Case: Compromised HEB Servers
Capstone Project

Student Name: Raymond Ng
IS 4533 – 001
Spring 2023
April 19, 2023

Introduction:
The purpose of this lab report is to document the investigation of the HEB Servers hacking incident, which resulted in the exposure of customer data and a monetary demand from the perpetrator, Benjamin R. Brown of Dallas, TX. This report summarizes the findings of the investigation, including Brown's statements during the interviews, the method used to compromise the HEB Servers, and the measures taken to prevent the automatic release of customer data.

Case Description:
The investigation began with the execution of a search warrant at Brown's residence in Dallas, where he confessed to compromising HEB SERVER-2 using an SQL injection attack in early April 2023. Brown indicated that he moved laterally across the network to SERVER-3, where he found customer data, then used an XOR tool to encrypt data before exfiltrating. A search of Brown's computer revealed a UPX-packed executable containing the kill-switch password but requiring a four-digit-PIN to access.

Brown confessed to creating a "countdown" website for HEB executives to pay his monetary demands. The website URL is embedded in malware on SERVER-1, and Brown's IP address is also embedded in the same malware. The XOR key used to encrypt the data is `2023`, and the same XOR tool is also on Brown's computer in a 'tools' directory. However, Brown claimed to have forgotten the location of the XOR tool and the encrypted data on Server-3. Brown also stated that once the money is paid, there is a "kill-switch" password to stop the automatic release of customer data.

Methodology:

In, **SERVER-2**, I used the given YARA rule, `find_sql_injection.yar`, via command terminal I discovered `access_log_20230404.txt` contains the SQL Injection attack **`%201=@@version--`** (*Figure 1*).

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

C:\Users\admin\Desktop\Capstone\SERVER-2\SERVER-2>type find_sql_injection.yar
rule SQL_Injection_Found
{
        strings:
          $s1 = "%201=@@version--"

