Ironchain

Technical Analysis — V3

REVERSE-ENGINEERED REPORT

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

Sample SHA-256: `2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22`

IronChain 3.0 Ransomware — Full Analysis

1. Sample Identification

Field	Value
Family	IronChain 3.0
SHA-256	`2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22`
MD5	`71318a233ae849db62eb4037486ba3ff`
Type	PE32+ x64, Windows GUI
Size	11,282,718 bytes (11.0 MB)
Language	Python 3.14.4
Compile timestamp	`0x69ac4ccc`
PDB path	None (PyInstaller bootloader)
Image base	`0x140000000` (default MinGW-w64)
Sections	7: `.text`, `.rdata`, `.data`, `.pdata`, `.fptable`, `.rsrc`, `.reloc` (~350 KB)
Overlay	10,933,534 bytes (PyInstaller CArchive)
Functions	53 Python functions + 1 lambda + 1 ctypes.Structure

IronChain 3.0 is a Python 3.14 ransomware distributed as a PyInstaller-packed Windows executable. The binary consists of a minimal PE32+ bootloader with the entire ransomware logic contained in a Python 3.14 bytecode module (`IronChain.pyc`) within the overlay archive. No obfuscation is applied — the bytecode disassembles to readable Python logic with clear string literals and function names.

Archive Contents

Component	Size	Purpose
IronChain.pyc	367 KB	Main ransomware module (53 functions)
python314.dll	2.0 MB	Python 3.14 runtime
PyCrypto dependencies	1.4 MB	AES, RSA, PKCS1_OAEP implementations
Standard library	4.2 MB	subprocess, threading, winreg, ctypes
Other modules	2.9 MB	concurrent.futures, base64, datetime

2. Infrastructure

Field	Value
Onion	`ironchaindecrypt7xfzq5tclm9jzpwq72uofgy2znkdsxm54zbcu2yid.onion`
Email	None observed
TOX ID	`1A51DCBB33FBF603B385D223F599C6D64545E631F7C870FFEA320D84CE5DAF076C1F94100B5B`
Chat URL	`http://ironchaindecrypt7xfzq5tclm9jzpwq72uofgy2znkdsxm54zbcu2yid.onion/chat/<unique_id>`
Note filename	`PLEASEREADTHIS.txt`
Extension	`.Chained`
Mutex	`IronChainMutex_<8_random_chars>`
Payment	Bitcoin, 1000 USD initial, doubles every 24 hours

---

3. Ransom Note

Filename

`PLEASEREADTHIS.txt` — dropped on Desktop and `C:\Users\Public\`, opened automatically via `os.startfile()`

Content (from bytecode strings @ multiple addresses)

```
IRONCHAIN RANSOMWARE

Your computer has been encrypted by IronChain 3.0.

All your important files (documents, photos, videos, databases, and other files) have been
encrypted with military-grade encryption and unique key for your computer.

WHAT HAPPENED TO MY COMPUTER?
We encrypted all your files and you cannot access them anymore. Nobody can recover your files
without our decryption software.

HOW TO RECOVER FILES?
The only way to recover your files is to buy decrypt tool and unique key for your computer.
This software will decrypt all your encrypted files.

WHAT GUARANTEES DO I HAVE?
We value our reputation in the security community. If you do not recover your files, nobody will
pay us in the future, so it is in our own interests to get your files back to you.

You have 72 hours to contact us and make payment. After 72 hours, your unique key will be deleted
from our servers and your files will be lost forever.

To prove that we can recover your files, you can send us 1 small file (less than 1MB) and we will
decrypt it for free.

CONTACT:
1. Download Tor Browser from: https://www.torproject.org/
2. Visit: ironchaindecrypt7xfzq5tclm9jzpwq72uofgy2znkdsxm54zbcu2yid.onion
3. Enter your unique ID: [UNIQUE_ID_PLACEHOLDER]

Alternative contact:
TOX ID: 1A51DCBB33FBF603B385D223F599C6D64545E631F7C870FFEA320D84CE5DAF076C1F94100B5B

IMPORTANT WARNINGS:
* Do not rename encrypted files.
* Do not try to decrypt your data using third party software, it may cause permanent data loss.
* Decryptors of other vendors are incompatible and may destroy your files forever.

Your unique ID: [UNIQUE_ID_PLACEHOLDER]
```

