Resident Evil

Category: Endpoint Threat Hunting
Difficulty: Easy
Date Completed: 2025-05-01

🔍 Summary

This lab involves endpoint detection and threat hunting of the REvil (Sodinokibi) ransomware. The infection chain started with the execution of a trojanized file (`facebook assistant.exe`), followed by PowerShell-based deletion of shadow copies, encryption of user files, and ransom note creation. The analysis included tracing process behavior, extracting hash values, decoding obfuscated payloads, and identifying the attacker's onion domain.

🔄 Timeline of Events

Timestamp	Event
2023-09-07 16:09:50.836	`facebook assistant.exe` executed from `C:\Users\Administrator\Downloads\`
2023-09-07 16:09:53.578	PowerShell command executed to delete shadow copies
2023-09-07 16:09:59.750	Ransom note `C:\Users\Default\5uizv5660t-readme.txt` created
2023-09-08 04:08:53.267	System activity observed under SoftwareDistribution

🔧 Technical Walkthrough

- **Tools Used**: Splunk, Regex, CyberChef, VirusTotal, Tria.ge
- **Artifacts Found**:
  - `facebook assistant.exe` (ransomware loader)
  - Obfuscated PowerShell for shadow copy deletion
  - `readme.txt` ransom note
- **IOCs**:
  - SHA256: `b8d7fb4488c0556385498271ab9fffdf0eb38bb2a330265d9852e3a6288092aa`
  - Onion domain: `aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion`

🖌️ MITRE Mapping

- **Execution**:
  - T1059.001 – Command and Scripting Interpreter: PowerShell
  - T1204.002 – User Execution: Malicious File
- **Defense Evasion**:
  - T1027 – Obfuscated Files or Information
- **Impact**:
  - T1490 – Inhibit System Recovery
  - T1486 – Data Encrypted for Impact

💡 Detection Logic

- **Sourcetype**: `revil`
- Over 1150 events in dataset
- Regex for identifying ransom-related file creation:

```
| regex winlog.event_data.TargetFilename="(?i)(readme|locked|encrypted).*\.((txt)|(html))$"
```

🧐 Lessons Learned

- **What surprised me?**
  The high level of operational detail in REvil's execution chain, especially how it adapts to privilege level and deletes shadow copies intelligently.
- **What would I do differently?**
  Apply optimized Splunk queries and filters upfront to accelerate pivoting.
- **Which new tool or concept did I master?**
  Regex targeting of ransomware indicators in file names and PowerShell decoding via CyberChef.

🧠 Background Before Questions

🔥 OVERVIEW

REvil / Sodinokibi is a highly sophisticated ransomware strain first discovered in **April 2019** by the Cybereason Nocturnus team. It's believed to be developed by the same threat actors behind **GandCrab**, which dominated 40% of global ransomware infections before its retirement.

Key Characteristics:

- Multi-stage execution involving JavaScript, PowerShell, and .NET modules
- Highly evasive: multiple layers of obfuscation and memory-only operations

- Modular design allows easy updates and variant customization
- Privilege-aware: adapts its behavior based on current access level
- Region-aware: avoids infecting Russian/Syrian systems
- Impactful: designed to encrypt files, remove backups, and demand ransom via Tor portals

Initial Access Methods:

- Exploits in public-facing apps (e.g., Oracle WebLogic CVE-2019-2725)
- Phishing emails with malicious ZIP attachments
- Malvertising and exploit kits

🛠️ ATTACK FLOW SUMMARY

1. **Initial Access** – Phishing email w/ ZIP attachment → obfuscated JavaScript
2. **Stage 1** – JavaScript drops and executes PowerShell loader
3. **Stage 2** – PowerShell decodes embedded .NET payload in memory
4. **Privilege Escalation** – UAC bypass using registry hijack and `CompMgmtLauncher.exe`
5. **Stage 3** – Memory injection and process hollowing of ransomware into `autoup.exe`
6. **Payload Execution** – Encrypt files, kill processes, delete shadow copies, change wallpaper
7. **Post-Infection** – Show ransom note, redirect to onion domain via Tor

---

📦 SHADOW COPY CONCEPTS

What Are Shadow Copies?

