PrivateMemorySize64 > 250 MB`, except those in the protected list: `chrome`, `opera`, `msedge`, `firefox`, `iexplore`, `explorer`, `winlogon`, `SearchIndexer`, `wininit`, `SearchApp`, `SearchUI`, `powershell`

This catches sandboxes (Cuckoo, Joe Sandbox), large debuggers (x64dbg with heavy plugins), and forensic tools while sparing the user's browsers and core OS shell so the victim can read the note.

---

10. Persistence & Evasion

String Obfuscation (ConfuserEx `constants` preset)

All literal strings encrypted in the UserStrings heap. Resolution flow: 1. Each call site does `<Module>.m( <Module>.<getter>() )`. 2. `<getter>()` (e.g. `smethod_2`, `dmDRWSNaSUn4`) returns a concatenation of 4 cipher fragments via `string.Concat( tObzrXtH7fNR(us#x), tObzrXtH7fNR(us#y), … )`. 3. `<Module>.m(string)` performs Base64 decode → GZip decompress → table lookup against a static char table `L` (initialized in `<Module>..cctor` from `smethod_1().ToCharArray()`).

The decryptor is **single-stage but stateful**: the static field `Bgz7AQodFFw2` is lazy-initialized on first call to `M()`. Subsequent calls hit the cache, so resolution is O(1) after warm-up.

Some methods (e.g. `<Module>.smethod_19` for the RSA pubkey XML, `<Module>.zgTsyaLKRZwg` for hex-32 constants) return raw strings without going through `m()` — they are consumed directly.

Anti-Analysis (Thread `iYqREGyyukhw` + `pwSlLXRBnDqf` lists)

- 35-process kill list (debuggers, decompilers, sniffers, packer-IDs)
- 4-second polling, kill on `ProcessName.Contains` or `MainWindowTitle.Contains`
- Termination uses Restart Manager `RmShutdown(force=1)` rather than `Process.Kill()` — survives many anti-tamper hooks

Process Self-Protection (`KZsyzkgZlDQw.UB()`)

Resolves `kernel32!GetCurrentProcess`, `advapi32!GetKernelObjectSecurity`, `advapi32!SetKernelObjectSecurity` dynamically.

```
RawSecurityDescriptor sd = GetKernelObjectSecurity(GetCurrentProcess(),
DACL_SECURITY_INFORMATION);
sd.DiscretionaryAcl.InsertAce(0,
  new CommonAce(AceFlags.None,
                AceQualifier.AccessDenied,
                accessMask: 0x1F0FFF /* PROCESS_ALL_ACCESS */,
                sid: WorldSid /* Everyone S-1-1-0 */,
                isCallback: false, opaque: null));
SetKernelObjectSecurity(GetCurrentProcess(), DACL_SECURITY_INFORMATION, sd);
```

After this call, **even SYSTEM-level processes cannot terminate the ransomware** without first re-acquiring `WRITE_DAC` rights (which requires owner privilege restoration). Defenders attempting `taskkill /F` will receive Access Denied.

Dynamic API Resolution (`zPyJQYTgymZw`)

The only static `DllImport`s are `kernel32!LoadLibrary`, `kernel32!GetProcAddress`, `user32!SystemParametersInfo`. Everything else is dyn-resolved through: - `kernel32` (cached in `intptr_0`) → `SetThreadExecutionState`, `GetCurrentProcess` - `rstrtmgr` (`intptr_1`) → `RmStartSession`, `RmRegisterResources`, `RmShutdown`, `RmEndSession` - `advapi32` (`XRfvtFXawnqJ`) → `GetKernelObjectSecurity`, `SetKernelObjectSecurity` - `Netapi32.dll` (`CORc3RU2G45v`) → `NetDfsEnum` - `Mpr.dll` (`intptr_2`) → `WNetOpenEnumA`, `WNetEnumResourceA`, `WNetCloseEnum` - `Srclient` (`D7KjYNQiTJHs`) → `SRRemoveRestorePoint`

Defeats simple IAT-based YARA signatures.