HTA Interface

11 KB multilingual HTML Application (`IronChain.hta`) with EN/ES/FR/DE support, countdown timers (72h deadline + 24h doubling), and contact information. Launched via `mshta.exe` and designed as always-on-top single-instance interface.

4. Execution Flow (`<module>` @ `IronChain.pyc:1`)

1. Single-instance mutex check (`IronChainMutex_<8random>`)
2. Command-line parsing (`--old-path` for self-deletion)
3. Admin elevation check + UAC re-launch if needed
4. RSA-4096 keypair generation (runtime, never exfiltrated)
5. Self-relocation to System32 subfolder with Microsoft binary masking
6. Original process termination via WMIC
7. Process criticality setting (kill = BSOD via NtSetInformationProcess)
8. Defense evasion (kill 39 processes, stop 16 services)
9. Persistence installation (6 layers: Run, Winlogon, Service, SafeBoot)
10. Recovery inhibition (VSS deletion, bcdedit safeboot minimal)
11. Install time anchor creation (`C:\ProgramData\IRONCHAIN\time.dat`)
12. Boot destruction (MBR + UEFI overwrite)
13. Ransom note generation and display (text + HTA)
14. Self-protection (HIDDEN+SYSTEM attributes, restrictive ACLs)
15. Network isolation (hosts file blocking 50 security/tool sites)
16. Background threads launch (6 daemons)
17. DDoS flood start (320 threads targeting gateway + external)
18. CD-ROM eject attempt (broken implementation)
19. Keyboard hook installation (WH_KEYBOARD_LL input blocking)
20. Message pump (maintains hook, ~11 days or until forced shutdown)

Thread Architecture

Main Thread: Message pump maintaining keyboard hook
Daemon Threads (6): - `dos` : Self-deletion cleanup (10s delay) - `ew` : Encryption orchestrator (ThreadPoolExecutor, 16 workers) - `nw` : Network lateral movement (10s rescan loop) - `uw` : USB propagation watcher (2s polling) - `em` : MFT destruction (parallel to encryption) - `fc` : Force shutdown (4min delay, kill explorer/notepad, shutdown)

Worker Pools: - Encryption: 16 threads via concurrent.futures - Network propagation: 50 threads for subnet scanning - DDoS: ~320 daemon threads (UDP/HTTP flood)

---

5. Encryption System

Key Exchange

Parameter	Value
Algorithm	RSA-OAEP
Key size	4096 bits
Implementation	PyCrypto `RSA.generate(4096)`
Runtime generation	Fresh keypair per execution
Attacker public key	Generated at runtime (never hardcoded)

Critical Flaw: The RSA private key exists only in process memory (`rsa_key` global) and is never serialized, transmitted, or stored. Process termination permanently destroys the only means of decryption.

Symmetric Cipher

Parameter	Value
Algorithm	AES-GCM
Mode	GCM (Galois/Counter Mode)
Key size	256 bits
Nonce/IV	16 bytes (random per chunk)
Per-file key	YES — `get_random_bytes(32)` per file

Multi-Layer Cipher (`mc` @ `IronChain.pyc:745`)

Critical Flaw: Before AES-GCM, files undergo a destructive transformation:

```
def mc(data):
    layers = random.randint(2, 7)        # 2-7 random Caesar layers
    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% probability per byte
                result[i] = (result[i] + shift) & 0xFF
    return bytes(result), shifts
```

The calling function `ef()` discards the `shifts` return value. These parameters are never stored and `random` is not seeded, making the transformation mathematically irreversible.