        condition:
          $s1

}

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 Logs
SQL_Injection_Found Logs\access_log_20230404.txt
```

Figure 1: Applying YARA rule to locate log with a SQL injection attack

Next, I opened `access_log_20230404.txt` with Notepad++, searching for **`%201=@@version--`**, I discovered IP **`68.191.149.136`** is associated with the injection attack (*Figure 2*).

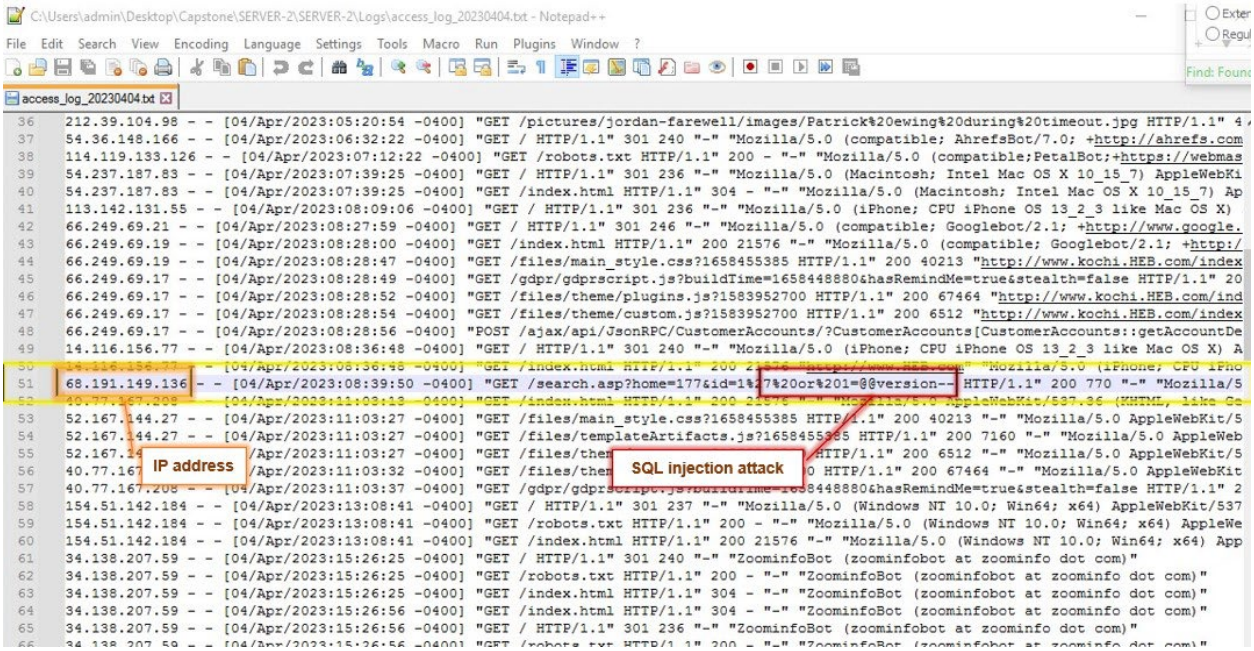

Figure 2: Opened log file via Notepad++ to search for injection attack.

Knowing the IP, I created a YARA rule and executed a search in the **SERVER-1** folder to find the malware via command terminal and discovered that `winmedia.exe` was the executable file, the malware (*Figure 3*).

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

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1>type find_ip.yar
rule SQL_Injection_Found
{
        strings:
          $s1 = "68.191.149.136"

        condition:
          $s1

}

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1>yara64.exe find_ip.yar -r Files
SQL_Injection_Found Files\system32\Boot\en-US\winmedia.exe

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

Figure 3: Created a YARA rule to find the malware.

To look for the “countdown” URL found in the malware, I changed directory via command terminal to where the malware was located. Next, I used the `bstrings` utility

and executed `bstrings -f winmedia.exe --lr url3986` to find the URL associated with the malware, `https://tinyurl.com/hebcountdown` (*Figure 4*).

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

C:\Users\admin\Desktop\Capstone\SERVER-1\SERVER-1\Files\system32\Boot\en-US>bstrings -f winmedia.exe --lr url3986
bstrings version 1.5.1.0

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

Command line: -f winmedia.exe --lr url3986

Searching via RegEx pattern: ^
                [a-z][a-z0-9+\-.]*://                       # Scheme
                ([a-z0-9\-._~%!$&'()*+,;=]+@)?              # User
                (?<host>[a-z0-9\-._~%]+                     # Named host
                |\[[a-f0-9:.]+\]                            # IPv6 host
                |\[v[a-f0-9][a-z0-9\-._~%!$&'()*+,;=:]+\])  # IPvFuture host
                (:[0-9]+)?                                  # Port
                (/[a-z0-9\-._~%!$&'()*+,;=:@]+)*/?          # Path
                (\?[a-z0-9\-._~%!$&'()*+,;=:@/?]*)?         # Query
                (\#[a-z0-9\-._~%!$&'()*+,;=:@/?]*)?         # Fragment
                $

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

Chunk 1 of 1 finished. Total strings so far: 500 Elapsed time: 0.074 seconds. Average strings/sec: 6,724
Primary search complete. Looking for strings across chunk boundaries...
Search complete.

Processing strings...

https://tinyurl.com/hebcountdown

Found 1 string in 0.079 seconds. Average strings/sec: 6,362
```

Figure 4: Using bstrings to search for URL associated with malware.

Ran a recursive scan via PEiD and I discovered the UPX-packed executable file on Brown's computer, discovered `winpass.exe` (*Figure 5*).

