Capstone Assignment

IS 4533

Malware Analysis

Jeremiah Poblete

ID: MNY215

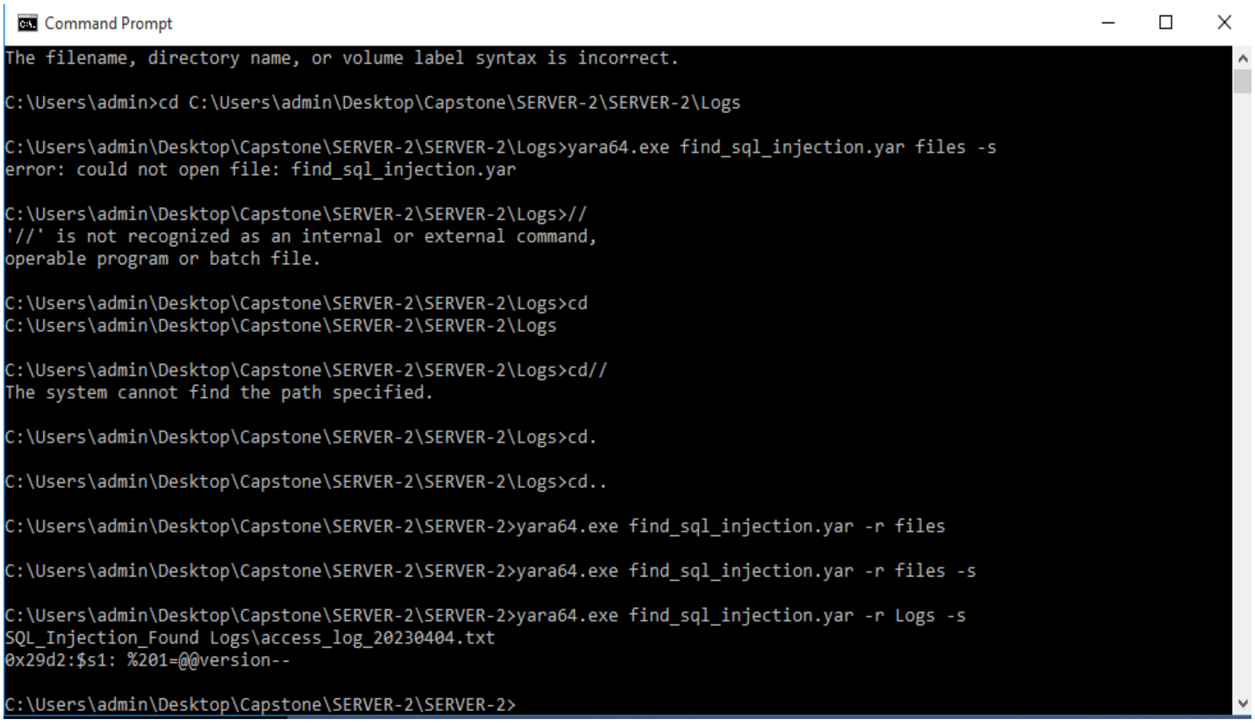
12/10/2024

This report investigates a cyberattack on HEB servers, where sensitive customer data was compromised through an SQL Injection attack. The analysis focuses on tracing the attacker's steps, recovering encrypted data and identifying critical artifacts. Skills used during the assignment, static analysis, data exfiltration, analyzing malware behaviors and reverse engineering. Tools used include:

- Yara
- Notepad++
- Bstrings
- UPX
- Cutter
- XOR
- Autopsy
- PEID

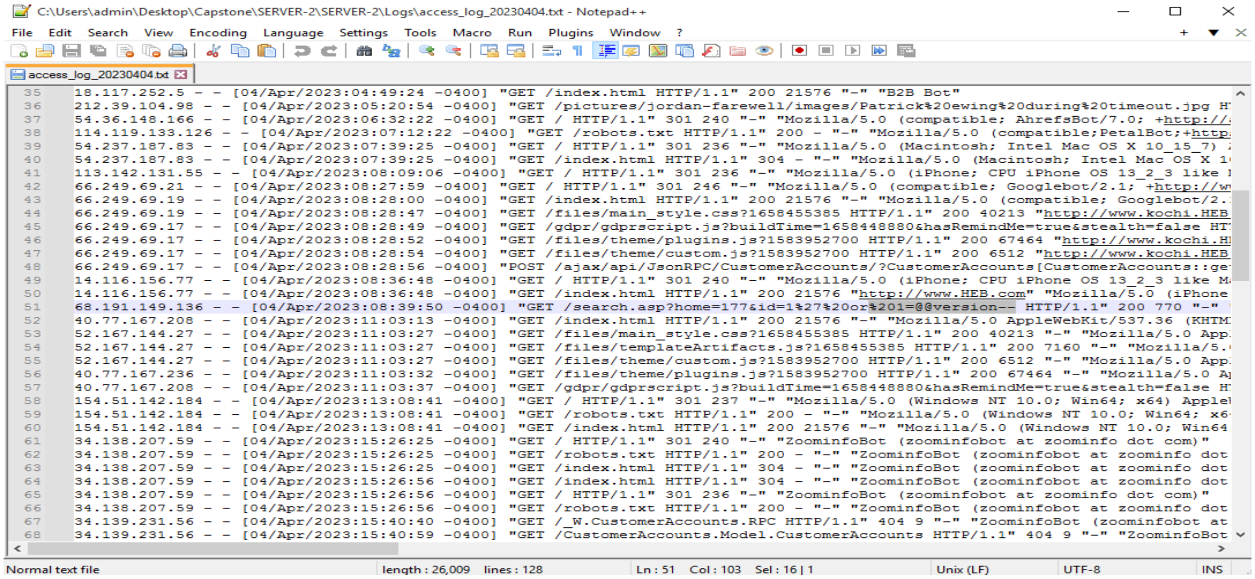
The assignment contained system files from Mr. Brown's computer, and files from 3 different servers that were compromised. Through a confession we learned that Server-2 was compromised with an SQL injection attack.

Used a Yara rule to determine which logs contained logs regarding the SQL Injection attack. The logs were contained within a text file labeled '*access_log_20230404.txt'*.



