**When Pentest Tools Go Brutal: Red-Teaming Tool Being Abused by Malicious Actors**

Brute Ratel C4 made its initial debut as a penetration testing tool in December 2020. At the time, its development was a part-time effort by a security engineer named Chetan Nayak (aka Paranoid Ninja) living in India. According to his website (Dark Vortex), Nayak amassed several years of experience working in senior red team roles across western cybersecurity vendors. Over the past 2.5 years, Nayak introduced incremental improvements to the pentest tool in terms of features, capabilities, support and training.

In January 2022, Nayak left his day job in order to pursue full-time development and training workshops. That same month, he released Brute Ratel v0.9.0 (Checkmate), which is described as the “biggest release for Brute Ratel till date.”

Checkmate release notes: "Brute Ratel v0.9.0 (Checkmate) is biggest release for Brute Ratel till date. This release brings major changes to the Brute Ratel's loader, reflective DLL, shellcode and the internal APIs being called. As detailed in the provided version, where several syscall injection techniques were added for evasion, but they were limited to the reflective DLLs loader of BRc4 and the VEH API of Windows. This version uses an updated version of Syscalls for almost everything except a few of those which I was pretty sure would never be hooked since they are too nosy. This release was built after reverse engineering several top tier EDR and Antivirus DLLs. A quick summary of the changes can be found in the release notes.

However, of greater concern, the release description also specifically noted that “this release was built after reverse engineering several top tier EDR and Antivirus DLLs.”

Our analysis highlights the ongoing and relevant debate within the cybersecurity industry surrounding the ethics relating to the development and use of penetration testing tools that can be exploited for offensive purposes.

BRc4 currently advertises itself as “A Customized Command and Control Center for Red Team and Adversary Simulation.” On May 16, Nayak announced that the tool had gained 480 users across 350 customers.

Twitter post by Paranoid Ninja (Brute Ratel C4): Extremely happy to announce that Brute Ratel has crossed 350 customers and around 480+ licenses last week. #BRc4 started off as a fun project 2 years back, before I decided to make it commercial. I had my doubts when I started, since BRc4 was very basic back then and had a few bugs. But a lot of my followers seemed to have faith in me and the product. I decided to make the whole support transparent on discord so that customers can discuss on the channel and report anything they find. And with a lot of feedback and support from the community, BRc4 became better and better with every release and has more features and is stable than any other C2 in the market. There are some seriously insane features in the roadmap in the next few months and I couldn't be happier with the decision I took 2 years back.

Figure 2. BRC4 customer announcement. Source: https://twitter.com/NinjaParanoid/status/1526110403356282880

The latest version, Brute Ratel v1.0 (Sicilian Defense) was released a day later on May 17, and is currently offered for sale at a price of $2,500 per user and $2,250 per renewal. With this price point and customer base, BRc4 is positioned to take in more than $1 million dollars in sales over the next year.

In terms of features, BRc4 advertises the following capabilities:

SMB and TCP payloads provide functionality to write custom external C2 channels over legitimate websites such as Slack, Discord, Microsoft Teams and more.

Built-in debugger To detect EDR userland hooks.

Ability to keep memory artifacts hidden from EDRs and AV.

Direct Windows SYS calls on the fly.

Egress over HTTP, HTTPS, DNS Over HTTPS, SMB and TCP.

LDAP Sentinel provides a rich GUI interface to query various LDAP queries to the domain or a forest.

Multiple command and control channels – multiple pivot options such as SMB, TCP, WMI, WinRM and managing remote services over RPC.

Take screenshots.

x64 shellcode loader.

Reflective and object file loader.

Decoding KRB5 ticket and converting it to hashcat.

Patching Event Tracing for Windows (ETW).

Patching Anti Malware Scan Interface (AMSI).

Create Windows system services.

Upload and download files.

Create files via CreateFileTransacted.

Port scan.

From Click to Brute

Screenshot of VirusTotal verdicts for Roshan\_CV sample: "No security vendors and no sandboxes flagged this file as malicious."

Figure 4. VirusTotal verdicts for sample as of June 27, 2022.

The file in VirusTotal named Roshan\_CV.iso (SHA256: 1FC7B0E1054D54CE8F1DE0CC95976081C7A85C7926C03172A3DDAA672690042C) appears to be a curriculum vitae (similar to a resume) of an individual named Roshan. It was uploaded to VirusTotal on May 19, 2022, from Sri Lanka. The ISO file extension refers to an optical disc image file, derived from the International Organization for Standardization’s ISO 9960 file system, which is typically used to back up files for CD/DVD. The ISO file is not malicious and requires a user to double-click, which mounts the ISO as a Windows drive. Finally, the archived files of the ISO are displayed to the user. In this case, when the ISO is double-clicked, a user is presented with the following:

As depicted in Figure 5, the user would see a file named Roshan-Bandara\_CV\_Dialog, which has a fake icon image of Microsoft Word, purporting to be an individual's CV, and written in Microsoft Word. From the window dialog box it can be ascertained that the ISO was assembled on May 17, 2022, which coincidentally is the same day the new BRc4 was released.