- Windows feature to create **point-in-time backups** of files and volumes
- Enables **System Restore** and recovery of previous file versions
- Stored in a hidden system folder on the same drive

Why Do Attackers Delete Them?

- **To prevent recovery** without paying ransom
- Disables Volume Shadow Copy Service-based backup tools
- Ensures that encrypted files cannot be rolled back

Typical command used:

```
vssadmin delete shadows /all /quiet
```

In REvil's case, this was executed via obfuscated PowerShell:

```
Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
```

🧠 Thought Process for Each Question

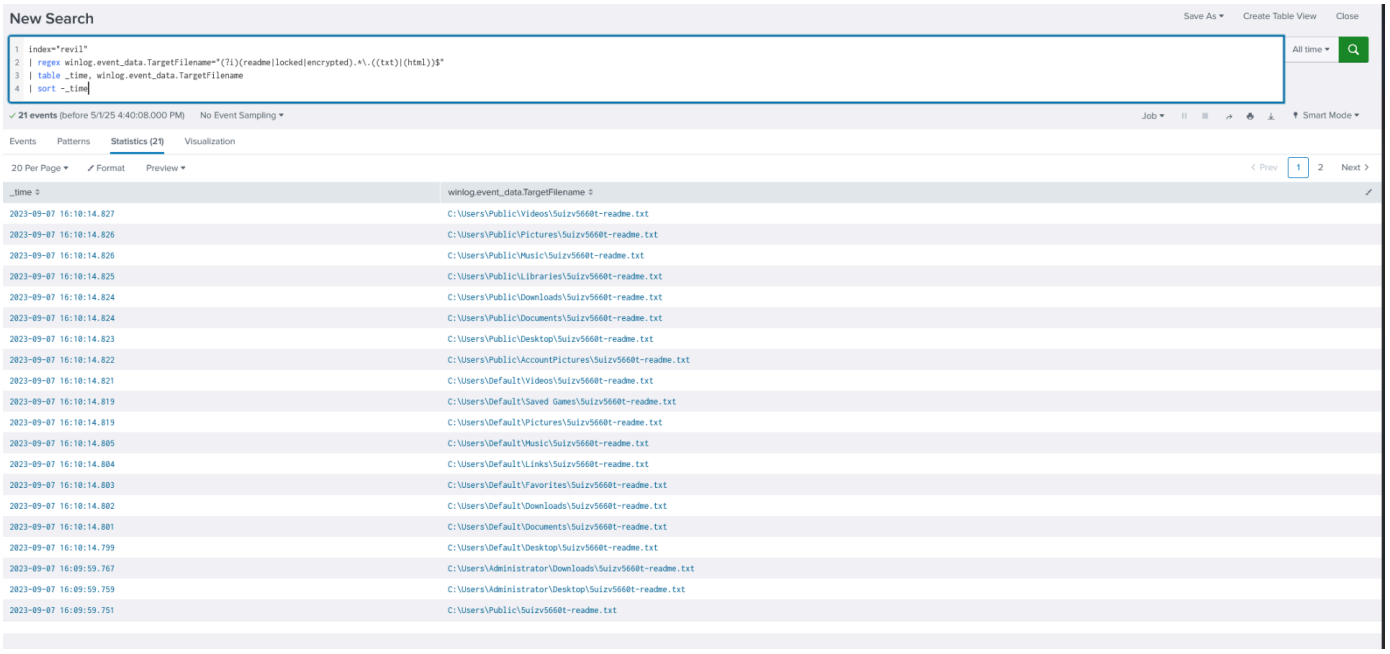
Q1: To begin your investigation, can you identify the filename of the note that the ransomware left behind?