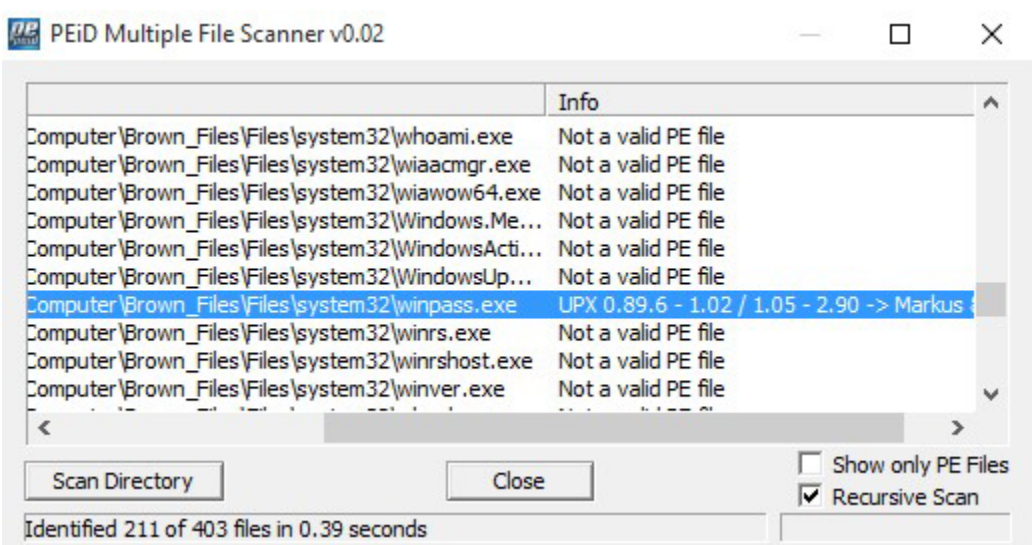

Figure 5: Executing recursive scan to look for UPX-packed file.

To confirm/validate the UPX-packed file, I executed `upx -t winpass.exe` in the command terminal (*Figure 6*).

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

Figure 6: Checking to see if executable file is packed with UPX via command terminal.

To unpack `winpass.exe`, I execute `upx -d winpass.exe` via command terminal (*Figure 7*).

```
C:\Users\admin\Desktop\Capstone\Brown_Computer\Brown_Files\Files\system32>upx -d winpass.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     winpass.exe

Unpacked 1 file.
```

Figure 7: Unpacking executing file

To confirm/validate `winpass.exe` was successfully unpacked, I executed `upx -t winpass.exe` in the command terminal (*Figure 8*).

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

upx: winpass.exe: NotPackedException: not packed by UPX

Tested 0 files.
```

Figure 8: Checking to see if executable file is unpacked via command terminal.

Next, I ran `winpass.exe` via Command Terminal, which prompted for a PIN (*Figure 9*).

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

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

Figure 9:Executed the executable file via command terminal.

Using Cutter, I disassembled `winpass.exe` and determined the PIN was `1906` (*Figure 10*).

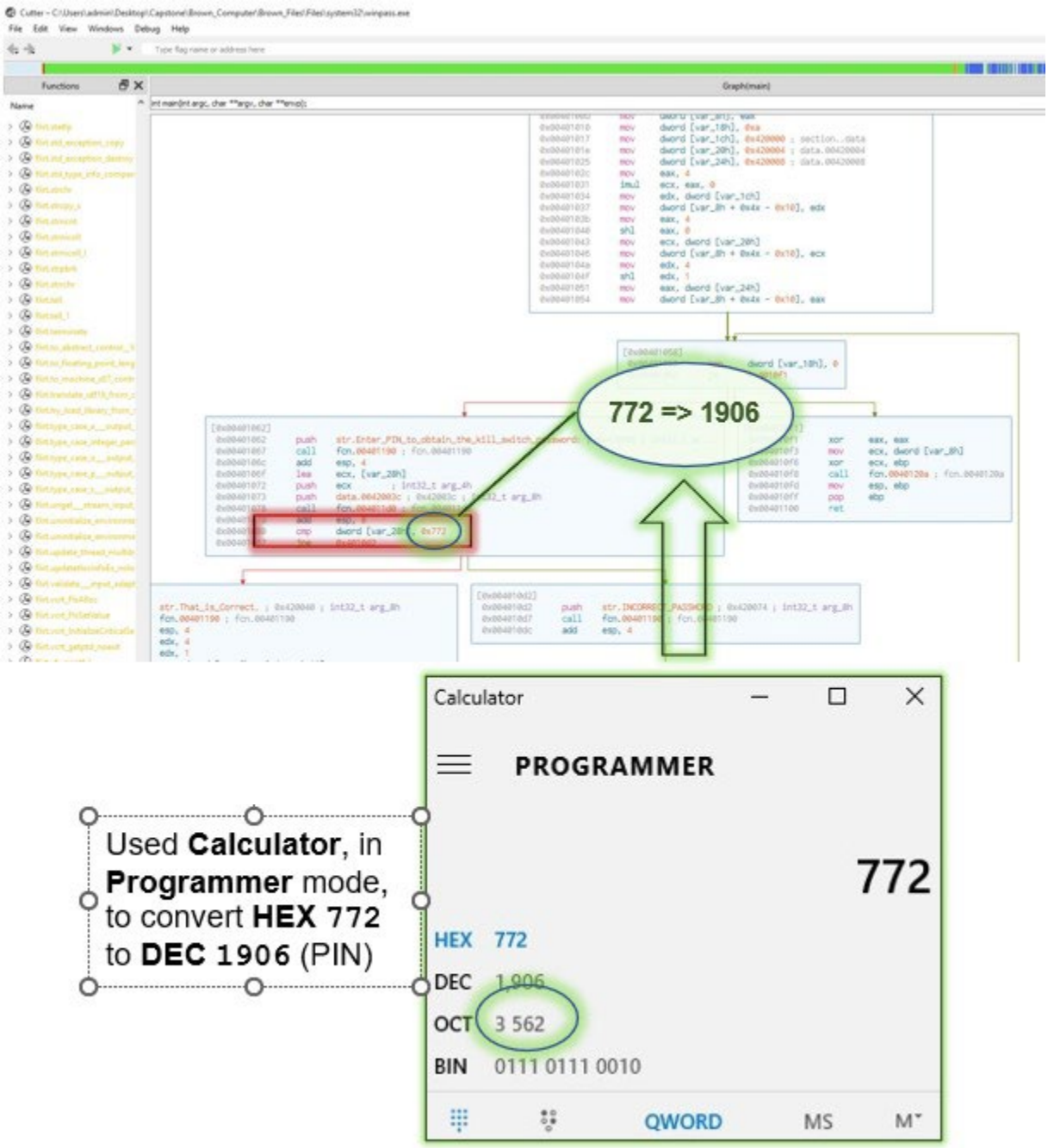

Figure 10: Using Cutter, I disassembled executable file to discover the PIN, PIN was converted via Calculator application via PC.

Next, ran `winpass.exe` in the command terminal again and entered the PIN which revealed the Kill-Switch password, `unlock` (*Figure 11*).

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

Enter PIN to obtain the kill-switch password: 1906


That is Correct.
The Kill-Switch is: unlock
```