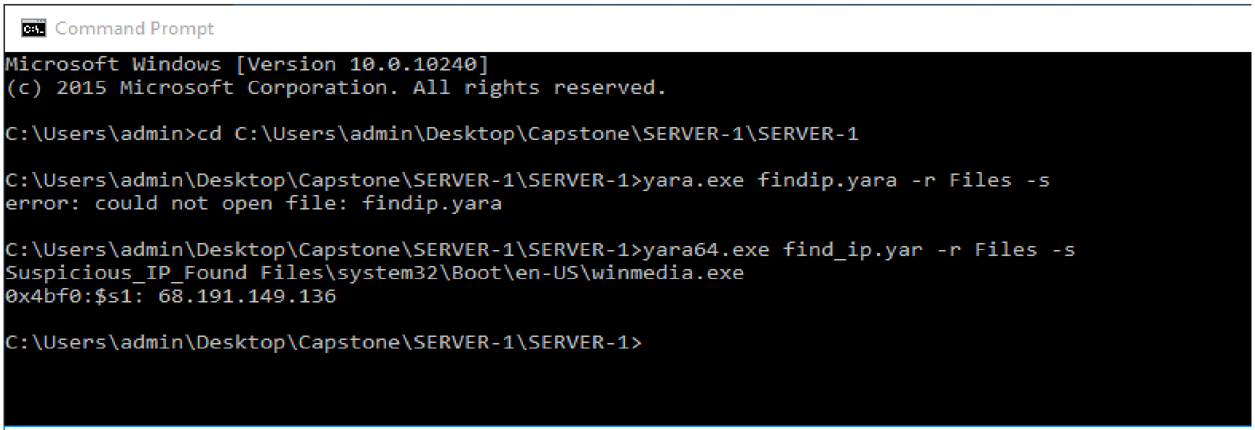
```
The filename, directory name, or volume label syntax is incorrect.

C:\Users\admin>cd C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>yara64.exe find_sql_injection.yar files -s
error: could not open file: find_sql_injection.yar

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>//
'//' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>cd
C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>cd//
The system cannot find the path specified.

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>cd.

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs>cd..

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2>yara64.exe find_sql_injection.yar -r files

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2>yara64.exe find_sql_injection.yar -r files -s

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2>yara64.exe find_sql_injection.yar -r Logs -s
SQL_Injection_Found Logs\access_log_20230404.txt
0x29d2:$s1: %201=@@version--

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2>
```

Opened Log file using Notepad ++ to find the IP address associated with the SQL injection. I found that the compromised IP address is ‘*68.191.149.136’*.