LAN Propagation Engine (`fscjvXdLHuJt.rc()`)

Five concurrent discovery sources:

1. **ARP cache scan** (`HuHoNzPzkzhd.ZFQ6IEALo4nc()` → `arp -a` parser) → produces (IP, MAC) pairs → for each: `PryijgOwcshz.ed(MAC, IP)` sends a **Wake-On-LAN magic packet** (UDP broadcast on ports 7 and 9, payload = `0xFF×6 + MAC×16` = 102 B). This wakes powered-down hosts before the SMB scan.
2. **`net view`** (`KsYiGUhgWoMh.smethod_0()`) → server browser list, parsing `\\<server>` lines.
3. **`NetDfsEnum`** (`KsYiGUhgWoMh.btgAofVbNWq7(MachineName)`) → DFS namespaces of the local domain.
4. **`Win32_NetworkAdapterConfiguration`** WMI → local IP addresses (used to skip self-encryption).
5. **WNet recursive global enumeration** (`XSXFlWKiczfX(RESOURCE_GLOBALNET, RESOURCETYPE_DISK, RESOURCEUSAGE_RESERVED|CONNECTABLE, RESOURCEDISPLAYTYPE_FILE)`) — enumerates entire visible Windows network tree.
6. **`Win32_Share`** WMI → local share names.
7. **`Win32_NetworkConnection`** + **`Win32_MappedLogicalDisk`** → already-mapped UNCs.

Per discovered UNC: `net.exe use \\<unc>`. Anonymous credentials by default (current user impersonation). Targets are deduplicated via recursive `GetDirectories` count (cheap) before being added to `uJBRudfoWFSi.jBZdLoBjJEgb` for encryption.

The final UNC list is **persisted to** **`C:\ProgramData\<MD5_hex>`** where `MD5_hex = MD5(ProcessorId + VolumeSerialNumber)`. The file is `File.WriteAllLines`-written with one UNC per line.

Anti-Recovery Stack (full)

In strict execution order: 1. Geo-fence (CIS exclusion) 2. Anti-Raccine (`IdSMieJfrIOy.cB`) — neutralize pre-installed protection 3. Anti-sleep (`uxTwxHBatbAA.smethod_0`) — `SetThreadExecutionState(ES_CONTINUOUS|ES_SYSTEM_REQUIRED|ES_DISPLAY_REQUIRED)` 4. Self-protection DACL (`KZsyzkgZlDQw.UB`) — block external kill 5. Backup file `del /s /f /q` sweep (`hyqyuDHMHPRC.aB`) 6. VSS shadow copy deletion (`jjOjKFMSDrCb.BB`) 7. System Restore Point wipe (`WHtZqsICMCns.dd`) 8. Defender CFA disabled (Windows 10/11 only, `Set-MpPreference -EnableControlledFolderAccess Disabled`) 9. Recycle Bin destruction (`rd /s /q $Recycle.bin` on C: and D:) 10. SMB1 re-enabled (`Enable-WindowsOptionalFeature SMB1Protocol`) 11. Firewall opened for `Network Discovery` and `File and Printer Sharing` 12. Network discovery services re-enabled (`Dnscache`, `SSDPSRV`, …) 13. Long path support enabled (`LongPathsEnabled=1`) 14. Token filter policy disabled (`LocalAccountTokenFilterPolicy=1`) — enables remote admin via SMB 15. Linked connections enabled (`EnableLinkedConnections=1`) — admin sees mapped drives 16. ACL grants `Everyone:F` on C:, D:, Z: drives (`icacls`) 17. Mount unlettered volumes (mountvol) 18. Wake-On-LAN ARP'ed hosts 19. SMB share enumeration + auto-mount 20. Self-delete after notepad spawn

Anti-Analysis Summary (mapped to commonly-tracked technique IDs)

