

Name – Soham Kadu

Intern ID – 235

#### Tool Name:

**GoStringUngarbler + HashMyFiles**

---

#### History:

- **GoStringUngarbler** is a community-developed Go-based tool used in malware analysis to decode garbled or obfuscated strings found in binary files.
  - **HashMyFiles** is a trusted Windows utility from NirSoft, active for over a decade, used to generate cryptographic hashes of files.
- 

#### Description:

A powerful combination of two lightweight tools. GoStringUngarbler is used to extract and decode suspicious strings (such as URLs, file paths, C2 domains) from malware or binaries. HashMyFiles is used to compute and verify file hashes (MD5, SHA1, SHA256) for integrity verification or forensic tracking.

---

#### What Is This Tool About?

These tools are commonly used in malware analysis, reverse engineering, and digital forensics.

- GoStringUngarbler helps investigators decode hidden or obfuscated strings in malicious executables.
  - HashMyFiles is used to verify the integrity of these suspicious files or create hash reports for forensic evidence.
- 

#### Key Characteristics / Features:

##### **GoStringUngarbler**

1. Written in Go, open source
2. Supports common obfuscation decoding
3. Works on binary dumps, strings files
4. Fast, CLI-based
5. Supports decoding with custom logic
6. Minimal dependencies

### **HashMyFiles**

1. Portable – no install needed
  2. Supports MD5, SHA1, SHA256, SHA384
  3. Export results in HTML, TXT, CSV
  4. Drag & drop interface
  5. Supports folder & batch hashing
  6. Easy integration into forensic workflows
- 

### **Types / Modules Available:**

- GoStringUngarbler CLI tool
  - Obfuscation decoding modules (e.g., Base64, XOR)
  - HashMyFiles GUI & Command-line versions
  - Context menu integration in Windows
  - Timestamp & hash export modules
- 

### **How Will This Tool Help?**

- Extracts hidden strings from malware
  - Decodes hardcoded indicators (C2s, payloads)
  - Verifies file integrity for chain of custody
  - Helps in malware variant comparison
  - Generates quick hash reports for evidence
  - Easy to automate in investigation scripts
-

## Proof of Concept (PoC) Images:

HashMyFiles										
Filename	MDS	SHA1	File Version	Product Ve...	Identical	Extension	File Attribu...	Hash Start Time	Hash End Time	Hashing Duration
monkey-g46eb24a10...	369a65a313b448faf8cc31f2e28c6a4f	a9246970ed34869d3f2fa9f03fed207de480f						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.011
Setup.log	40ccb65a56b7d27a001283ee8f8267c5	7cbfb3f449eb9d9205dbd22d9089fcffdf34l						24-07-2025 20:25:51	24-07-2025 20:25:51	00:00:00.002
download.jpeg	cee4c07c427eaef32a16978ed89bcf13a	776b1e7db96a154f2245b7ce9b5af59f997						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.003
download (1).png	340227e4247755f0fcf67910ca39555b	5ad51d5dbbac46d6f18d2214df781faaebe						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.010
download.png	340227e4247755f0fcf67910ca39555b	5ad51d5dbbac46d6f18d2214df781faaebe						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.002
download (1).jpeg	7f5e22dc0506fc55755c77d287cef67	55691ddee0b1be605e2543b028645a822a5						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.010
_122074265_hi0718438...	c046038e9312d01bf3a5ab81fb05a3f8	07a43a444eff2b3bac2c54f8704ea9be4353l						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.003
_122074265_hi0718438...	c046038e9312d01bf3a5ab81fb05a3f8	07a43a444eff2b3bac2c54f8704ea9be4353l						24-07-2025 20:32:21	24-07-2025 20:32:21	00:00:00.003