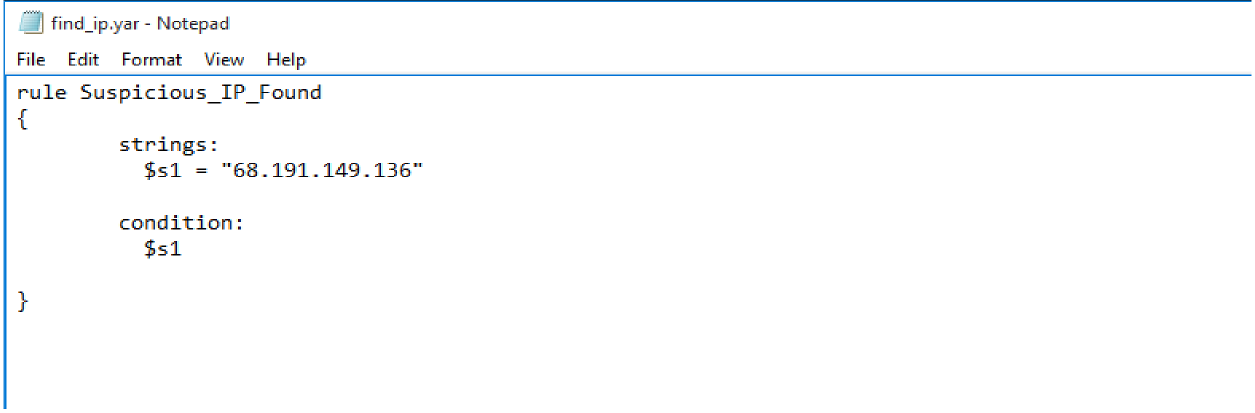

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2\Logs\access_log_20230404.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
access_log_20230404.txt
Normal text file
length : 26,009 lines : 128
Ln : 51 Col : 103 Sel : 16 | 1
Unix (LF)
UTF-8
INS

```
18.117.252.5 - - [04/Apr/2023:04:49:24 -0400] "GET /index.html HTTP/1.1" 200 21576 "-" "B2B Bot"
212.39.104.98 - - [04/Apr/2023:05:20:54 -0400] "GET /pictures/jordan-farewell/images/Patrick%20ewing%20during%20timeout.jpg H
54.36.148.166 - - [04/Apr/2023:06:32:22 -0400] "GET / HTTP/1.1" 301 240 "-" "Mozilla/5.0 (compatible; AhrefsBot/7.0; +http://
114.119.133.126 - - [04/Apr/2023:07:12:22 -0400] "GET /robots.txt HTTP/1.1" 200 - "-" "Mozilla/5.0 (compatible;PetalBot;+http
54.237.187.83 - - [04/Apr/2023:07:39:25 -0400] "GET / HTTP/1.1" 301 236 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
54.237.187.83 - - [04/Apr/2023:07:39:25 -0400] "GET /index.html HTTP/1.1" 304 - "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 1
113.142.131.55 - - [04/Apr/2023:08:09:06 -0400] "GET / HTTP/1.1" 301 236 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 13_2_3 like
66.249.69.21 - - [04/Apr/2023:08:27:59 -0400] "GET / HTTP/1.1" 301 246 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://w
66.249.69.19 - - [04/Apr/2023:08:28:00 -0400] "GET /index.html HTTP/1.1" 200 21576 "-" "Mozilla/5.0 (compatible; Googlebot/2.
66.249.69.19 - - [04/Apr/2023:08:28:47 -0400] "GET /files/main_style.css?1658455385 HTTP/1.1" 200 40213 "http://www.kochi.HEB
66.249.69.17 - - [04/Apr/2023:08:28:49 -0400] "GET /gdpr/gdprscript.js?buildTime=1658448880&hasRemindMe=true&stealth=false HT
66.249.69.17 - - [04/Apr/2023:08:28:52 -0400] "GET /files/theme/plugins.js?1583952700 HTTP/1.1" 200 67464 "http://www.kochi.H
66.249.69.17 - - [04/Apr/2023:08:28:54 -0400] "GET /files/theme/custom.js?1583952700 HTTP/1.1" 200 6512 "http://www.kochi.HEB
66.249.69.17 - - [04/Apr/2023:08:28:56 -0400] "POST /ajax/api/JsonRPC/CustomerAccounts/?CustomerAccounts[CustomerAccounts::ge
14.116.156.77 - - [04/Apr/2023:08:36:48 -0400] "GET / HTTP/1.1" 301 240 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 13_2_3 like M
14.116.156.77 - - [04/Apr/2023:08:36:48 -0400] "GET /index.html HTTP/1.1" 200 21576 "http://www.HEB.com" "Mozilla/5.0 (iPhone
68.191.149.136 - - [04/Apr/2023:08:39:50 -0400] "GET /search.asp?home=177&id=1%27%20or%201=@@version-- HTTP/1.1" 200 770 "-"
40.77.167.208 - - [04/Apr/2023:11:03:13 -0400] "GET /index.html HTTP/1.1" 200 21576 "-" "Mozilla/5.0 AppleWebKit/537.36 (KHTM
52.167.144.27 - - [04/Apr/2023:11:03:27 -0400] "GET /files/main_style.css?1658455385 HTTP/1.1" 200 40213 "-" "Mozilla/5.0 App
52.167.144.27 - - [04/Apr/2023:11:03:27 -0400] "GET /files/templateArtifacts.js?1658455385 HTTP/1.1" 200 7160 "-" "Mozilla/5.
52.167.144.27 - - [04/Apr/2023:11:03:27 -0400] "GET /files/theme/custom.js?1583952700 HTTP/1.1" 200 6512 "-" "Mozilla/5.0 App
40.77.167.236 - - [04/Apr/2023:11:03:32 -0400] "GET /files/theme/plugins.js?1583952700 HTTP/1.1" 200 67464 "-" "Mozilla/5.0 A
40.77.167.208 - - [04/Apr/2023:11:03:37 -0400] "GET /gdpr/gdprscript.js?buildTime=1658448880&hasRemindMe=true&stealth=false H
154.51.142.184 - - [04/Apr/2023:13:08:41 -0400] "GET / HTTP/1.1" 301 237 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) Apple
154.51.142.184 - - [04/Apr/2023:13:08:41 -0400] "GET /robots.txt HTTP/1.1" 200 - "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x6
154.51.142.184 - - [04/Apr/2023:13:08:41 -0400] "GET /index.html HTTP/1.1" 200 21576 "-" "Mozilla/5.0 (Windows NT 10.0; Win64
34.138.207.59 - - [04/Apr/2023:15:26:25 -0400] "GET / HTTP/1.1" 301 240 "-" "ZoominfoBot (zoominfobot at zoominfo dot com)"
34.138.207.59 - - [04/Apr/2023:15:26:25 -0400] "GET /robots.txt HTTP/1.1" 200 - "-" "ZoominfoBot (zoominfobot at zoominfo dot
34.138.207.59 - - [04/Apr/2023:15:26:25 -0400] "GET /index.html HTTP/1.1" 304 - "-" "ZoominfoBot (zoominfobot at zoominfo dot
34.138.207.59 - - [04/Apr/2023:15:26:56 -0400] "GET /index.html HTTP/1.1" 304 - "-" "ZoominfoBot (zoominfobot at zoominfo dot
34.138.207.59 - - [04/Apr/2023:15:26:56 -0400] "GET / HTTP/1.1" 301 236 "-" "ZoominfoBot (zoominfobot at zoominfo dot com)"
34.138.207.59 - - [04/Apr/2023:15:26:56 -0400] "GET /robots.txt HTTP/1.1" 200 - "-" "ZoominfoBot (zoominfobot at zoominfo dot
34.139.231.56 - - [04/Apr/2023:15:40:40 -0400] "GET /_W.CustomerAccounts.RPC HTTP/1.1" 404 9 "-" "ZoominfoBot (zoominfobot at
34.139.231.56 - - [04/Apr/2023:15:40:59 -0400] "GET /CustomerAccounts.Model.CustomerAccounts HTTP/1.1" 404 9 "-" "ZoominfoBot
```

I then created a Yara rule to find the IP address within the system Files of Server-1 where the URL is embedded. The compromised executable is ‘*winmedia.exe’*.

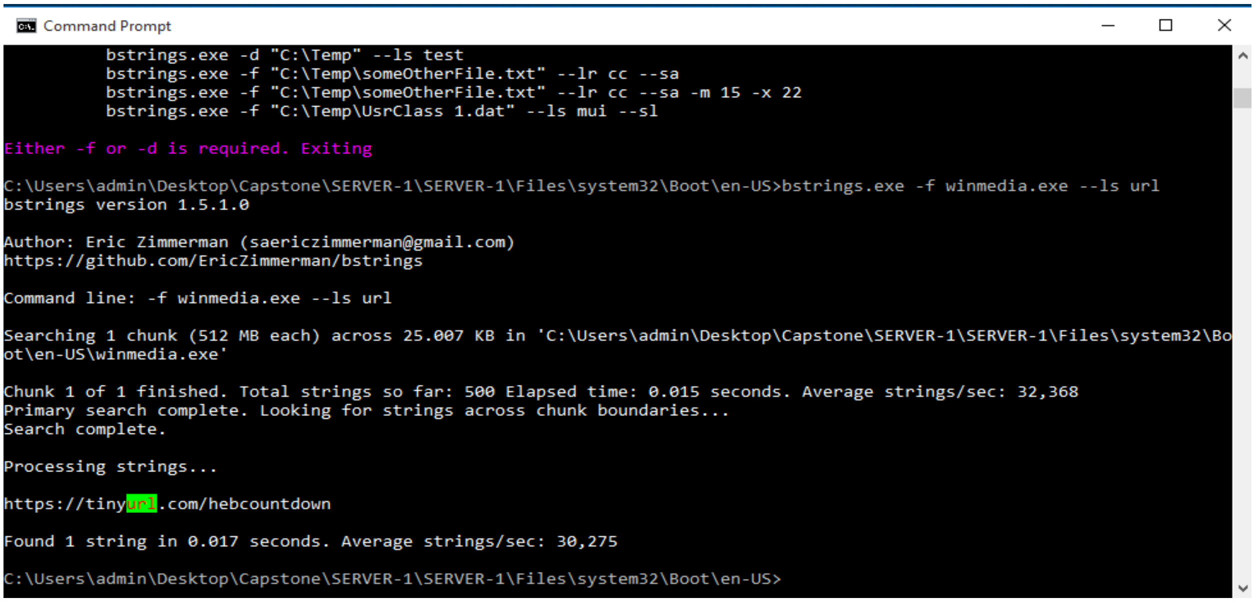


```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1>yara.exe findip.yara -r Files -s
error: could not open file: findip.yara

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1>yara64.exe find_ip.yar -r Files -s
Suspicious_IP_Found Files\system32\Boot\en-US\winmedia.exe
0x4bf0:$s1: 68.191.149.136

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1>
```