File Encryption Process (`ef` @ `IronChain.pyc:760`)

1. **Open file** for binary read, check exclusions
2. **Generate AES key** via `get_random_bytes(32)`
3. **Apply mc() transformation** (irreversible, shifts discarded)
4. **Intermittent encryption logic:** - Files ≤ 1,153,434 bytes: Full encryption (single AES-GCM) - Files > 1,153,434 bytes: Three chunks
    - First 500 KB
    - Random 50 KB (position between first_size and filesize-last_size-50KB)
    - Last 500 KB
5. **RSA-OAEP wrap** AES key with runtime public key
6. **Write encrypted file** with header + wrapped key + encrypted chunks
7. **Delete original** via `os.remove()`

Intermittent Encryption

- Files ≤ **1,153,434 bytes** (~1.1 MB): Full encryption
- Files > **1.1 MB**: First 500 KB + random middle 50 KB + last 500 KB
- **Plaintext preservation**: Middle sections between encrypted chunks remain unmodified

Encrypted File Format

```
[HEADER: 38 bytes]
  Marker: "CHAINED_6617-382+819=...!_(13RANDOM)"

[KEY_LENGTH: 2 bytes little-endian]

[WRAPPED_KEY: 512 bytes for RSA-4096]

--- For intermittent files (>1.1MB) ---
[NONCE_FIRST: 16 bytes][TAG_FIRST: 16 bytes]
[NONCE_MID: 16 bytes][TAG_MID: 16 bytes]
[NONCE_LAST: 16 bytes][TAG_LAST: 16 bytes]
[ENCRYPTED_DATA: chunks with plaintext gaps]

--- For small files (≤1.1MB) ---
[NONCE: 16 bytes][TAG: 16 bytes]
[ENCRYPTED_DATA: complete]
```

---

6. File Targeting

Targeted Directories

Phase 1 — User Folders (8):
Desktop, Downloads, Documents, Pictures, Videos, Music, Favorites, 3D Objects

Phase 2 — Critical System Files:
Boot managers, system executables, critical DLLs (deliberately targeted to brick system)

Phase 3 — System Directories:
`%SystemRoot%\System32`, `%SystemRoot%`, `%ProgramFiles%`, `%ProgramFiles(x86)%`

Phase 4 — Additional Drives:
All fixed, removable, and CD-ROM drives (D:, E:, F:, etc.) — C: skipped as covered in phases 1-3

Excluded Directories

None implemented — the `was()` walker function applies no directory exclusions. All discovered directories are recursively processed.

Excluded Files

File/Pattern	Reason
Files with `.Chained` extension	Already encrypted check
`PLEASEREADTHIS.txt`	Ransom note preservation
Files with `.hta` extension	HTA interface preservation
Basename `icBG`	Legacy wallpaper exclusion
Already encrypted (header check)	Duplicate encryption prevention
SELF_PATH variations	Dropper self-preservation

Critical System Targeting

Deliberately targeted for destruction: - `bootmgr`, `bootmgr.exe` - `winload.exe`, `winload.efi` - `winresume.exe`, `winresume.efi` - `taskmgr.exe`, `regedit.exe`, `services.msc` - `winlogon.exe`, `services.exe`, `lsass.exe` - `advapi32.dll`, `cfgmgr32.dll`

This targeting ensures system unbootability and critical process failure, supporting the assessment that IronChain functions as a wiper rather than recoverable ransomware.