Technique	UXXX/TXXX	Address (RVA)	Description
String encryption (ConfuserEx constants)	U0322	`<Module>.m` `0x3EB0`	GZip+B64+lookup
Code obfuscation (renamed identifiers + Int confusion)	U0123 / T1027	sample-wide	ConfuserEx rename + math arithmetic for constants
Process kill list	U0008 / T1518.001	`iYqREGyyukhw` `0x64CC`	RE/sniffer tools
Service stop list	U0072 / T1489	`avlgfDFtmIlr` `0x74A4`	AV/backup services
Memory-based termination heuristic	T1622	`wqqiwcJkIwbq` `0x6B24`	>250 MB outliers
Process self-protection via DACL	U1311 / T1564	`KZsyzkgZlDQw.UB` `0x70A4`	AccessDenied for Everyone
Dynamic API resolution	T1027.007	`zPyJQYTgymZw` `0x65BC`	LoadLib + GetProcAddr per use
Geo-fencing CIS exclusion	U0203 / T1614.001	EP `0x4584`	Cultures + LCIDs
Mutex single-instance	U0073 / T1480	EP `0x4584`	Global GUID
Anti-Raccine	U1109	`IdSMieJfrIOy.cB` `0x66F8`	IFEO clean + reg + taskkill + schtasks
Anti-sleep	U0211	`uxTwxHBatbAA.smethod_0` `0x6F38`	SetThreadExecutionState
Wake-On-LAN	T1018	`PryijgOwcshz.ed` `0x9058`	Pre-encryption host wake-up
Anti-recovery (VSS+RP+backups)	U0027 / T1490	`jjOjKFMSDrCb.BB` `0x6664` + `WHtZqsICMCns.dd` `0x8F3C` + `hyqyuDHMHPRC.aB` `0x65E8`	full stack
Self-delete	U0026	`XXGcXDwCBmxY.kahdFWq2bR0P` (cmd /C ping & fsutil setZeroData & Del)	wipe + delete
Restart Manager file unlock	T1490	`ODeLPhLRcCdj.Ic` `0x72CC`	RmStart/Register/Shutdown/End
Wallpaper change	T1491.001	`QmTYuuYAWWkY.qNvEjwNtgDZ3` `0x77FC`	Bitmap with note text

11. Command-Line Arguments

The sample takes **no command-line arguments**. All configuration is build-time embedded: - Hard-coded victim salutation in note (encrypted in `<Module>.smethod_5`) - RSA public key in `xb` field - Mutex GUID - Drive whitelist in `FHS6J9WwaF2I` (default `[auto]`) - Extension whitelist in `ZDcvczx4V1j9` (default `[FULL]`) - Threshold for intermittent encryption (10 MB)

A more complete Everest builder may exist (with command-line `[FULL]/[auto]` overrides) but this binary is hard-coded.

12. Static Imports Summary

Category	Key APIs / .NET classes
Crypto	`AesCryptoServiceProvider` (CBC, KeySize=128), `RSACryptoServiceProvider` (1024 bits, PKCS#1 v1.5), `Rfc2898DeriveBytes` (PBKDF2-HMACSHA1, 1000 iter, salt `01..08`), `RNGCryptoServiceProvider` (used **only** for secure-delete file overwrite — not for seed gen!), `MD5`, `Convert.ToBase64String / FromBase64String`
File I/O	`FileStream`, `CryptoStream`, `Directory.EnumerateFiles/Directories`, `DriveInfo.GetDrives`, `File.GetAttributes/SetAttributes/Move/Delete`, `FileInfo`, `Path.Combine/GetDirectoryName/GetTempPath`
Process	`Process.GetProcesses`, `Process.GetCurrentProcess`, `Process.Start`, `Process.Kill`, `ProcessStartInfo` (RedirectStandardOutput, CreateNoWindow=true)
Registry	`Registry.CurrentUser/LocalMachine`, `OpenSubKey/CreateSubKey/DeleteSubKey/SetValue`
Threading	`Thread`, `ThreadStart`, `Mutex`, `Task`, `Parallel.ForEach`, `TaskScheduler.UnobservedTaskException` (suppress crashes)
Network / SMB	`Ping`, `UdpClient.Send` (Wake-On-LAN broadcast), `IPAddress`, `NetworkInterfaceType`, dyn-loaded `Mpr!WNet*`, `Netapi32!NetDfsEnum`, ARP via `arp.exe` parsing. `WebClient.DownloadData` is referenced by `Helpers.DownloadUrl` but that method is unreachable in this binary (no caller other than its own retry path).
WMI	`ManagementObjectSearcher`, `ManagementClass`, `Win32_Processor`, `Win32_LogicalDisk`, `Win32_OperatingSystem`, `Win32_Share`, `Win32_NetworkAdapterConfiguration`, `Win32_NetworkConnection`, `Win32_MappedLogicalDisk`, `\\.\root\default\systemrestore`
Service Control	`ServiceController.GetServices/Stop/Start/WaitForStatus` (via `System.ServiceProcess`)
UI / Wallpaper	`Bitmap`, `Graphics.DrawString`, `Image.Save`, `SystemParametersInfo`, `Screen.PrimaryScreen`, `NotifyIcon` (balloon tip via `OwMvDWEPTi4W`)
Security	`WindowsIdentity.GetCurrent`, `FileSystemAccessRule`, `DirectorySecurity`, `RawSecurityDescriptor`, `CommonAce`, `SecurityIdentifier(WellKnownSidType.WorldSid)`
Globalization	`CultureInfo.InstalledUICulture`, `CultureInfo.CurrentCulture`, `CultureInfo.LCID`
Native (P/Invoke declared)	`kernel32!LoadLibrary`, `kernel32!GetProcAddress`, `user32!SystemParametersInfo`