HashMyFiles					
Filename	MDS	SHA1	CRC32	SHA-256	SHA-512
House of Biryan.mp4	89d9eef16768cbe46dfc75fbf882df29	ccfe76496de477cd2f286a380106962ed371...	c1edb5fa	81bfd9a1fa29808c4f9a0124ea47176f4609f6...	a4a600db022cfcdce39b42f
Cheesewala.mp4	989641ca6a069fb0fb76fb1bd1b9dd0634	cbef8af43f532ed0f94e73a2661ccdd68fa1670e	a056f3ab	19ba3494566a44d86bb6d4b71f91eb0a58b92...	24252f5e3b8d8be6579d89e
monkey-g46eb24a10...	369a65a313b448faf8cc31f2e28c6a4f	a9246970ed34869d3f2fa9f03fed207de480f4b	2211ccdf	1fa61f4403a70c4588b9d0c72cc3cd10da3...	7e5a6760c8ba569c6ebab5
Blue Bop Cafe.mp4	04cef54e1d1cce6936e2e662ea6af4383	81c7f089438901bbd41097b70da4af9f8b169e	2ccb2d63	cfbd1f2fbfa7dc91f8332d83d11b3b2bbd32a7...	1c1edcc56822f904fcc58f6
Setup.log	40ccb65a56b7d27a001283ee8f8267c5	7cbfb3f449eb9d9205dbd22d9089fcffdf340acd	a3d5d826	9ee7f632c53d3c3929eb4223a94a6f3602332...	a3268ce1f05c3a5b4b94f
download.jpeg	cee4c07c427eaef32a16978ed89bcf13a	776b1e7db96a154f2245b7ce9b5af59f99704...	a1f65a38	a957cb45cd3e074da182441734f1fb7b62c...	fb240bb8f539f165479584e
download (1).png	340227e4247755f0fcf67910ca39555b	5ad51d5dbbac46d6f18d2214df781faaebe5...	d31b2428	a80731659b3f6c837ccfc70de94fceba4128f2...	bb184fb8974dc2d299e80
download.png	340227e4247755f0fcf67910ca39555b	5ad51d5dbbac46d6f18d2214df781faaebe5...	d31b2428	a80731659b3f6c837ccfc70de94fceba4128f2...	bb184fb8974dc2d299e80
download (1).jpeg	7f5e22dc0b56f55755c77d287cef67	55691ddee0b1be605e2543b028645a822a5f9...	b448acdb	00157defbc533e2154dff41ae1e9b16d5c14e...	14cb3933cf7445931ebfe3
Thank Gourd.mp4	afbdb0e40e72d9969df90dc759be273	44acf7d0d27a58242b6e64d663f980b75ad82...	949fe13c	527064863c390f1f665141c05ebfd4015e30e...	6ee2dc0d0fa24300127a8e6f
Ramashray.mp4	90c2ca9a9cb6b75746bc135636421ae42	19622ec027e63fb90c0d0774f5ae846b04feda	8ea64880	26c972016149708b0834fb37cfbbfba441d2fc...	30d847f5568f2a3670b82d4
_122074265_hi0718438...	c046038e9312d01bf3a5ab81fb05a3f8	07a43a444eff2b3ba2c54f8704ea9b43530646	05863de	ce1afce70f925b05658ad8c74c7d61d89fc0c...	bed5acb1d1d00d820fc99b
_122074265_hi0718438...	c046038e9312d01bf3a5ab81fb05a3f8	07a43a444eff2b3ba2c54f8704ea9b43530646	05863de	ce1afce70f925b05658ad8c74c7d61d89fc0c...	bed5acb1d1d00d820fc99b
Arggo.s.mp4	097b20fb2a68fb7a1d5583dfd7e5a71	009242fb200954fb29ce74c50d8e9256b2a01...	09fb5a5	01a0dd81c4b57556c05333e583b7f3fb5b31...	002a76e058052a2578a958

14 file(s)