If the user were to double-click on the file, it would then install Brute Ratel C4 on the user's machine.

By default, on Windows operating systems, hidden files are not displayed to the user. In Figure 6, there are four hidden files concealed from view. If the display of hidden files is enabled, the user sees the following:

With "show hidden files" enabled, more files are visible: Roshan-Bandara-CV-Dialog, version.dll, vresion.dll, OneDrive.Update, OneDriveUpdater.exe

Figure 6. Viewing ISO image with “show hidden files” enabled.

The lure file, the one visible to the user, is a Windows shortcut file (LNK) with the following properties:

Microsoft shortcut files, those with a .lnk file extension, contain enriched metadata that can be used to provide artifacts about the file. Some key artifacts of this file are:

Link CLSID: 20D04FE0-3AEA-1069-A2D8-08002B30309D

The CLSID used here isn’t the normal CLSID for LNK files, as this CLSID is associated with the Control Panel (always Icons view).

Command line arguments: %windir%/system32/cmd.exe /c start OneDriveUpdater.exe

Icon location: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE

When Roshan-Bandara\_CV\_Dialog is double-clicked, the following actions occur:

cmd.exe is launched with the parameters of:

/c start OneDriveUpdater.exe. The /c parameter instructs cmd.exe to launch OneDriveUpdater.exe via Windows start command from the current working directory and exit.

OneDriveUpdater.exe is a digitally signed binary by Microsoft that is used to synchronize data from a local machine to the cloud. It is not malicious and is being abused to load the actor’s DLL. Once OneDriveUpdater.exe is executed, the following actions occur:

Since Version.dll is a dependency DLL of OneDriveUpdater.exe and exists in the same directory as OneDriveUpdater.exe, it will be loaded.

Version.dll has been modified by the actors to load an encrypted payload file, OneDrive.update. The modification decrypts the file and in-memory loads the first stage of shellcode. To maintain code capabilities, the actors use DLL API proxying to forward requests to the legitimate version.dll named vresion.dll. Vresion.dll is a dependency file of the actor’s version.dll and will be loaded with the actor’s version.dll.

The in-memory code, that is Brute Ratel C4, executes as a Windows thread in the RuntimeBroker.exe process space and begins to communicate with IP 174.129.157[.]251 on TCP port 443.

Packaging of Roshan\_CV.ISO

The composition of the ISO file, Roshan\_CV.ISO, closely resembles that of other nation-state APT tradecraft. The following table shows a side-by-side comparison of Roshan\_CV.ISO and that of a previously identified APT29 sample (Decret.ISO).

Comparison between Roshan file and Decret file (known APT29 sample). It shows that both abuse trusted applications digitally signed, contain hidden files, share a file extension, drop a shortcut (LNK) file as a lure, use a fake icon and use Version.dll. Roshan uses an encrypted payload where Decret does not. The image also contains SHA-256 hashes for both files.

The following images show how Roshan\_CV.ISO and Decret.ISO would look to a user when double-clicked. Figure 9 is a screenshot of the default Windows File Explorer; “show hidden files” is not checked. In both images, the user is presented with a shortcut file (LNK file) that starts the malicious activity when double-clicked.

Figure 10 shows how the ISOs would appear when show hidden files” is enabled for viewing.

The flow of execution is the following:

The delivery of packaged ISO files is typically sent via spear phishing email campaigns or downloaded to the victim by a second-stage downloader.

While we lack insight into how this particular payload was delivered to a target environment, we observed connection attempts to the C2 server originating from three Sri Lankan IP addresses between May 19-20.

Modification of Version.dll

Version.dll is a modified version of a legitimate Microsoft file written in C++. The implanted code is used to load and decrypt an encrypted payload file. The decrypted payload is that of shellcode (x64 assembly) that is further used to execute Brute Ratel C4 on the host.

In order for Version.dll to maintain its code capabilities for OneDriveUpdater.exe, the actors include the legitimate digitally signed Microsoft version.dll and named it vresion.dll. Any time OneDriveUpdater.exe makes a call into the actor’s Version.dll, the call is proxied to vresion.dll. Because of this, the actor’s version.dll will load vresion.dll as a dependency file.

The implanted code begins when the DLL is loaded via DLL\_PROCESS\_ATTACH and performs the following at the DLLMain subroutine:

Enumerate all processes and locate the process ID (PID) for Runtimebroker.exe.

Read the payload file OneDrive.Update from the current working directory.