13. Class-Level Renaming Map (analysis-side identifiers)

Symbol renaming was performed at analysis time. The cleaned binary retains ConfuserEx-randomized names; below is the inferred semantic map used in this report.

Token	Obfuscated	Semantic name	Methods	Purpose
0x02000003	iWvpiwZPYibt	**Program**	5	EP, write file helper, exit, global state
0x02000004	uJBRudfoWFSi	**Encryptor**	11	Drive walker + AES-CBC encryption + PBKDF2
0x02000007	pwSlLXRBnDqf	**Lists**	0	Static decrypted lists (excl folders/files/exts, RFC1918, admin shares, ...)
0x02000008	XXGcXDwCBmxY	**Helpers**	13	Process spawn, key gen, secure delete, balloon, OS info, WebClient (unreachable), self-delete
0x0200000B	thbXuslnyRzc	**MachineId**	1	ProcessorId + VolumeSerialNumber
0x0200000C	ACWpKFzdfHch	**RsaWrap**	2	RSA-1024-PKCS1v1.5 encrypt seed → Base64
0x0200000D	vvljguSShBVu	**IntermittentEnc**	4	Large-file chunked AES-CBC NoPadding
0x0200000E	zPyJQYTgymZw	**NativeLoader**	2	LoadLibrary + GetProcAddress + 6 module handles
0x0200000F	iYqREGyyukhw	**AntiAnalysisKiller**	2	Thread #1 — kill RE tools
0x02000010	hyqyuDHMHPRC	**BackupWiper**	2	del backup patterns
0x02000011	jjOjKFMSDrCb	**VssDelete**	1	PowerShell or vssadmin shadow wipe
0x02000012	IdSMieJfrIOy	**AntiRaccine**	2	IFEO + reg + taskkill + schtasks
0x02000013	NYPSjWpoCyzg	**NetServicesEnable**	3	Re-enable network discovery services
0x02000014	LcKDzxOgRJco	**AclGrant**	3	icacls + DirectorySecurity grant
0x02000015	wqqiwcJkIwbq	**MemPigKiller**	1	Thread #3 — kill processes >250 MB
0x02000016	NquuvOEOwNeN	**MountUnlettered**	1	mountvol of GUID volumes
0x02000017	uxTwxHBatbAA	**AntiSleep**	1	SetThreadExecutionState
0x02000019	KZsyzkgZlDQw	**SelfProtect**	4	DACL deny-Everyone on own process
0x0200001D	cWcgqDvSOTiq	**ClearReadOnly**	1	unset FileAttributes.ReadOnly
0x0200001E	ODeLPhLRcCdj	**RestartManager**	2	rstrtmgr.dll force-release file locks
0x02000025	avlgfDFtmIlr	**ServiceProcessKiller**	7	Thread #2 — services + procs

Token	Obfuscated	Semantic name	Methods	Purpose
`0x02000026`	`QmTYuuYAWWkY`	**`Wallpaper`**	2	Bitmap + SystemParametersInfo
`0x02000027`	`HuHoNzPzkzhd`	**`ArpParser`**	3	arp -a → (IP, MAC) tuples
`0x02000028`	`fscjvXdLHuJt`	**`LanPropagation`**	3	5-source UNC discovery + WoL + share mount
`0x02000029`	`KsYiGUhgWoMh`	**`LanDiscovery`**	4	net view + NetDfsEnum + Ping + IPs
`0x0200002E`	`XSXFlWKiczfX`	**`WNetScanner`**	1 (ctor)	Mpr.dll global net recursive enum
`0x02000038`	`WHtZqsICMCns`	**`RestorePointWipe`**	1	Srclient.dll!SRRemoveRestorePoint
`0x0200003A`	`PryijgOwcshz`	**`WakeOnLan`**	2	UDP magic packet broadcast