---

7. Recovery Inhibition

Commands executed via `rh()` / `rhl()` subprocess wrappers:

Command	Address	Description
`wbadmin delete catalog -quiet`	`db:169`	Remove Windows backup catalog
`vssadmin delete shadows /all /quiet`	`db:169`	Delete Volume Shadow Copies
`bcdedit /set {default} recoveryenabled No`	`db:169`	Disable Windows Recovery
`bcdedit /set {default} bootstatuspolicy ignoreallfailures`	`db:169`	Ignore boot failures
`bcdedit /set {default} safeboot minimal`	`db:169`	Force SafeBoot mode permanently
`reg add HKLM\...\SystemRestore /v DisableSR /d 1`	`db:169`	Disable System Restore
`reg add HKLM\...\SystemRestore /v DisableConfig /d 1`	`db:169`	Block restore configuration
`sc config swprv start= disabled`	`db:169`	Disable Volume Shadow Provider
`sc stop swprv`	`db:169`	Stop shadow provider service

Additional MFT Destruction (`em` @ `IronChain.pyc:1258`):

Raw volume access to `\\.\C:` with AES-CTR overwrite of 1 GB NTFS Master File Table region. The encryption key (`get_random_bytes(32)`) is never stored, ensuring permanent filesystem metadata corruption.

---

8. Targeted Services (16)

Stopped via `net stop` + disabled via `sc config` + deleted via `sc delete`:

Security services: AVP, KAV, BDSS, Bitdefender, avast! Antivirus, avast! Service, AVG, MBAMService, DrWeb, wscsvc (Security Center), SecurityHealthService, Sense (Defender ATP), WinDefend, WdNisSvc, MpKslDrv, NisSrv

Implementation: `ss()` @ `IronChain.pyc:195` — 16 hardcoded service names processed sequentially with error suppression.

---

9. Targeted Processes (39)

Terminated via `taskkill /f /im`:

Administrative tools: taskmgr, regedit, cmd, msconfig, mmc, services.msc, control, SystemSettings, WindowsSecurity
Process analysis: procexp, procexp64
Security products: MsMpEng (Defender), AVP/KSDE/KAVSVC/AVPUI (Kaspersky), BDAGENT/BDSERVICEHOST/BDREDLINE/BDWTXAG (Bitdefender), AVASTUI/AvastSvc/AvastEmUpdate (Avast), AVGUI/AVGSvc/avgwdsvc (AVG), MBAM/mbamtray/MBAMService/mbamguard (Malwarebytes), dwengine/dwservice/dwagent (Dr.Web)
Command-line tools: powershell, pwsh, taskkill, wmic, diskpart, bootrec

Implementation: `kp()` @ `IronChain.pyc:191` — 39 hardcoded process names with forced termination (`/f` flag).

---

10. Persistence & Evasion

Self-Relocation (`mt` @ `IronChain.pyc:56`)

Target: `C:\Windows\System32\<random_subfolder>\<microsoft_binary_name>.exe`
Masking: Copies self with legitimate Microsoft executable names for camouflage
Process: Creates batch script for original deletion, launches relocated copy with `--old-path` argument

Persistence Layers (6)

Mechanism	Registry Path	Description
Run Keys	`HKCU/HKLM\...\Run\IronChain`	Standard autostart
Winlogon Shell	`HKCU/HKLM\...\Winlogon\Shell`	Replace Windows shell
Service	`HKLM\...\Services\IronChainSecurity`	Windows service (auto-start)
Service Critical	Service failure actions	5s/5s/5s restart + critical flag
SafeBoot Minimal	`HKLM\...\SafeBoot\Minimal\IronChainSecurity`	Safe mode persistence
SafeBoot Network	`HKLM\...\SafeBoot\Network\IronChainSecurity`	Network safe mode persistence

Process Protection

Critical Process Flag: `mpc()` @ `IronChain.pyc:155` `NtSetInformationProcess(handle, 29, &flag, 4)` with `ProcessBreakOnTermination`
Effect: Terminating the process triggers BSOD (`CRITICAL_PROCESS_DIED`)

Registry Tampering (`dci` @ `IronChain.pyc:257`)

Image File Execution Options: - `HKLM/HKCU\...\IFEO\cmd.exe` → `Debugger=svchost.exe` - **Effect:** All cmd.exe executions redirect to svchost.exe