Call the Windows API ntdll ZwOpenProcess using the process ID from step 1. The process is opened with full control access.

Decrypt the payload file using the XOR encryption algorithm with a 28-byte key of: jikoewarfkmzsdlhfnuiwaejrpaw

Call the Windows API NtCreateSection, which creates a block of memory that can be shared between processes. This API is used to share memory with Runtimebroker.exe and Version.dll.

Two calls into the Windows API NtMapViewOfSection. The first call maps the contents of the decrypted payload into the current process memory space, and the second call maps the contents into the Runtimebroker.exe memory space.

Calls the Windows API NtDelayExecution and sleeps (pauses execution) for ~4.27 seconds.

Call the Windows API NtCreateThreadEx. This API starts a new thread with the start address of the memory copied to Runtimebroker.exe.

Calls the Windows API NtDelayExecution and sleeps (pauses execution) for ~4.27 seconds.

Finished.

The technique outlined above uses process injection via undocumented Windows NTAPI calls. The decrypted payload is now running within the runtimebroker.exe memory space. The following is a snippet of code from version.dll that starts the execution of the in-memory decrypted payload.

Version.dll calling NtCreateThreadEx.

X64 Shellcode – Decrypted OneDrive.Update

The decrypted payload file is x64 shellcode (assembly instructions) that involves a series of executions to unpack itself. The assembly instructions involve multiple push and mov instructions. The purpose of this is to copy the Brute Ratel C4 code (x64 assembly) onto the stack eight bytes at a time and eventually reassemble it into a memory space for execution – a DLL with a stripped MZ header. Using a series of push and mov instructions evades in-memory scanning as the shellcode is assembled in blocks versus the entire code base being exposed for scanning. The entry point of the decrypted payload is the following:

The unpacking involves 25,772 push and 25,769 mov instructions. When finished, the code performs the following.

Using API hashing (ROR13 – rotate right 13) looks up the hash for NtAllocateVirtualMemory. All API calls are made via API hash lookups.

Resolves the system call index from the System Service Dispatch Table (SSDT) for NtAllocateVirtualMemory. All Windows API functions are made via syscalls, which is a feature of Brute Ratel C4 (Syscall Everything).

Calls the Windows API NtAllocateVirtualMemory, allocating 0x3000 bytes of memory.

Makes a second Windows API call into NtAllocateVirtualMemory, which allocates memory for the configuration file used by Brute Ratel C4.

Copies the shellcode that was pushed onto the stack in the previous steps to the newly allocated memory segment.

Changes the protection of the newly allocated memory block using Windows API call NtProtectVirtualMemory.

Calls NtCreateThreadEx with the start address of the newly allocated memory and passes the configuration data as a parameter.

Finished.

The following is a snippet of the code that calls NtCreateThreadEx and starts the execution of the second-stage shellcode.

The configuration data is passed as a parameter to the start address of the new thread. This data includes the encrypted configuration settings for Brute Ratel C4. The encrypted contents are the following:

The data is base64-encoded and RC4-encrypted. The 16-byte RC4 decryption key is: bYXJm/3#M?:XyMBF

The decrypted configuration file is:

Contents of decrypted configuration file

Each parameter is delineated with a pipe | character, and one of the values is the IP seen earlier of 174.129.157[.]251 and port of 443.

Target Network Infrastructure

The IP 174.129.157[.]251 is hosted on Amazon AWS, and Palo Alto Networks Cortex Xpanse history shows the IP had TCP port 443 open from April 29, 2022, until May 23, 2022, with a self-signed SSL certificate impersonating Microsoft Security:

Once the SSL handshake to IP 174.129.157[.]251 is complete, the following data is sent via HTTP POST to the Brute Ratel C4 listener port.

Identifying OneDrive.Update

To identify the decrypted in-memory payload as being associated with Brute Ratel C4, we conducted hunting based on the unique in-memory assembly instructions, push and mov. These instructions are used to build the second layer of shellcode. Searching across VirusTotal, we found a second sample with the same push and mov instructions:

Initially, what stood out to us was the filename containing the word “badger.” According to the Brute Ratel C4 website, the word “badger” represents payloads used for remote access. When uploaded to VirusTotal, only two out of 66 vendors considered the sample malicious. Currently, 12 vendors identify the sample as malicious with eight classifying this sample as “Brutel,” further supporting that our in-memory code is somehow associated with that of Brute Ratel C4.

Side-by-side comparison of the entry point of badger\_x64.exe and our decrypted OneDrive.Update sample:

When badger\_x64.exe is finished with the push and mov instructions, it makes the same Windows API calls as OneDrive.Update using API hashing, but does not use direct syscall (a user configuration feature of Brute Ratel C4). Example of badger\_x64.exe:

Like the OneDrive.Update sample, the parameter passed to the calling thread is the configuration data for Brute Ratel C4. In this sample, the data is not base64-encoded or RC4-encrypted, and is passed in the clear. The following is the configuration used for this sample:

The configuration used for the OneDrive.Update sample

In this case, the sample is configured to communicate with IP 159.65.186[.]50 on TCP port 443.