(Delegate types `Delegate0`, `LB`, `PB`, `OpRNIylj5zRs`, `YB`, `EoYUNEvD8H10`, `MZWe0rEq0d6C`, `cNhApOjr70FU`, `Fc`, `V5q76ApXx2EK`, `M3tyndpTmJXU`, `TCZM7r3jKmGI`, `iC`, `jaGJo5BxJgwo`, `Cd` are P/Invoke delegate trampolines for the dyn-resolved native APIs.)

---

14. Indicators of Compromise (IOCs)

Hashes

Type	Value
SHA-256 (original protected)	`1df92bf4c967297d8a39fc3f619a56702ee96d5cf9196b8e1d5b3654746c6514`
MD5 (original protected)	`0dd70c334507188714bae0af7229b379`
SHA-256 (de4dot-cleaned x4)	`b30b7911b256daeceb631dec923f83c0ef17f34eb46452d5ab32aab2540be361`

Network

Type	Value
Onion blog	`ransomocmou6mnbquqz44ewosbkjk3o5qjsl3orawojexfook2j7esad.onion`
Email	`everestaf@onionmail.org`
User-Agent string (embedded, **dead code**)	`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134` — referenced by `Helpers.DownloadUrl`, which has no callers in this binary
WoL UDP ports	7, 9 (broadcast)

Files

Indicator	Value
Ransom note	`EVERESTRANSOMWARE.txt` (in C:\ProgramData, Desktop, every traversed dir)
Encrypted extension	`.everest`
Wallpaper file	`%TEMP%\[WALLNAME].bmp` (literal `[WALLNAME]`)
Spread tracking file	`C:\ProgramData\<MD5(ProcessorId+VolumeSerial)>` (lowercase hex, no extension)

Registry

Key	Value	Description
`HKCU\Software\AppName`	`PublicKey` = `<base64>`	RSA-wrapped per-victim seed (recovery token)
`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System`	`LocalAccountTokenFilterPolicy=1`	Enable remote admin
Same key	`EnableLinkedConnections=1`	Mapped drives visible to admin processes
`HKLM\SYSTEM\CurrentControlSet\Control\FileSystem`	`LongPathsEnabled=1`	Bypass MAX_PATH for long shares
`HKCU\Control Panel\Desktop`	`WallPaper=%TEMP%\[WALLNAME].bmp`	Persistence of ransom wallpaper

Mutex

Name	Scope
`Global\7efc73f7-fda1-42d1-a4c5-8f1670bd08a5`	Global session-wide single-instance

Behavioral

- 3 concurrent worker threads (4 s, 15 s, 2.5 s loops) running for entire process lifetime
- `mountvol.exe` invocation enumerating `\\?\Volume{...}\` GUIDs and mounting them on free letters
- Mass `cmd.exe /c del /s /f /q <pattern>` calls for backup file extensions on drives c-h
- `powershell.exe Get-CimInstance Win32_ShadowCopy | Remove-CimInstance`
- `arp -a` execution
- UDP broadcast traffic to ports 7 and 9 (Wake-On-LAN)
- `net.exe view`, `net.exe use \\…`, `netsh advfirewall` invocations
- `vssadmin resize shadowstorage /maxsize=401MB` then `Delete Shadows /all /quiet` on XP-class systems
- DACL of own process modified with AccessDenied ACE for World SID
- `schtasks /DELETE /TN "Raccine Rules Updater" /F`
- Final cmdline: `cmd /C ping 127.0.0.7 -n 3 > Nul & fsutil file setZeroData offset=0 length=524288 "<self>" & Del /f /q "<self>"`

Distinctive Strings (typo signatures, useful for hunting)

- `"trully"` (instead of "truly") in note
- `"singifically"` (instead of "significantly") in note

