
# Ironchain

Technical Analysis — Cti Report

**REVERSE-ENGINEERED REPORT**

**RansomLook** · ransomlook.io
File last modified: 2026-04-30

Sample SHA-256: `2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22`

% IronChain 3.0 Ransomware — CTI Brief

# Executive Summary

| | |
|---|---|
| **Family** | IronChain 3.0 |
| **Platform** | Windows x64 (PE32+, Python 3.14 PyInstaller) |
| **Encryption** | AES-GCM intermittent + RSA-4096-OAEP runtime-generated |
| **Sophistication (Operational)** | **Technically advanced but poorly engineered** — comprehensive persistence (6 layers), SafeBoot survival, SMB/WMI lateral movement, MFT destruction, UEFI+BIOS boot destruction, but critical bugs and auto-sabotage |
| **Sophistication (Cryptographic)** | **Fundamentally broken** — multi-layer cipher shifts discarded, runtime RSA key never exfiltrated |
| **Exfiltration** | **None observed** — no C2 communication, networking limited to lateral movement and DDoS |
| **Recovery feasibility** | **Zero** — even with operator cooperation, shift values lost make files unrecoverable |
| **Sample SHA-256** | `2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22` |

# Critical Weaknesses

## 1. Multi-layer cipher with lost shift values

The encryption pipeline applies a pseudo-cipher layer before AES-GCM that destroys recoverability:

```
def mc(data):
    layers = random.randint(2, 7)        # 2-7 random passes
    shifts = []
    result = bytearray(data)
    for _ in range(layers):
        shift = random.randint(-255, 255)
        shifts.append(shift)
        for i in range(len(result)):
            if random.random() < 0.75:   # 75% of bytes shifted per layer
                result[i] = (result[i] + shift) & 0xFF
    return bytes(result), shifts
```

The calling function `ef()` discards the second return value (`shifts`) with `_`. These shifts are never written to disk, never transmitted, and `random` is not seeded. The transformation is **mathematically irreversible**.

**Impact**: Even if the operator possessed the RSA private key and cooperated fully, victim files remain permanently unrecoverable.

## 2. Runtime RSA key generation with no exfiltration

The ransomware generates a fresh 4096-bit RSA keypair at startup via `RSA.generate(4096)`. The private key exists only in process memory and is never serialized, transmitted, or stored. No C2 communication channels exist in the binary.

**Impact**: The private key is lost when the process terminates. No entity can decrypt victim files, including the operator.

### 3. MFT destruction parallel to encryption

Function `em()` overwrites 1 GB of the NTFS Master File Table with AES-CTR using a random key that is never saved:

```
mft_start_sector = 786432 × 8  # typical NTFS MFT location
mft_size_sectors = 1073741824 // 512  # 1 GB
```

This runs as a parallel daemon thread during encryption, destroying filesystem metadata for all files, including those not yet encrypted.

**Impact**: Even files never touched by the encryption logic become inaccessible due to MFT corruption. Recovery tools cannot reconstruct file names or directory structures.

### 4. Boot destruction before encryption completes

The main thread overwrites both BIOS and UEFI boot loaders immediately after initial setup:

- **BIOS MBR**: 512-byte custom boot sector with "CHAINED - PAY TO DECRYPT" message (contains off-by-one bug preventing message display)
- **UEFI**: Zeroes first 512 bytes of `bootmgfw.efi` on all EFI System Partitions

Combined with `bcdedit /set {default} safeboot minimal`, the machine becomes permanently unbootable regardless of encryption completion.

**Impact**: If encryption is interrupted or fails, the machine remains bricked. The operator gains no leverage from partial encryption.