I first used a generic regex filter to detect ransom notes:

```
index="revil"
| regex winlog.event_data.TargetFilename="(?i)(readme|locked|encrypted).*\.((txt)|(html))$"
| table _time, winlog.event_data.TargetFilename
| sort -_time
```

This produced:

```
2023-09-07 16:09:59.750 | C:\Users\Default\5uizv5660t-readme.txt
```



Q2: What's the process ID of the ransomware that's likely involved?

I tried using both `winlog.process.pid` and `winlog.event_data.ProcessId` — and learned the difference:

- `winlog.process.pid`: ID of the process **logging** the event.
- `winlog.event_data.ProcessId`: ID of the **actual** process that performed the action.

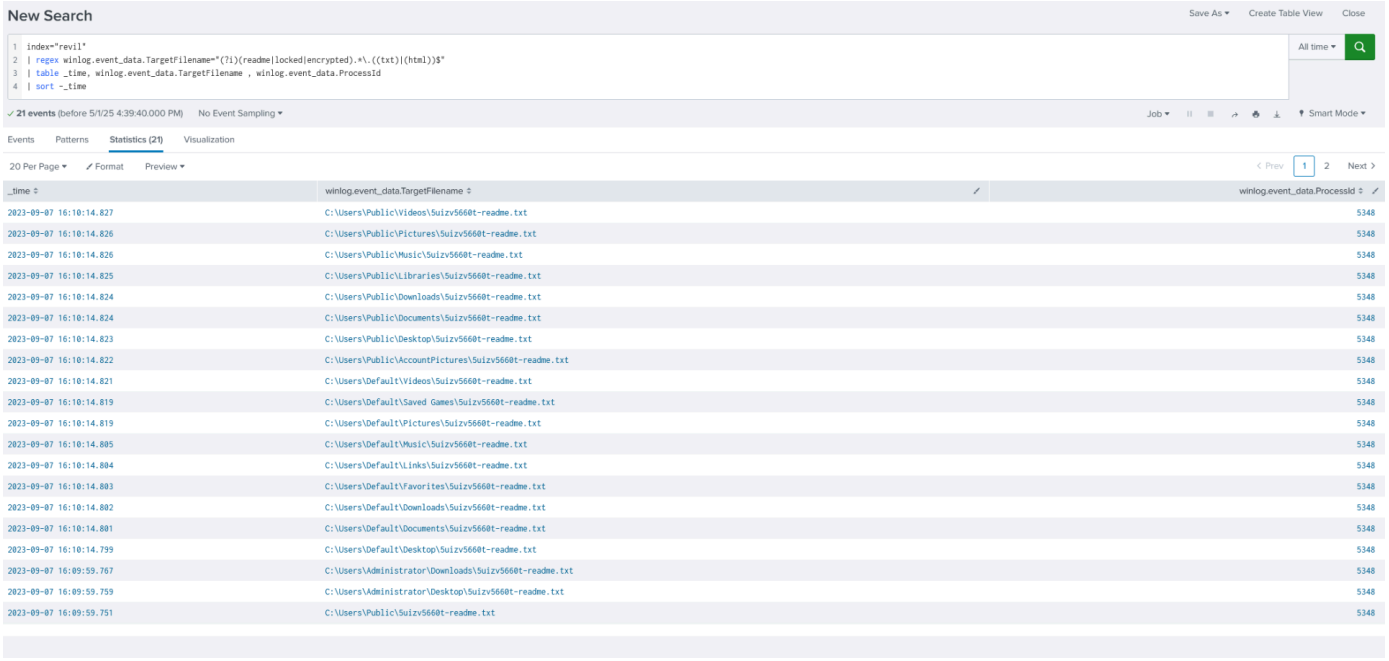
I used:

```
index="revil"
| regex winlog.event_data.TargetFilename="(?i)(readme|locked|encrypted).*\.((txt)|(html))$"
```

```
| table _time, winlog.event_data.TargetFilename ,
winlog.event_data.ProcessId, winlog.event_data.Image
| sort -_time
```