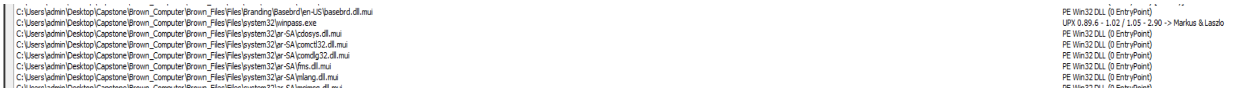
```
rule Suspicious_IP_Found
{
        strings:
          $s1 = "68.191.149.136"

        condition:
          $s1

}
```

Used Bstrings to find the “countdown” URL ‘*https://tinyurl.com/hebcountdown’* found within Server-1.



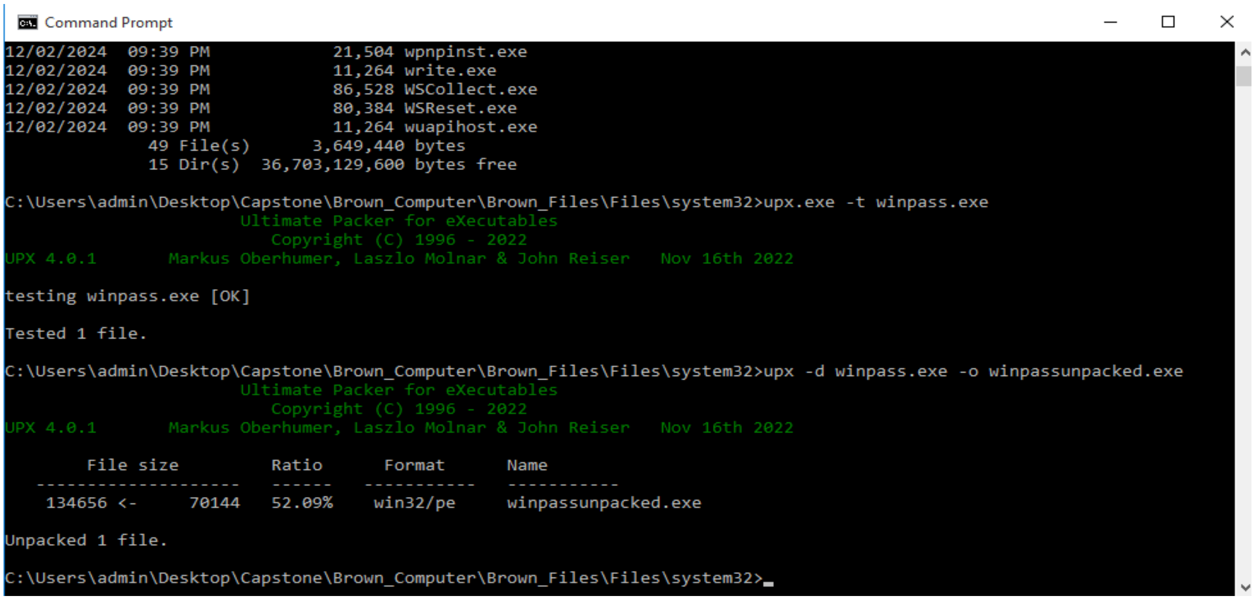
```
         bstrings.exe -d "C:\Temp" --ls test
         bstrings.exe -f "C:\Temp\someOtherFile.txt" --lr cc --sa
         bstrings.exe -f "C:\Temp\someOtherFile.txt" --lr cc --sa -m 15 -x 22
         bstrings.exe -f "C:\Temp\UsrClass 1.dat" --ls mui --sl

Either -f or -d is required. Exiting

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1\Files\system32\Boot\en-US>bstrings.exe -f winmedia.exe --ls url
bstrings version 1.5.1.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/bstrings

Command line: -f winmedia.exe --ls url

Searching 1 chunk (512 MB each) across 25.007 KB in 'C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1\Files\system32\Bo
ot\en-US\winmedia.exe'

Chunk 1 of 1 finished. Total strings so far: 500 Elapsed time: 0.015 seconds. Average strings/sec: 32,368
Primary search complete. Looking for strings across chunk boundaries...
Search complete.

Processing strings...

https://tinyurl.com/hebcountdown

Found 1 string in 0.017 seconds. Average strings/sec: 30,275

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1\Files\system32\Boot\en-US>
```

On Brown’s computer was a UPX-packed executable that contains the password for the kill-switch but requires a PIN to access. I used PEID to scan for UPX packed files in Mr. Brown’s system to begin reverse engineering the password. The packed file was contained in ‘*winpass.exe’*.