Policy Restrictions: | Policy | Registry Value | Effect | |---|---|---| | DisableTaskMgr | 1 | Block Task Manager | | NoRun | 1 | Disable Run dialog | | DisableRegistryTools | 1 | Block regedit access | | NoControlPanel | 1 | Hide Control Panel | | EnableLUA | 0 | Disable UAC | | DisallowRun | 1-11 entries | Block specific tools |

Defender Disabling

Registry tampering: `DisableAntiSpyware = 1` in Windows Defender policies
Service termination: WinDefend, WdNisSvc, SecurityHealthService, Sense
Process killing: MsMpEng
Exclusion addition: PowerShell `Add-MpPreference -ExclusionPath` for dropper location

Keyboard Input Blocking (`hp` @ `IronChain.pyc:1313`)

Hook: `SetWindowsHookEx(WH_KEYBOARD_LL, hp, ...)` with global low-level keyboard hook
Callback: Returns `1` for all `WM_KEYDOWN` messages
Effect: All keyboard input swallowed — user cannot type in any application

Network Isolation (`bw` @ `IronChain.pyc:1054`)

Hosts file modification: `C:\Windows\System32\drivers\etc\hosts`
Blocked domains (50): Security vendors, update servers, tutorial sites, and **critically: torproject.org**
Auto-sabotage bug: Torproject.org is blocked despite being needed for payment access

Anti-Analysis Summary

Technique	Unprotect ID	Address	Description
Process Criticality	U0527	`mpc:155`	NtSetInformationProcess BreakOnTermination
IFEO Redirection	U0064	`dci:257`	cmd.exe → svchost.exe debugger
Keyboard Hook	U0391	`hp:1313`	WH_KEYBOARD_LL input blocking
Hosts File Tamper	U0412	`bw:1054`	DNS redirection to localhost
Service Persistence	U0072	`ip:122`	Critical service with SafeBoot

---

11. Command-Line Arguments

Argument	Description
`--old-path <path>`	Specifies original executable path for deletion cleanup

Usage: Set by self-relocation mechanism (`mt`) to enable cleanup of original dropper location via background thread (`dos`).

---

12. Static Imports Summary

Note: As a PyInstaller Python executable, traditional PE imports are minimal. Functionality is provided through Python bytecode imports:

Category	Key Modules
Crypto	`Crypto.Cipher.AES`, `Crypto.Cipher.PKCS1_OAEP`, `Crypto.PublicKey.RSA`
File I/O	`os`, `shutil`, `tempfile`
Process	`subprocess`, `ctypes.windll.kernel32`
Registry	`winreg`
Network	`socket`, `threading`, `concurrent.futures`
System	`ctypes.wintypes`, `datetime`, `base64`

---

13. IDA Analysis — Python Functions

Note: Analysis performed via Python 3.14 native `dis` module on extracted bytecode. Function addresses reference line numbers in disassembly.