NirSoft Freeware. <https://www.nirsoft.net>

```

INFO:root:678 in 693 | result at 0x53c0e0: 'UnlockFileEx'
INFO:root:679 in 693 | result at 0x53cc80: 'NetUserGetLocalGroups'
INFO:root:680 in 693 | result at 0x53cfe0: 'GetProcessMemoryInfo'
INFO:root:681 in 693 | result at 0x53d2c0: 'CreateEnvironmentBlock'
INFO:root:682 in 693 | result at 0x53d5a0: 'DestroyEnvironmentBlock'
INFO:root:683 in 693 | result at 0x53f080: 'file type does not support deadline'
INFO:root:684 in 693 | result at 0x53f420: 'not pollable'
INFO:root:685 in 693 | result at 0x542840: 'internal error: unknown network type '
INFO:root:686 in 693 | result at 0x542be0: 'wsaioclt'
INFO:root:687 in 693 | result at 0x548020: 'os: process already finished'
INFO:root:688 in 693 | result at 0x548940: 'winreadlinkvolume'
INFO:root:689 in 693 | result at 0x548be0: 'pattern contains path separator'
INFO:root:690 in 693 | result at 0x54a1c0: 'syntax error scanning complex number'
INFO:root:691 in 693 | result at 0x552f60: '0123456789ABCDEFX'
INFO:root:692 in 693 | result at 0x554140: '0123456789abcdefx'
INFO:root:693 in 693 | result at 0x5557a0: 'Hello, world.'
INFO:root:Error occurred: 3
INFO:root:Stack obfuscated string count: 360 strings.
INFO:root:Split obfuscated string count: 152 strings.
INFO:root:Seed obfuscated string count: 181 strings.
INFO:root:Processing took 5817 ms!

```

```

// Generate a random key with the same length as the input string
key := make([]byte, len(data))

// Fill the key with random bytes
obfRand.Read(key)

// Select a random operator (XOR, ADD, SUB) to be used for encryption
op := randOperator(obfRand)

// Encrypt each byte of the data with the key using the random operator
for i, b := range key {
    data[i] = evalOperator(op, data[i], b)
}

```

```

1 | __int64 __fastcall sub_48D780()
2 | {
3 |     // [COLLAPSED LOCAL DECLARATIONS. PRESS NUMPAD "+" TO EXPAND]
4 |
5 |     if ( &retaddr <= *(v0 + 16) )
6 |         sub_46A400();
7 |     *data = 0xAF7E8C66BECA0046LL;
8 |     for ( i = 0LL; i < 8; ++i )
9 |         data[i] -= KEY[i];
10 |    return runtime_slicebytetostring(data, 8);
11 |

```

### **15-Liner Summary:**

1. CLI-based decoding with Go
2. GUI-based file hashing
3. Supports malware string extraction
4. Works on Windows/Linux
5. Lightweight and fast
6. Good for YARA rule crafting
7. Easy file hash reporting
8. Export results to TXT, HTML, CSV
9. Useful in static malware analysis
10. Ideal for integrity validation
11. Custom decoding logic support
12. Drag & drop functionality
13. No installation required
14. Low resource usage
15. Open source / freeware

### **Time to Use / Best Case Scenarios:**

- During malware reverse engineering
- Before dynamic analysis or sandboxing
- For file comparison after patching
- In evidence packaging for courts
- Before uploading files to threat intel platforms

---

**When to Use During Investigation:**

- Post-infection artifact analysis
  - Unknown binary behavior tracking
  - Creating IoCs (Indicators of Compromise)
  - Timeline correlation with file hashes
  - Investigating persistence mechanisms
- 

**Best Person to Use This Tool & Required Skills:****Best Users:**

- Malware Analyst
- Digital Forensics Investigator
- Threat Intel Researcher

**Required Skills:**

- Basic command-line usage
  - Familiarity with obfuscation techniques
  - Understanding of file hashing concepts
  - Comfort with static malware analysis
- 

**Flaws / Suggestions to Improve:**

- GoStringUngarbler lacks GUI for beginners
  - Needs more decoding logic (ROT13, AES)
  - HashMyFiles doesn't support recursive sub-folder hashing by default
  - No direct integration with threat intel platforms
  - Needs centralized log generation
- 

**Good About the Tools:**

- Extremely lightweight and portable
- Effective in early-stage malware triage
- No installation needed – plug and play
- High speed for bulk processing

- Reliable outputs for forensic reporting