Based on the following, we can conclude that OneDrive.Update is indeed associated with Brute Ratel C4.

The configuration file structure is the same and uses pipes to delineate fields.

Same Windows calling pattern used to run the second-stage shellcode via NtCreateThreadEx/CreateThread.

Function instruction match for copying shellcode to memory allocation.

Both samples of second-stage shellcode have the following strings referencing the word “badger.” Note: The OneDrive.Update sample RC4 encrypts these strings.

imp\_Badger

BadgerDispatch

BadgerDispatchW

BadgerStrlen

BadgerWcslen

BadgerMemcpy

BadgerMemset

BadgerStrcmp

BadgerWcscmp

BadgerAtoi

The Badger\* names match the export names listed on the BRc4 GitHub website.

The file badger\_x64.exe is a standalone x64 executable that runs Brute Ratel C4 (badger payload) while the decrypted OneDrive.Update file is the in-memory component of Brute Ratel C4 that is executed using the actor's modified DLL, version.dll.

Badger\_x64.exe Employment

After validating that badger\_x64.exe and OneDrive.Update were both BRc4 payloads, we set to work analyzing the employment of this second sample.

VirusTotal records revealed that the sample was uploaded by a web user in Ukraine on May 20, 2022. Coincidentally, this happens to be one day after the ​​Roshan\_CV.ISO file was uploaded.

As noted above, the sample was configured to call home to 159.65.186[.]50 on port 443. Palo Alto Networks Cortex Xpanse history shows that this port was open from May 21-June 18, 2022, with the same “Microsoft Security” self-signed SSL certificate seen above. Given this timeline, it's worth noting that the sample was actually uploaded to VirusTotal prior to the C2 infrastructure being configured to listen for the callbacks.

Evaluating netflow connections for 159.65.186[.]50 during this time window revealed several connections to ports 22, 443 and 8060 originating from a Ukrainian IP (213.200.56[.]105). We assess this Ukrainian address is likely a residential user IP that was leveraged to administer the C2 infrastructure. A deeper look at connections in and out of 213.200.56[.]105 further revealed several flows over UDP port 33445. This port is commonly used by Tox, a secure peer-to-peer chat and video application that offers end-to-end encryption.

Examining additional connections to port 443 on 159.65.186[.]50, we identified several suspected victims including an Argentinian organization, an IP television provider providing North and South American content, and a major textile manufacturer in Mexico. Coincidentally, recent attempts to browse the textile manufacturer’s website result in a 500 internal server error message.

Given the geographic dispersion of these victims, the upstream connection to a Ukrainian IP and several other factors, we believe it is highly unlikely that BRc4 was deployed in support of legitimate and sanctioned penetration testing activities.

Other Samples and Infrastructure

Over the past year, the fake Microsoft Security X.509 certificate has been linked to 41 IP addresses. These addresses follow a global geographic dispersion and are predominantly owned by large virtual private server (VPS) hosting providers. Expanding our research beyond the two samples discussed above, we have also identified an additional seven samples of BRc4 dating back to February 2021.

Protections and Mitigations

For Palo Alto Networks customers, our products and services provide the following coverage associated with this group:

Threat Prevention provides protection against Brute Ratel C4. The "Brute Ratel C4 Tool Command and Control Traffic Detections" signature is threat ID 86647.

Cortex XDR detects and protects endpoints from the Brute Ratel C4 tool.

WildFire cloud-based threat analysis service accurately identifies Brute Ratel C4 samples as malware.

The emergence of a new penetration testing and adversary emulation capability is significant. Yet more alarming is the effectiveness of BRc4 at defeating modern defensive EDR and AV detection capabilities.

Over the past 2.5 years this tool has evolved from a part-time hobby to a full-time development project with a growing customer base. As this customer base has expanded into the hundreds, the tool has gained increased attention across the cybersecurity domain from both legitimate penetration testers as well as malicious cyber actors.

The analysis of the two samples described in this blog, as well as the advanced tradecraft used to package these payloads, make it clear that malicious cyber actors have begun to adopt this capability. We believe it is imperative that all security vendors create protections to detect BRc4 and that all organizations take proactive measures to defend against this tool.

Palo Alto Networks has shared these findings, including file samples and indicators of compromise, with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Note that the Microsoft name and logo shown are an attempt to impersonate a legitimate organization and do not represent an actual affiliation with Microsoft. This impersonation does not imply a vulnerability in Microsoft’s products or services.