Figure 11: Executed file in command terminal which revelaed the kill-switch password.

Next, to look for where Mr. Brown stored the modified XOR tool and the encrypted customer data, I used *HashMyFiles* to open the directory on Brown's computer where the unmodified XOR tool could be found to look for the MD5 hash associated with `xor.exe` (*Figure 12*).

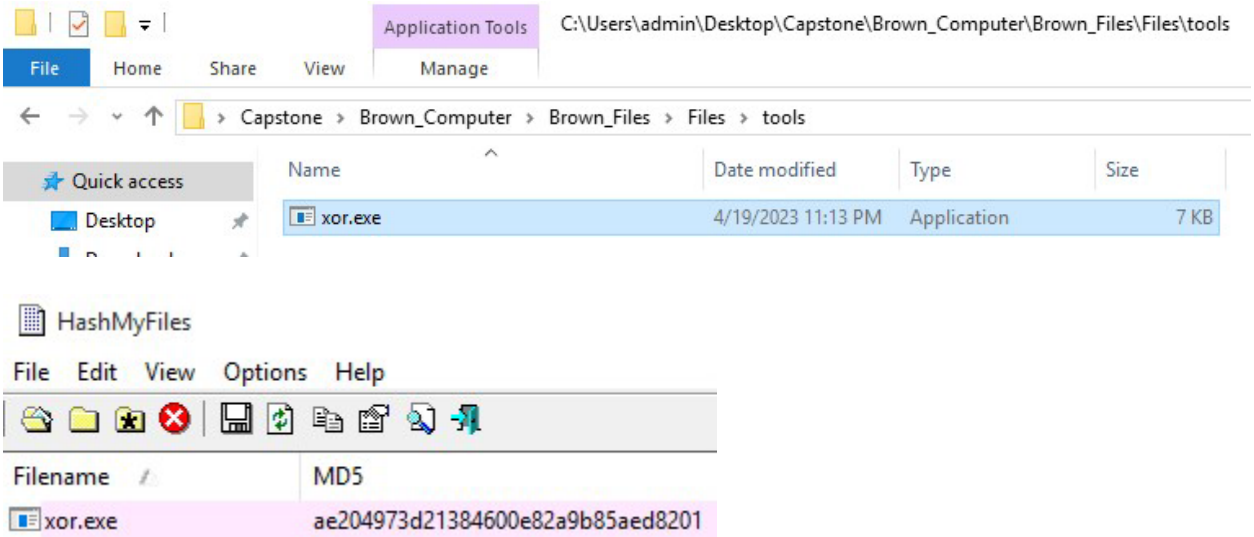

Figure 12: Using HashMyFiles to determine MD5 hash associated with xor.exe.