---

## Sample Fingerprint

| | |
|---|---|
| SHA-256 | `2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22` |
| MD5 | `71318a233ae849db62eb4037486ba3ff` |
| Size | 11,282,718 bytes (11 MB) |
| Format | PE32+ x64 PyInstaller bundle |
| Python version | 3.14.4 (magic `2b0e0d0a`) |
| Overlay size | 10,933,534 bytes (10.4 MB archive) |
| String obfuscation | None — Python bytecode in plaintext |
| Mutex | `IronChainMutex_<8_random_chars>` |

---

## Killchain at a Glance

```
[1] Admin check      OpenProcessToken + GetTokenInformation(TokenElevation)
[2] Self-relocate    C:\Windows\System32\<random_subfolder>\<MS_binary_name>.exe
[3] Process critical  NtSetInformationProcess BreakOnTermination (kill = BSOD)
[4] Defense disable  Kill 39 processes, stop 16 services, IFEO cmd.exe → svchost.exe
[5] Persistence      6 layers: Run keys, Winlogon shell, service, SafeBoot survival
[6] Boot destruction BIOS MBR + UEFI bootmgfw.efi overwrite
[7] Recovery disable vssadmin delete, bcdedit safeboot minimal, registry tamper
[8] Network isolate  50 sites to hosts file (incl. torproject.org auto-sabotage)
[9] UI display       HTA multilingual ransom note + wallpaper + text note
[10] Keyboard lock   WH_KEYBOARD_LL hook swallows all WM_KEYDOWN
[11] Encrypt         Intermittent AES-GCM (>1.1MB: first 500KB + random 50KB + last 500KB)
[12] MFT wipe        Parallel AES-CTR destruction of filesystem metadata
[13] Lateral move    SMB + WMI propagation to discovered subnets (50 thread pool)
[14] DDoS flood      320 threads target local gateway + external host
[15] Force shutdown  Kill explorer, 4-minute timer, shutdown /s /f
```

## Lateral Movement

Network propagation uses SMB admin shares with WMI remote execution:

### Subnet discovery

- PowerShell `Get-WmiObject Win32_NetworkAdapterConfiguration` for all /24 subnets
- Ping sweep `.1` to `.255` with 50 concurrent threads
- Continuous 10-second rescan loop

### Infection vector

```
# Primary: C$ share
\\<ip>\C$\Users\Public\<10_random_chars>.exe
powershell -Command "Invoke-WmiMethod -Class Win32_Process -Name Create -ArgumentList
'<remote_path>' -ComputerName <ip>"

# Fallback: ADMIN$ share
\\<ip>\ADMIN$\<random>.exe
powershell -Command "Invoke-CimMethod ..."
```

### USB propagation

- 2-second polling for new removable drives
- Self-copy to drive root as `<SystemDir>_<8_random>.exe`
- HIDDEN+SYSTEM+READONLY attributes

# Detection & Response

## Priority IOCs

| Type | Value |
|---|---|
| File extension | `.Chained` (appended to original) |
| File header | `CHAINED_6617-382+819=...!_(13RANDOM)` (38 bytes) |
| Ransom note | `PLEASEREADTHIS.txt` (Desktop + C:\Users\Public) |
| HTA interface | `IronChain.hta` (C:\Users\Public) |
| Wallpaper artifact | `%TEMP%\IronChain_wallpaper.bmp` (PNG content, 1920×1080) |
| Infection marker | `C:\ProgramData\IRONCHAIN\time.dat` |
| Service name | `IronChainSecurity` |
| Mutex | `IronChainMutex_<8_chars>` |

## High-signal behaviors

- `bcdedit /set {default} safeboot minimal` from non-administrative tool
- Registry write to `HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\{Minimal,Network}\IronChainSecurity`
- `NtSetInformationProcess` with `ProcessBreakOnTermination` (value 29)
- IFEO registry manipulation: `cmd.exe` → `Debugger=svchost.exe`
- Raw disk access to `\\.\PhysicalDrive0` with write operations
- Raw volume access to `\\.\C:` with large sequential writes (MFT region)
- WH_KEYBOARD_LL hook installation with message pump
- Mass SMB share access to `\\<subnet>\C$\Users\Public\` or `\\<subnet>\ADMIN$\`
- PowerShell WMI remote process creation across multiple hosts
- Hosts file modification blocking security vendor domains

## Response priorities

1. **Do not pay**: Payment cannot recover files due to fundamental cryptographic flaws. The operator lacks the means to decrypt even with full cooperation.
2. **Immediate isolation**: Block SMB traffic from infected hosts to prevent lateral spread. The 50-thread subnet scanner can compromise an entire network segment rapidly.
3. **Boot media preparation**: Machines are unbootable after infection. Prepare external boot environments for critical systems before attempting recovery.
4. **MFT forensics**: For partially encrypted systems, attempt MFT reconstruction from disk backups before the 1 GB randomization completes.
5. **Memory acquisition**: If the process is still running, capture full memory dumps. The RSA private key exists in process space but will be lost on termination.