- `"backups , etc."` (spaced comma) in note
- `"FindNoteOnDesktop!"` in balloon
- `"Find note file on desktop!"` in balloon
- `"Greetings from the Everest team"` (note opener)
- `"Yours trully Everest Team"` (note closing — combined typo)
- `"EVERESTRANSOMWARE"` (filename string, also in code)
- `[WALLNAME]` (literal, present as-is in the wallpaper file path string)

---

15. MITRE ATT&CK Mapping

ID	Technique	Sub-technique	Implementation in Sample
T1486	Data Encrypted for Impact	—	AES-128-CBC PKCS#7 / NoPadding (large files) of all eligible files on local drives + mounted unlettered volumes + LAN shares
T1490	Inhibit System Recovery	—	VSS (PS or vssadmin), SRRemoveRestorePoint, backup file `del /s /q`, Recycle Bin `rd /s /q`, Restart Manager force-shutdown of locking apps
T1489	Service Stop	—	`ServiceController.Stop` for ~100 services (AV, backup, MSSQL, Exchange, MBAM, Veeam, Acronis, ...) + `sc config X start= disabled`
T1078.003	Valid Accounts: Local Accounts	—	Implicit current-user impersonation for `net use \\<unc>`
T1018	Remote System Discovery	—	`arp -a` parsing + ARP-driven Wake-On-LAN
T1135	Network Share Discovery	—	`net view`, `NetDfsEnum`, `WNetEnumResource*`, `Win32_Share`, `Win32_NetworkConnection`
T1021.002	Remote Services: SMB/Windows Admin Shares	—	`net.exe use \\<unc>` over discovered shares
T1071.001	Application Layer Protocol: Web Protocols	—	`WebClient.DownloadData` is **referenced** in `Helpers.DownloadUrl` but **not reachable** from any other code path in this binary (only its own retry loop calls it). Capability is present at the byte level but inert at runtime
T1027	Obfuscated Files or Information	—	ConfuserEx (rename + constants encryption + Int confusion + modified module .cctor)
T1027.007	Dynamic API Resolution	—	All non-corlib native APIs via `LoadLibrary` / `GetProcAddress`
T1140	Deobfuscate/ Decode Files or Information	—	Runtime UserStrings GZip+Base64 decryption via `<Module>.m`
T1480	Execution Guardrails	—	`Global\<GUID>` mutex single-instance, CIS culture/ LCID geo-fence
T1564	Hide Artifacts	—	Process self-DACL deny-Everyone (`KZsyzkgZlDQw.UB`)
T1614.001	System Location Discovery: System Language Discovery	—	CIS culture/LCID exclusion
T1622	Debugger Evasion	—	Memory-pig killer (>250 MB heuristic) targets sandboxes/heavy debuggers
T1518.001	Software Discovery: Security Software Discovery	—	Anti-analysis kill list matches against `ProcessName` and `MainWindowTitle`
T1112	Modify Registry	—	Multiple HKCU/HKLM writes (LongPathsEnabled, LocalAccountTokenFilterPolicy, EnableLinkedConnections, WallPaper, AppName\PublicKey)

ID	Technique	Sub-technique	Implementation in Sample
T1497.001	Virtualization/ Sandbox Evasion: System Checks	—	`Win32_OperatingSystem` Caption query (e.g. branch on "10" / "XP")
T1491.001	Defacement: Internal	—	Wallpaper change to ransom note bitmap
T1490 (sub)	Disable System Defenses	—	`Set-MpPreference -EnableControlledFolderAccess Disabled`, SMB1 re-enabled
T1562.001	Impair Defenses: Disable or Modify Tools	—	Anti-Raccine: IFEO clean + reg + processes + scheduled task
T1070.004	Indicator Removal: File Deletion	—	Secure-delete wb() (overwrite + timestomp 2037-01-01 + delete)
T1070.006	Indicator Removal: Timestomp	—	wb() sets CreationTime/LastAccessTime/LastWriteTime to 2037-01-01
T1059.003	Command and Scripting Interpreter: Windows Command Shell	—	Heavy `cmd.exe /c …` usage
T1059.001	Command and Scripting Interpreter: PowerShell	—	`Set-MpPreference`, `Enable-WindowsOptionalFeature`, `Get-CimInstance Win32_ShadowCopy`
T1136.001	Account Manipulation (registry tweaks)	—	LocalAccountTokenFilterPolicy enables remote admin tokens
T1003.002	Wake-on-LAN	—	UDP magic packet broadcast (T1018-adjacent — pre-encryption power-on of dormant hosts)
T1133	External Remote Services	—	SMB1 re-enabled (SMB1Protocol Windows Optional Feature) — broadens lateral surface
T1485	Data Destruction	—	Backup file `del /s /f /q`, Recycle Bin `rd`, restore points wiped
T1490 + T1486	Combined	—	Restart Manager forces release of file handles before encryption
T1083	File and Directory Discovery	—	`Directory.EnumerateFiles/Directories` recursive
T1622	Process Self-Protection	—	DACL inserts AccessDenied ACE for World SID
TA0040	Impact (tactic)	—	All of the above culminate in encryption + ransom note + wallpaper