Now that I knew the MD5 hash I needed to look for, `ae204973d21384600e82a9b845aed8201`, I opened up the **Files** directory in **SERVER-3** to search for the same hash (*Figure 13*).

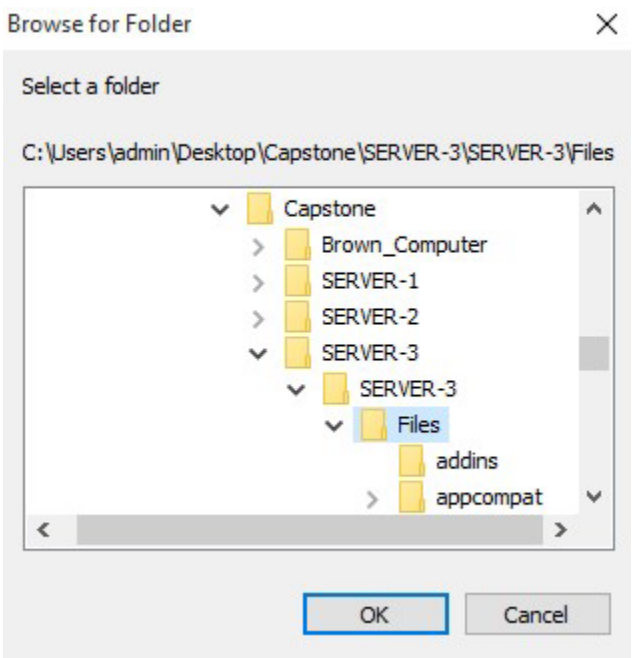

Figure 13: Searching directory for executable file with same MD5 hash via HashMyFile.

One the results loaded, I analyzed the output and used the Find feature in *HashMyFiles* to locate another executable file with the same MD5 hash, `ae204973d21384600e82a9b845aed8201` (*Figure 14*). Then, I went to the directory where the executable, `winrox.exe`, could be found (*Figure 15)*.

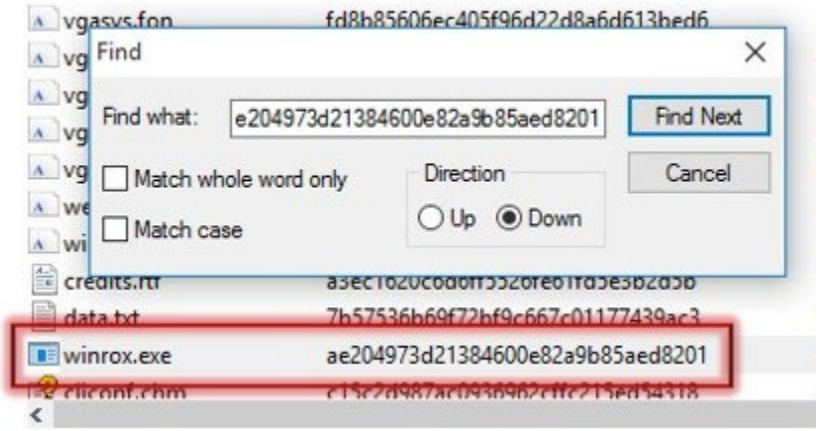

Figure 14: Locating similar executable file with same MD5 hash via HashMyFile.

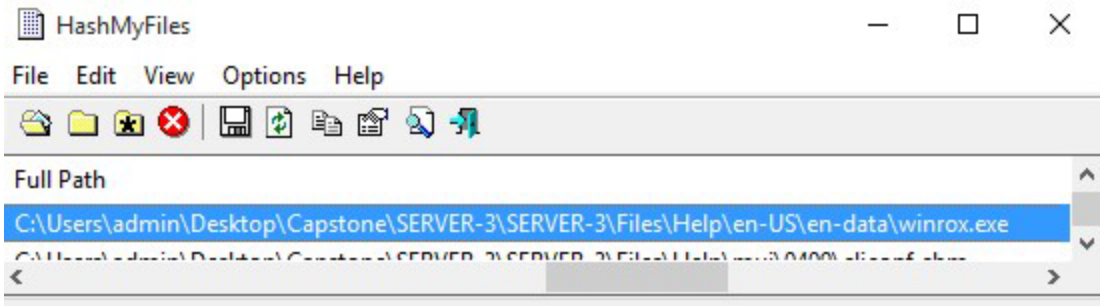

Figure 15: Looking for the directory where the modified executable is located via HashMyFiles

In the directory I noticed two files, `winrox.exe` and `data.txt`. From the previous steps I know that `winrox.exe` is the modified xor.exe tool and when I opened up the `data.txt`, the contents were encrypted. In the command terminal with the PIN given to us by Mr. Brown (`2023`), I used the modified XOR tool (`winrox.exe`) to decrypt `data.txt`, by `executing winrox data.txt data_decrypted.txt 2023` (*Figure 16*).