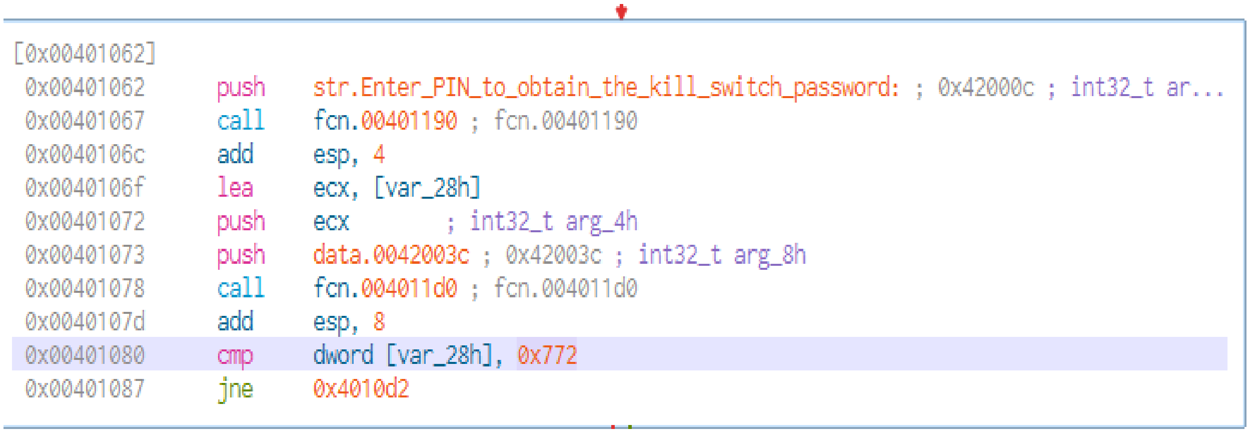
```
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\Branding\Basebrd\en-US\basebrd.dll.mui   PE Win32 DLL (0 EntryPoint)
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\winpass.exe                     UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\ar-SA\cdosys.dll.mui            PE Win32 DLL (0 EntryPoint)
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\ar-SA\comctl32.dll.mui          PE Win32 DLL (0 EntryPoint)
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\ar-SA\comdlg32.dll.mui          PE Win32 DLL (0 EntryPoint)
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\ar-SA\fms.dll.mui               PE Win32 DLL (0 EntryPoint)
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32\ar-SA\mlang.dll.mui             PE Win32 DLL (0 EntryPoint)
```

I then used UPX to unpack the executable to allow for further investigation.



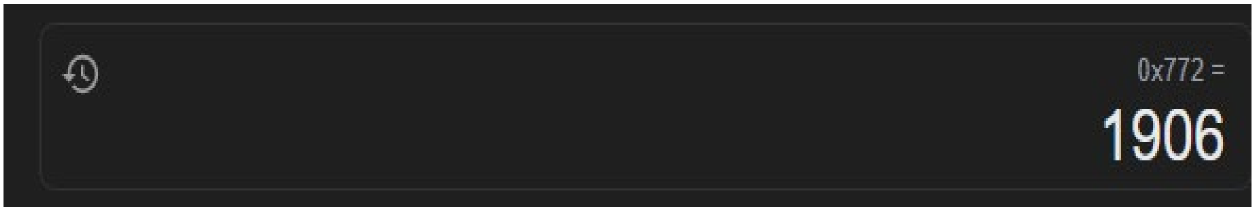
```
12/02/2024  09:39 PM            21,504 wpnpinst.exe
12/02/2024  09:39 PM            11,264 write.exe
12/02/2024  09:39 PM            86,528 WSCollect.exe
12/02/2024  09:39 PM            80,384 WSReset.exe
12/02/2024  09:39 PM            11,264 wuapihost.exe
              49 File(s)      3,649,440 bytes
              15 Dir(s)  36,703,129,600 bytes free

C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32>upx.exe -t winpass.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2022
UPX 4.0.1       Markus Oberhumer, Laszlo Molnar & John Reiser   Nov 16th 2022

testing winpass.exe [OK]

Tested 1 file.

C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32>upx -d winpass.exe -o winpassunpacked.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2022
UPX 4.0.1       Markus Oberhumer, Laszlo Molnar & John Reiser   Nov 16th 2022

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    134656 <-     70144   52.09%    win32/pe     winpassunpacked.exe

Unpacked 1 file.

C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32>
```

Using Cutter I reverse engineered the executable to determine the PIN to obtain the kill-switch password. I found the decision structure that compares user input to the PIN, allowing me to see the Hex Value of the PIN ‘*0x772*’.



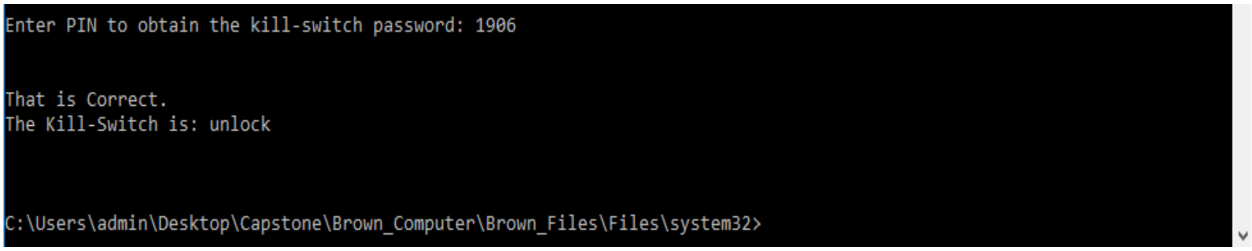
```
[0x00401062]
 0x00401062      push    str.Enter_PIN_to_obtain_the_kill_switch_password: ; 0x42000c ; int32_t ar...
 0x00401067      call    fcn.00401190 ; fcn.00401190
 0x0040106c      add     esp, 4
 0x0040106f      lea     ecx, [var_28h]
 0x00401072      push    ecx          ; int32_t arg_4h
 0x00401073      push    data.0042003c ; 0x42003c ; int32_t arg_8h
 0x00401078      call    fcn.004011d0 ; fcn.004011d0
 0x0040107d      add     esp, 8
 0x00401080      cmp     dword [var_28h], 0x772
 0x00401087      jne     0x4010d2
```

I used a calculator to determine the decimal value of the PIN ‘*1906*’.



```
0x772 =
1906
```