And found:

```
ProcessId: 5348
Image: facebook assistant.exe
```



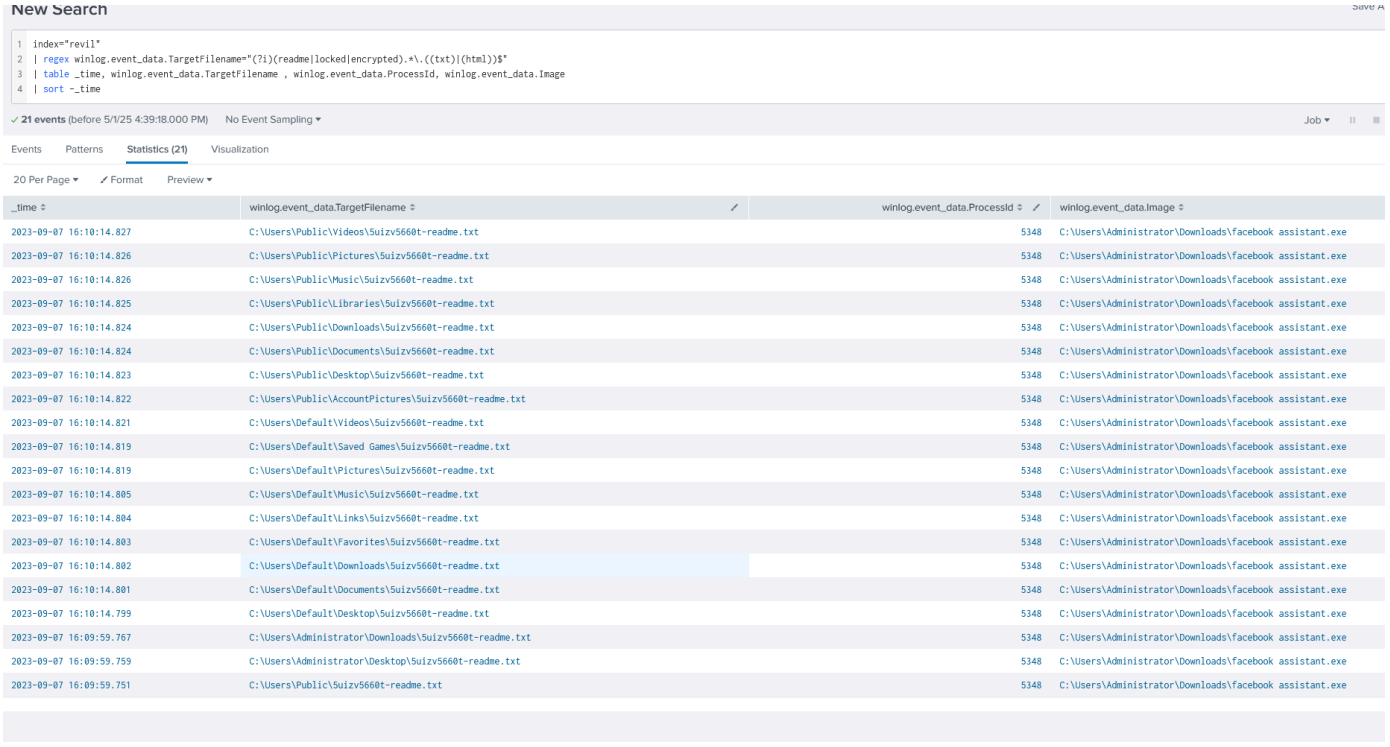
Q3: Where can we find the ransomware's executable file?