Line	Name	Size	Description
1	<module>	5586	Main execution flow + global setup
21	ia	106	IsUserAnAdmin check via ctypes
38	rh	142	Run hidden subprocess (CREATE_NO_WINDOW)
44	rhl	140	Run hidden list-form subprocess
56	mt	1618	Move to System32, Microsoft name masking
105	tos	600	Terminate old self via WMIC process kill
122	ip	1012	Install persistence (6 mechanisms)
155	mpc	478	Make process critical (BSOD on termination)
169	db	204	Disable backups, force SafeBoot
180	rs	428	Registry set wrapper with fallback
191	kp	54	Kill 39 processes via taskkill
195	ss	114	Stop 16 services + disable + delete
201	das	652	Delete AV startup registry entries
257	dci	502	Disable critical interfaces (IFEO, policies)
321	git	692	Get install time from persistent marker
341	gi	238	Get local IP via socket route trick
351	crn	788	Create ransom note text + drop + open
389	chr	1152	Construct HTML ransom (HTA) + launch
690	om	406	Overwrite MBR with boot message
724	ou	516	Overwrite UEFI bootmgfw.efi
745	mc	398	Multi-layer cipher (shifts discarded)
760	ef	3696	Encrypt file (main encryption logic)
875	gsf	548	Get special folders (8 user directories)
888	gcsp	1888	Get critical system paths (deliberate targets)
916	was	212	Walk and skip (recursive directory walker)
925	ew	1538	Encrypt walk (orchestrates 4-phase encryption)
968	gad	368	Get all drives (fixed, removable, CDROM)
982	gcd	364	Get CDROM drives only
995	ec	84	Eject CDs (broken implementation)
1003	grd	364	Get removable drives only
1016	mpl	148	Make path long (\?\ prefix)
1023	to	152	Take ownership via takeown + icacls
1031	sfan	262	Set file attributes normal (clear RO/HIDDEN)
1040	ssa	396	Self-set attributes (hide + protect dropper)
1054	bw	470	Block websites in hosts file

Line	Name	Size	Description
1079	swfb	392	Set wallpaper from base64 PNG
1097	dos	316	Delete old self (cleanup background thread)
1109	gdg	320	Get default gateway via route print
1121	uf	228	UDP flood (DDoS component)
1133	hf	258	HTTP flood (DDoS component)
1146	sf	982	Start flood (orchestrates 320 DDoS threads)
1168	ctu	866	Copy to USB (propagation on insertion)
1183	gls	442	Get local subnets via WMI
1198	p	122	Ping (connectivity test)
1205	ir	984	Infect remote (SMB + WMI lateral movement)
1224	pnw	398	Propagate network (50-thread subnet scanner)
1237	nw	88	Network wrapper (10s rescan loop)
1245	uw	184	USB watcher (2s polling for new drives)
1258	em	1168	Erase MFT (1GB AES-CTR filesystem destruction)
1285	<lambda>	46	AES-CTR counter for MFT destruction
1304	KBDLLHOOKSTRUCT	188	ctypes structure for keyboard hook
1313	hp	136	Hook procedure (keyboard input blocker)
1325	fc	168	Final cleanup (4min delay + force shutdown)

14. Indicators of Compromise (IOCs)

Hashes

Type	Value
SHA-256	2d57b05e8fbcae3da196ed972074577fa6604ba7f3afcadbf79b94c8f8f6de22
MD5	71318a233ae849db62eb4037486ba3ff

Network

Type	Value
Onion	ironchaindecrypt7xfzq5tclm9jzpwq72uofgy2znkdsxm54zbcu2yid.onion
TOX ID	1A51DCBB33FBF603B385D223F599C6D64545E631F7C870FFEA320D84CE5DAF076C1F94100B5B

Files

Indicator	Value
Ransom note	`PLEASEREADTHIS.txt`
HTA interface	`IronChain.hta`
Encrypted extension	`.Chained`
Infection marker	`C:\ProgramData\IRONCHAIN\time.dat`
Wallpaper artifact	`%TEMP%\IronChain_wallpaper.bmp`
Self-relocated path	`C:\Windows\System32\<random>\<MS_name>.exe`

Registry

Key	Description
`HKLM\SYSTEM\...\Services\IronChainSecurity`	Persistence service
`HKLM\SYSTEM\...\SafeBoot\Minimal\IronChainSecurity`	SafeBoot persistence
`HKLM/HKCU\...\Run\IronChain`	Autostart persistence
`HKLM/HKCU\...\IFEO\cmd.exe`	Debugger=svchost.exe
`HKLM\...\Policies\Microsoft\Windows Defender\DisableAntiSpyware`	Defender disable