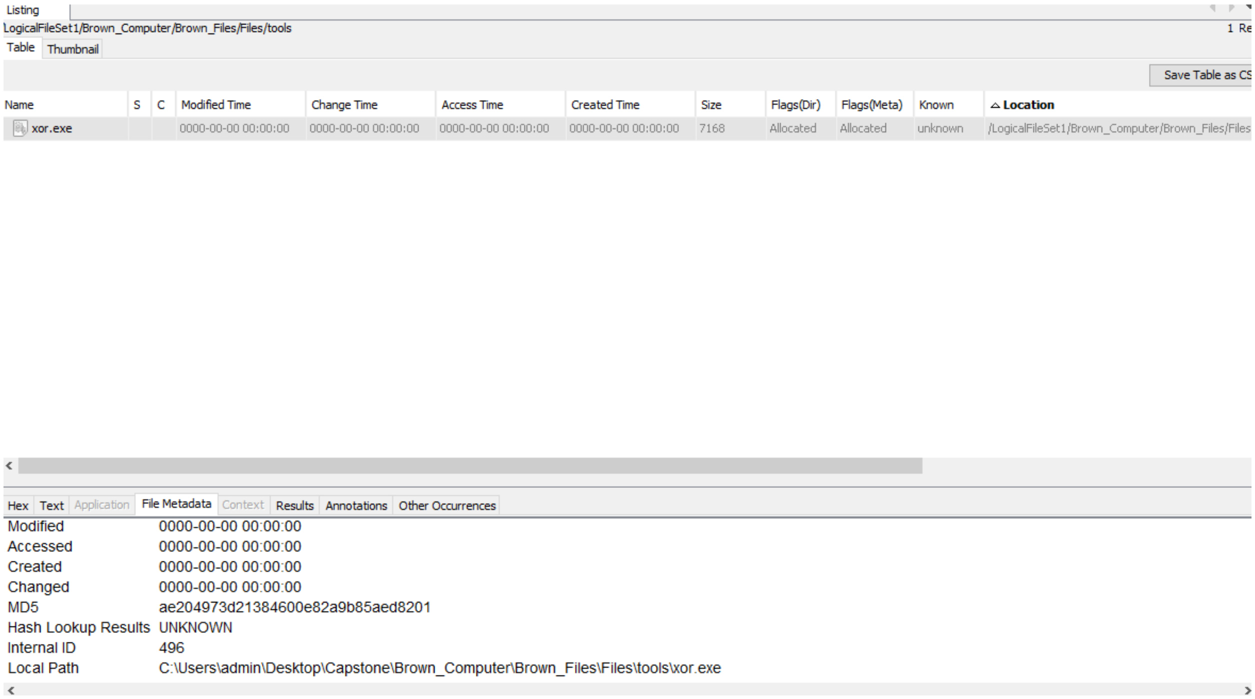
Ran the executable and used the PIN that was obtained. This provided the Kill-Switch password: ‘*unlock*’.



```
Enter PIN to obtain the kill-switch password: 1906


That is Correct.
The Kill-Switch is: unlock



C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32>
```

The next step is to retrieve the customer data that was encrypted using XOR. Mr. Brown stated that he stored the tools and data both reside in the same directory as his system and stated the key is ‘*2023*’. This means the hash value will be the same across both systems.

I began by determining the hash value of the XOR file within Mr. Brown's system. Using Autopsy I found that the MD5 hash is '*ae204973d21384600e82a9b85aed8201*'.

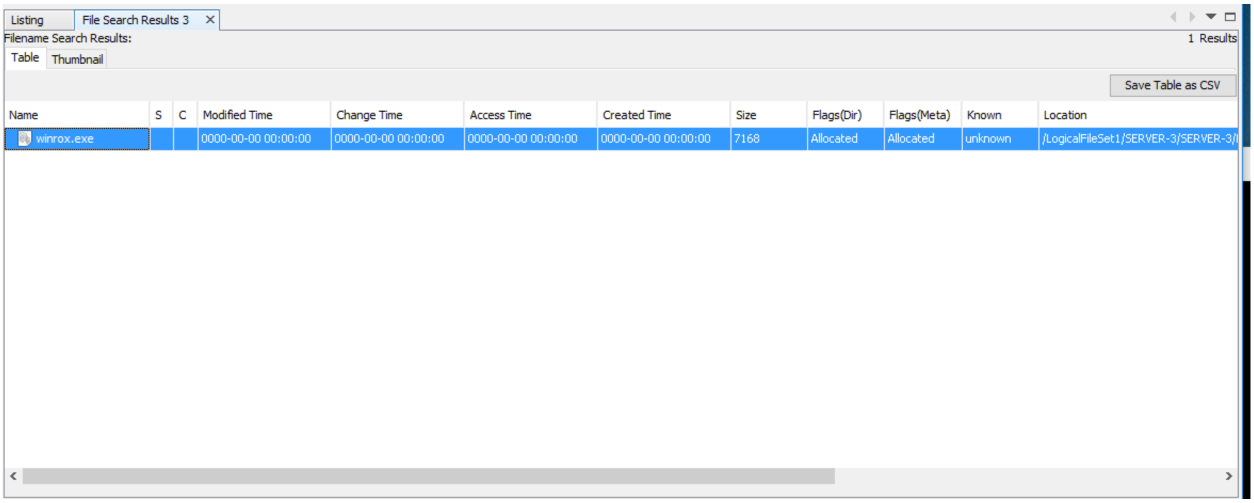


LogicalFileSet1/Brown_Computer/Brown_Files/Files/tools 1 Re

Name	S	C	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
xor.exe			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	7168	Allocated	Allocated	unknown	/LogicalFileSet1/Brown_Computer/Brown_Files/Files

Modified	0000-00-00 00:00:00
Accessed	0000-00-00 00:00:00
Created	0000-00-00 00:00:00
Changed	0000-00-00 00:00:00
MD5	ae204973d21384600e82a9b85aed8201
Hash Lookup Results	UNKNOWN
Internal ID	496
Local Path	C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\tools\xor.exe

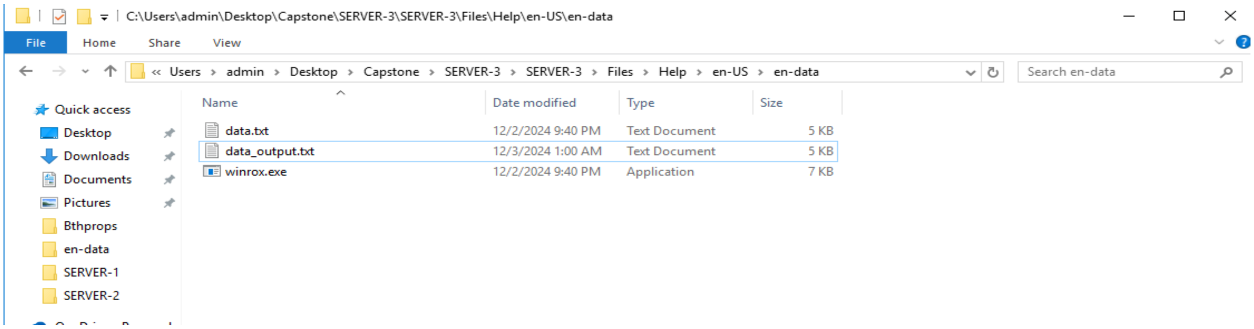
I then used Autopsy to search for an executable in Server-3 containing the same MD5 hash. The executable was hidden under '*winrox.exe*'.