To find the binary location:

```
index="revil" winlog.event_data.ProcessId="5348"
```

I traced the process and confirmed its path as:

```
C:\Users\Administrator\Downloads\facebook assistant.exe
```



New Search

```
index="revil"
| regex winlog.event_data.TargetFilename="(?i)(readme|locked|encrypted).*\.((txt)|(html))$"
| table _time, winlog.event_data.TargetFilename , winlog.event_data.ProcessId, winlog.event_data.Image
| sort -_time
```

✓ 21 events (before 5/1/25 4:39:18.000 PM) No Event Sampling — Job

Events Patterns Statistics (21) Visualization

20 Per Page — Format — Preview

_time	winlog.event_data.TargetFilename	winlog.event_data.ProcessId	winlog.event_data.Image
2023-09-07 16:10:14.827	C:\Users\Public\Videos\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.826	C:\Users\Public\Pictures\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.826	C:\Users\Public\Music\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.825	C:\Users\Public\Libraries\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.824	C:\Users\Public\Downloads\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.824	C:\Users\Public\Documents\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.823	C:\Users\Public\Desktop\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.822	C:\Users\Public\AccountPictures\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.821	C:\Users\Default\Videos\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.819	C:\Users\Default\Saved Games\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.819	C:\Users\Default\Pictures\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.805	C:\Users\Default\Music\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.804	C:\Users\Default\Links\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.803	C:\Users\Default\Favorites\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.802	C:\Users\Default\Downloads\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.801	C:\Users\Default\Documents\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:10:14.799	C:\Users\Default\Desktop\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:09:59.767	C:\Users\Administrator\Downloads\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:09:59.759	C:\Users\Administrator\Desktop\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe
2023-09-07 16:09:59.751	C:\Users\Public\5uizv5660t-readme.txt	5348	C:\Users\Administrator\Downloads\facebook assistant.exe

---

Q4: What command was used to delete shadow copies?

From earlier analysis:

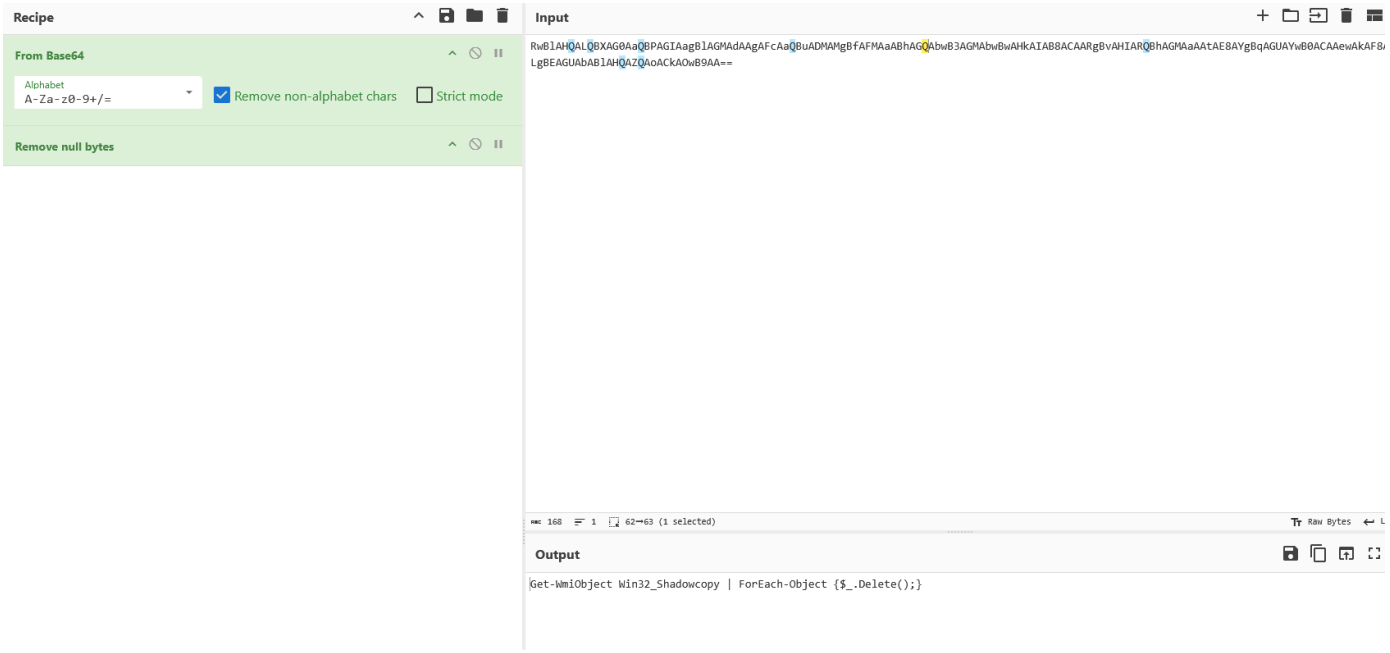
```
index="revil"
winlog.event_data.CurrentDirectory="C:\\Users\\Administrator\\Downloads\\"
| table winlog.event_data.CommandLine, winlog.event_data.Hashes
```