Behavioral

- Raw disk write to `\\.\PhysicalDrive0` (MBR overwrite)
- Raw volume write to `\\.\C:` at MFT offset (filesystem destruction)
- `bcdedit /set {default} safeboot minimal` execution
- Mass `taskkill /f /im` across 39 process names
- `net stop` + `sc delete` across 16 service names
- WH_KEYBOARD_LL hook with input swallowing
- SMB share access to `\\<IP>\C$\Users\Public\` across subnet ranges
- PowerShell WMI remote process creation (`Invoke-WmiMethod`)
- Hosts file modification blocking 50 domains including torproject.org

Distinctive Strings

- `"CHAINED_6617-382+819=...!_(13RANDOM)"` (file header)
- `"IronChainMutex_"` (mutex prefix)
- `"IRONCHAIN v3.0 - Your data is our business"` (HTA footer)
- `"CHAINED - PAY TO DECRYPT"` (MBR message)
- `"IronChainSecurity"` (service name)

---

15. MITRE ATT&CK Mapping

ID	Technique	Implementation
T1486	Data Encrypted for Impact	AES-GCM intermittent encryption with RSA-OAEP key wrap
T1485	Data Destruction	MFT overwrite via AES-CTR with lost key
T1561.002	Disk Structure Wipe	MBR + UEFI bootmgfw.efi overwrite
T1490	Inhibit System Recovery	vssadmin delete, bcdedit modifications
T1562.001	Disable or Modify Tools	Kill 39 processes, stop 16 services
T1112	Modify Registry	IFEO, policies, persistence keys, Defender disable
T1547.001	Boot or Logon Autostart Execution	Run keys, Winlogon shell override
T1543.003	Create or Modify System Process	Service creation with critical flag
T1055.012	Process Hollowing	Remote WMI process creation for lateral movement
T1021.002	SMB/Windows Admin Shares	C$ and ADMIN$ share access for propagation
T1047	Windows Management Instrumentation	WMI Win32_Process Create for remote execution
T1059.001	PowerShell	WMI and propagation commands via PowerShell
T1087.002	Domain Account Discovery	WMI Win32_NetworkAdapterConfiguration enumeration
T1135	Network Share Discovery	SMB share enumeration and access
T1018	Remote System Discovery	Subnet ping sweep with 50 concurrent threads
T1570	Lateral Tool Transfer	Self-copy to remote admin shares
T1091	Replication Through Removable Media	USB propagation on device insertion
T1499.004	Application or System Exploitation	DDoS flood with 320 threads
T1056.001	Keylogging	WH_KEYBOARD_LL hook (blocking, not logging)
T1491.001	Internal Defacement	Desktop wallpaper replacement
T1027	Obfuscated Files or Information	PyInstaller packaging
T1070.004	File Deletion	Original executable cleanup
T1083	File and Directory Discovery	Recursive directory walking

---

16. Summary

IronChain 3.0 represents a sophisticated destructive malware masquerading as ransomware. While the operational tradecraft demonstrates advanced understanding of Windows persistence, lateral movement, and anti-forensics techniques, the cryptographic implementation contains fundamental flaws that prevent file recovery regardless of payment.

Key Findings:

1. **Cryptographic Impossibility:** The multi-layer cipher discards shift values and the RSA private key is never exfiltrated, making decryption mathematically impossible even with operator cooperation.

2. **Destructive Intent:** Parallel MFT destruction, deliberate system file targeting, and boot loader overwrite before encryption completion indicate wiper functionality rather than profit-motivated ransomware.
3. **Operational Sophistication:** Six-layer persistence including SafeBoot survival, comprehensive lateral movement via SMB/WMI, and 320-thread DDoS anti-recovery demonstrate advanced threat actor capabilities.
4. **Self-Sabotage:** Multiple bugs including Tor portal blocking suggest either intentional destruction or extraordinary incompetence in monetization implementation.

IronChain 3.0 should be classified as a destructive wiper with ransomware theater. Organizations should prioritize containment and backup recovery rather than payment negotiation, as the threat actor lacks the technical means to restore encrypted data.