## Threat Positioning

IronChain 3.0 exhibits hallmarks of destructive malware masquerading as ransomware. The operational profile demonstrates advanced understanding of specific Windows techniques, but with critical engineering flaws throughout:

- **Persistence robustness**: Six independent autostart mechanisms including SafeBoot survival suggest experience with enterprise remediation efforts
- **Lateral movement breadth**: SMB + WMI propagation with continuous subnet scanning indicates domain-aware threat modeling
- **Anti-recovery depth**: MFT destruction parallel to encryption, boot loader overwrite, and keyboard input blocking show comprehensive denial-of-service intent

The cryptographic implementation reveals fundamental misunderstanding of recovery requirements:

- **Multi-layer cipher**: Applying random transformations with discarded parameters suggests confusion between obfuscation and encryption
- **No key management**: Runtime RSA generation without exfiltration indicates no planned victim recovery workflow
- **Self-sabotage**: Blocking torproject.org prevents victims from accessing the payment portal, undermining any financial motive

The technical sophistication mismatch between operational tradecraft (high) and cryptographic engineering (broken) suggests either:

1. **Destructive intent**: The "ransomware" framing is cover for a wiper designed to destroy data while creating plausible financial motive
2. **Incompetent monetization**: Skilled operators attempted to add payment extraction to existing destructive tools without understanding encryption requirements

The auto-sabotage bugs (Tor blocking, shift value discarding, boot destruction timing) lean toward scenario 1: intentional destruction with ransomware theater.

---

# MITRE ATT&CK

| ID | Technique |
|---|---|
| T1486 | Data Encrypted for Impact |
| T1485 | Data Destruction (MFT wipe) |
| T1561.002 | Disk Structure Wipe (MBR/UEFI overwrite) |
| T1490 | Inhibit System Recovery (vssadmin delete, bcdedit) |
| T1562.001 | Disable or Modify Tools (kill security processes) |
| T1547.001 | Boot or Logon Autostart (Registry Run Keys) |
| T1547.004 | Winlogon Helper DLL (Shell replacement) |
| T1543.003 | Create or Modify System Process (service install) |
| T1112 | Modify Registry (IFEO, policies, SafeBoot) |
| T1083 | File and Directory Discovery |
| T1057 | Process Discovery |
| T1007 | System Service Discovery |
| T1135 | Network Share Discovery |
| T1018 | Remote System Discovery (subnet scan) |
| T1021.002 | SMB/Windows Admin Shares |
| T1047 | Windows Management Instrumentation |
| T1059.001 | PowerShell |
| T1055 | Process Injection (via remote WMI) |
| T1570 | Lateral Tool Transfer |
| T1091 | Replication Through Removable Media (USB) |
| T1499.004 | Application or System Exploitation (DDoS flood) |
| T1056.001 | Keylogging (keyboard hook) |
| T1491.001 | Internal Defacement (wallpaper) |
| T1027 | Obfuscated Files (PyInstaller packing) |
| T1070.004 | File Deletion (original file removal) |

# Verdict

| Axis | Score |
|---|---|
| Operational sophistication | Advanced but inconsistent |
| Cryptographic engineering | Fundamentally broken |
| Recovery feasibility | Zero (by design or incompetence) |
| Lateral movement capability | High |
| Persistence robustness | High |
| Business disruption impact | Critical (permanent) |

**IronChain 3.0 should be treated as a destructive wiper, not recoverable ransomware.** The cryptographic flaws are not implementation bugs but architectural impossibilities. No payment amount can recover victim data due to mathematical irreversibility of the transformation chain.

The combination of advanced operational techniques with fundamentally broken recovery mechanisms and critical engineering flaws suggests either malicious intent to destroy data under ransomware cover, or extraordinary incompetence in threat monetization. Given the number and severity of auto-sabotage bugs, deliberate destruction appears more likely.

Organizations should prioritize containment and recovery from backups rather than payment negotiation. The threat actor cannot honor recovery promises regardless of intent.