I found a suspicious PowerShell base64 string:

```
powershell -e RwBlAHQALQBXAG0AaQBPAGIA...
```

Decoded using CyberChef → revealed:

```
Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
```



This confirms REvil attempts to inhibit recovery by removing shadow copies.

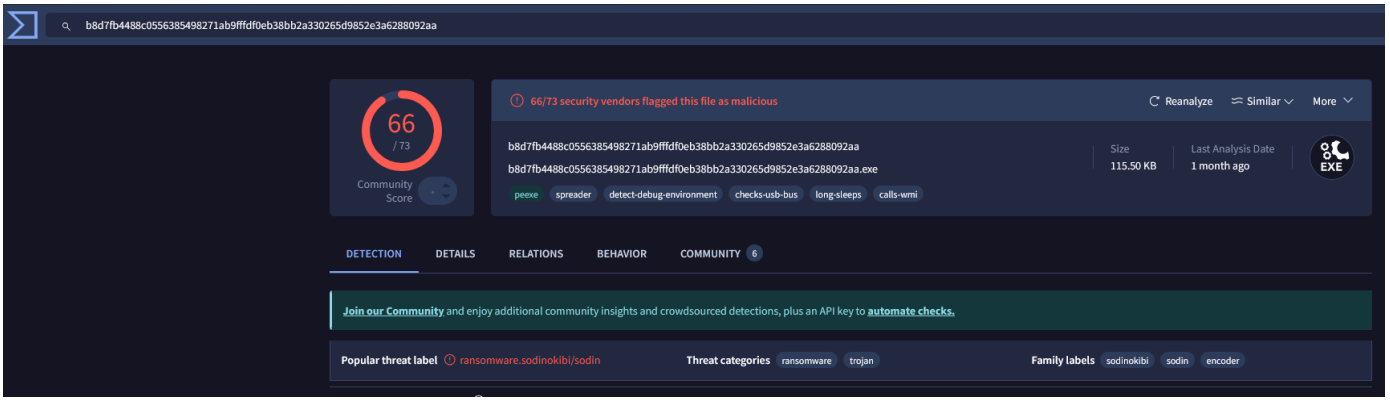
Q5: Provide the SHA256 hash of the ransomware's executable

To extract the hash:

```
index="revil" "facebook assistant.exe" event.code="1"
| table winlog.event_data.CommandLine, winlog.event_data.Hashes
```

SHA256 hash of `facebook assistant.exe`:

```
b8d7fb4488c0556385498271ab9fffdf0eb38bb2a330265d9852e3a6288092aa
```



Q6: What is the ransomware author's onion domain?

Used the SHA256 hash in Tria.ge sandbox → ransom note provided the onion URL:

```
http://aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion
```

✅ Correct format for answer field:

```
aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd.onion
```