Filename Search Results: 1 Results

Name	S	C	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
winrox.exe			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	7168	Allocated	Allocated	unknown	/LogicalFileSet1/SERVER-3/SERVER-3/

The directory containing the encrypted XOR file is under ‘*Server-3\Files\Help\en-US\en-data*’.

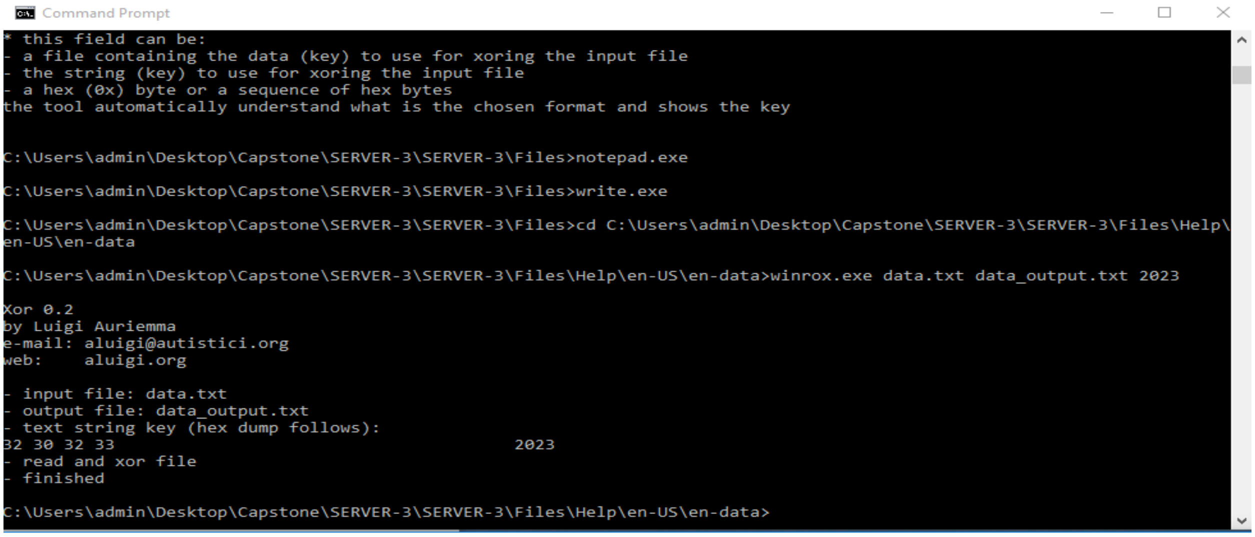


C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files\Help\en-US\en-data

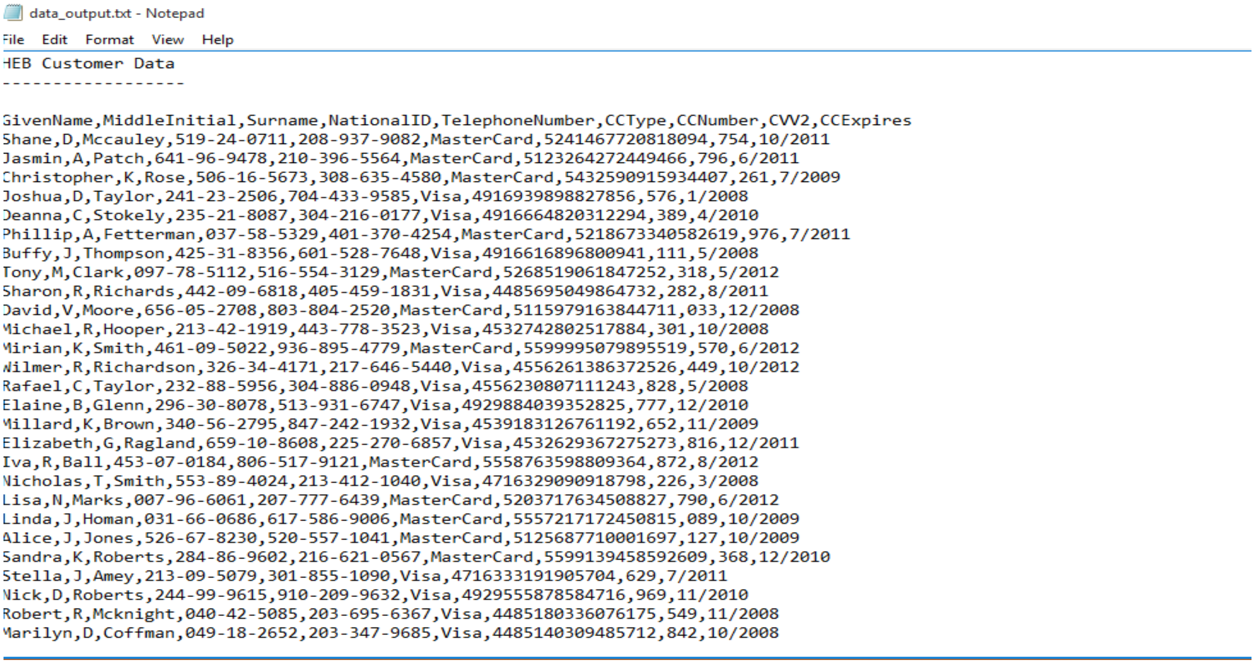
« Users › admin › Desktop › Capstone › SERVER-3 › SERVER-3 › Files › Help › en-US › en-data

Name	Date modified	Type	Size
data.txt	12/2/2024 9:40 PM	Text Document	5 KB
data_output.txt	12/3/2024 1:00 AM	Text Document	5 KB
winrox.exe	12/2/2024 9:40 PM	Application	7 KB

Used XOR to decrypt the file using the key provided by Mr. Brown. The screenshot below shows the decrypted text file containing customer data.