16. Summary

The sample is a .NET ransomware identified as **EVEREST** by the embedded strings (extension `.everest`, note filename `EVERESTRANSOMWARE.txt`, salutation "Greetings from the Everest team", contact email `everestaf@onionmail.org`, blog onion URL). The note in `<Module>.smethod_5` carries a hard-coded victim salutation (redacted in this report). The build pipeline is **ConfuserEx 1.x** (or compatible fork) with `rename`, `constants`, `anti-tamper`, and `compressor` presets — confirmed by DIE heuristic flags (Modified managed EP + Int confusion + Short names + Bad .cctor format) and the lazy GZip+Base64+lookup string decryptor.

Sophistication highlights (above-average for .NET ransomware): - 3 concurrent worker threads with distinct triggers (RE-tool kill, AV-service stomp, memory-pig terminate) running for entire process lifetime - Restart Manager wrapper (`rstrtmgr.dll`) for force-releasing file handles on locked Office/ PDF/DB files before encryption - Process self-protection via DACL `AccessDenied` ACE for World SID — blocks even SYSTEM kill - Anti-Raccine module (delete IFEO entries + registry + scheduled task) - Mount of unlettered volumes (EFI/recovery/hidden partitions) via `mountvol.exe` - Wake-On-LAN broadcast to ARP'ed hosts pre-encryption (rare in ransomware corpus) - Five-source LAN share discovery (ARP-driven WoL + `net view` + `NetDfsEnum` + WNet recursive global enum + `Win32_Share` + `Win32_NetworkConnection` / `Win32_MappedLogicalDisk`) - Recursive directory-count dedup of UNC paths

Crypto-stack weaknesses (below-average for .NET ransomware): - **RSA-1024** despite a misleading `RSACryptoServiceProvider(4096)` constructor — `FromXmlString` overrides with a 128-byte modulus - **AES-128** despite `KeySize=256` setter — `Key=byte[16]` overrides - Per-victim seed generated by `System.Random` (not `RNGCryptoServiceProvider`) — predictable from boot time - **Static 8-byte salt** `01..08` for PBKDF2 - **Same Key+IV for every file in a single run** (CBC-mode IV reuse → identical plaintexts produce identical ciphertexts) - **PKCS#1 v1.5** padding for RSA (no OAEP) - 1000 PBKDF2 iterations (modern recommend ≥ 600 000)

Operational implications: - The seed is generated by `System.Random.Next(33,127)` (32 bytes of printable ASCII excluding space and backslash). The PBKDF2 salt is the static 8-byte literal `01 02 03 04 05 06 07 08`. The PBKDF2 iteration count is 1000. - The ransomware aggressively **broadens the lateral surface** before encryption (SMB1 re-enabled, LocalAccountTokenFilterPolicy, EnableLinkedConnections, firewall opened, network discovery services started) — defenders observing these registry/firewall changes have a clear pre-encryption signal. - The `[WALLNAME]` placeholder is present **as a literal** in the wallpaper file path string (`%TEMP%\[WALLNAME].bmp`). This is an unsubstituted template token in the binary.

Two facts from the binary are sufficient to characterize the deployment: 1. The note string in `<Module>.smethod_5` contains a hard-coded victim salutation. 2. The RSA public key in `<Module>.smethod_19` is a 1024-bit modulus specific to this sample.

These two strings are sample-unique. The mutex GUID, onion URL, contact email, and the rest of the operational strings (kill lists, command sequences) are reusable across builds.