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

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

- input file: data.txt
- output file: data_decrypted.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>
```

Figure 16: Using XOR utility to decrypt data.txt.

A new file created, `data_decrypted.txt`, in the same directory. I opened the file and observed the unencrypted customer data (*Figure 17*).

```
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
Stephen,M,Parker,460-17-5293,512-707-8963,MasterCard,5149939217893692,155,9/2010
Justin,L,York,414-17-0327,731-772-1560,MasterCard,5312750925897284,425,8/2008
Erin,L,Dickey,536-30-4518,509-581-9490,Visa,4539364185514744,805,6/2011
Kathleen,W,Carpenter,557-01-8922,949-742-7684,MasterCard,5287947933656428,831,11/2011
Vera,T,Gray,593-25-9842,352-282-6307,Visa,4485517374451747,943,11/2010
Anita,D,Fitzsimmons,533-23-4910,253-458-6499,Visa,4532629133133806,131,3/2008
Daniel,D,Haldeman,021-50-8793,413-248-3772,MasterCard,5433990114736828,470,6/2010
Sherry,F,Fierro,494-11-4182,660-214-9271,Visa,4716899002780114,552,1/2008
```

Figure 17:Decrypted file with customer information.

Finally, I opened the URL, `https://tinyurl.com/hebcountdown`, in a browser and entered the kill-switch password, `unlock`, on the “countdown” website (*Figure 18*).

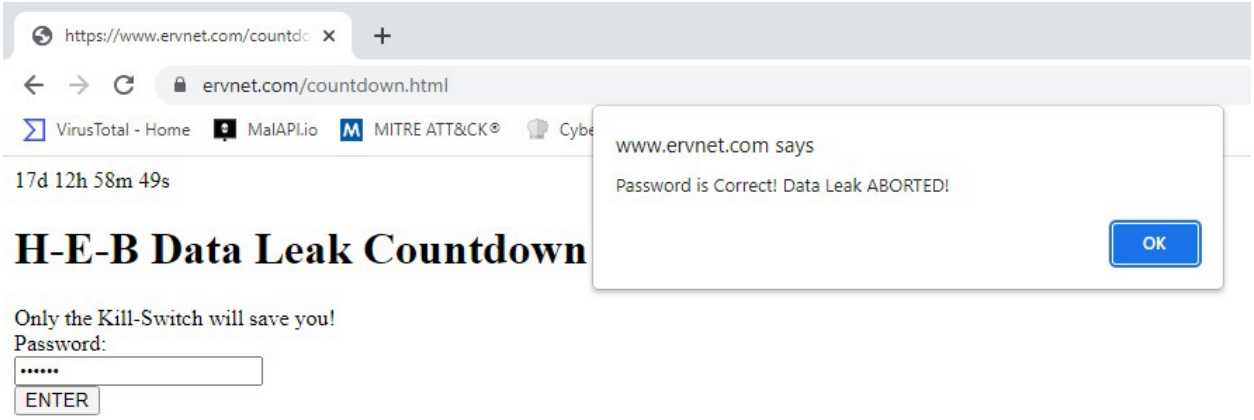

Figure 18: Opened URL found earlier and inputted kill-switch password.

Conclusion:
The investigation has revealed critical information that will enable the prevention of the automatic release of customer data. The malware embedded on SERVER-1 was identified and removed, and the logs from the SQL injection were examined to locate Brown's IP address. The data exfiltrated from SERVER-3 was discovered and the XOR key `2023`, provided by Mr. Brown, was successfully decrypted the data. The UPX-packed executable on Brown's computer containing the kill-switch password required a four-digit-PIN, which was discovered. Further, the contents of his computer were thoroughly analyzed to locate the executable and recover the PIN. The investigation underscores the need for robust cybersecurity measures to protect against SQL injection attacks and the importance of proactive measures to prevent data breaches.