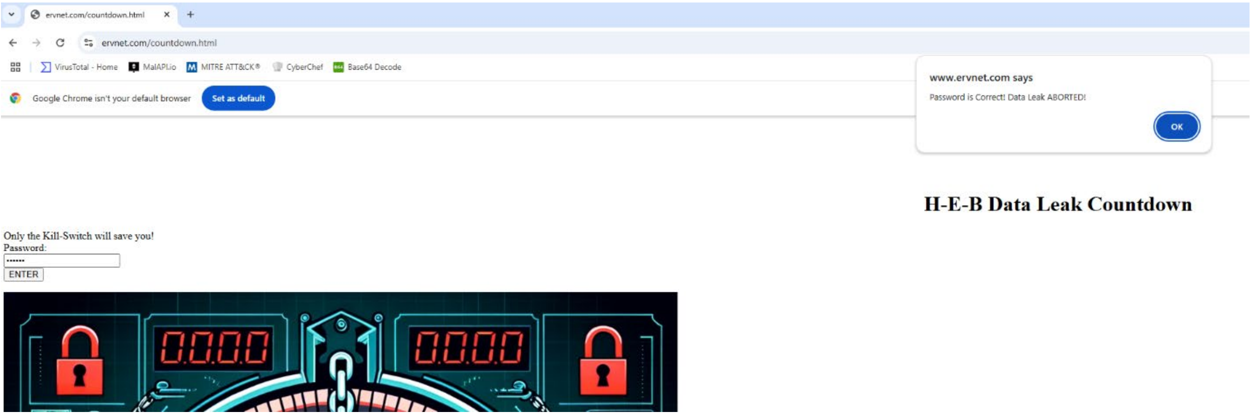


```
* this field can be:
- a file containing the data (key) to use for xoring the input file
- the string (key) to use for xoring the input file
- a hex (0x) byte or a sequence of hex bytes
the tool automatically understand what is the chosen format and shows the key


C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files>notepad.exe

C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files>write.exe

C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files>cd C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files\Help\
en-US\en-data

C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files\Help\en-US\en-data>winrox.exe data.txt data_output.txt 2023

Xor 0.2
by Luigi Auriemma
e-mail: aluigi@autistici.org
web:    aluigi.org

- input file: data.txt
- output file: data_output.txt
- text string key (hex dump follows):
32 30 32 33                                        2023
- read and xor file
- finished

C:\Users\admin\Desktop\Capstone\SERVER-3\SERVER-3\Files\Help\en-US\en-data>
```



```
HEB Customer Data
------------------

GivenName,MiddleInitial,Surname,NationalID,TelephoneNumber,CCType,CCNumber,CVV2,CCExpires
Shane,D,Mccauley,519-24-0711,208-937-9082,MasterCard,5241467720818094,754,10/2011
Jasmin,A,Patch,641-96-9478,210-396-5564,MasterCard,5123264272449466,796,6/2011
Christopher,K,Rose,506-16-5673,308-635-4580,MasterCard,5432590915934407,261,7/2009
Joshua,D,Taylor,241-23-2506,704-433-9585,Visa,4916939898827856,576,1/2008
Deanna,C,Stokely,235-21-8087,304-216-0177,Visa,4916664820312294,389,4/2010
Phillip,A,Fetterman,037-58-5329,401-370-4254,MasterCard,5218673340582619,976,7/2011
Buffy,J,Thompson,425-31-8356,601-528-7648,Visa,4916616896800941,111,5/2008
Tony,M,Clark,097-78-5112,516-554-3129,MasterCard,5268519061847252,318,5/2012
Sharon,R,Richards,442-09-6818,405-459-1831,Visa,4485695049864732,282,8/2011
David,V,Moore,656-05-2708,803-804-2520,MasterCard,5115979163844711,033,12/2008
Michael,R,Hooper,213-42-1919,443-778-3523,Visa,4532742802517884,301,10/2008
Mirian,K,Smith,461-09-5022,936-895-4779,MasterCard,5599995079895519,570,6/2012
Wilmer,R,Richardson,326-34-4171,217-646-5440,Visa,4556261386372526,449,10/2012
Rafael,C,Taylor,232-88-5956,304-886-0948,Visa,4556230807111243,828,5/2008
Elaine,B,Glenn,296-30-8078,513-931-6747,Visa,4929884039352825,777,12/2010
Millard,K,Brown,340-56-2795,847-242-1932,Visa,4539183126761192,652,11/2009
Elizabeth,G,Ragland,659-10-8608,225-270-6857,Visa,4532629367275273,816,12/2011
Iva,R,Ball,453-07-0184,806-517-9121,MasterCard,5558763598809364,872,8/2012
Nicholas,T,Smith,553-89-4024,213-412-1040,Visa,4716329090918798,226,3/2008
Lisa,N,Marks,007-96-6061,207-777-6439,MasterCard,5203717634508827,790,6/2012
Linda,J,Homan,031-66-0686,617-586-9006,MasterCard,5557217172450815,089,10/2009
Alice,J,Jones,526-67-8230,520-557-1041,MasterCard,5125687710001697,127,10/2009
Sandra,K,Roberts,284-86-9602,216-621-0567,MasterCard,5599139458592609,368,12/2010
Stella,J,Amey,213-09-5079,301-855-1090,Visa,4716333191905704,629,7/2011
Nick,D,Roberts,244-99-9615,910-209-9632,Visa,4929555878584716,969,11/2010
Robert,R,Mcknight,040-42-5085,203-695-6367,Visa,4485180336076175,549,11/2008
Marilyn,D,Coffman,049-18-2652,203-347-9685,Visa,4485140309485712,842,10/2008
```

Navigated to the URL to find the Kill-Switch that will stop the customer data from being released and entered the password that was obtained.



www.ervnet.com says

Password is Correct! Data Leak ABORTED!

H-E-B Data Leak Countdown

Only the Kill-Switch will save you!
Password:
••••••
ENTER

The investigation of the HEB server breach reveals a sophisticated attack. This was achieved through SQL Injection and lateral movement within the network. Brown was able to access sensitive data and encrypted the files to hold for Ransomware. In the future it is important to improve security posture by monitoring lateral movement and deploying more advanced IDS tools.