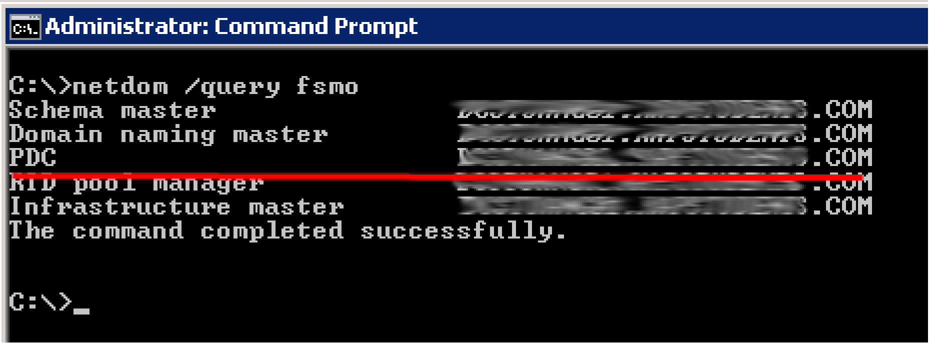
Active Directory

Sync Time from External Time Source

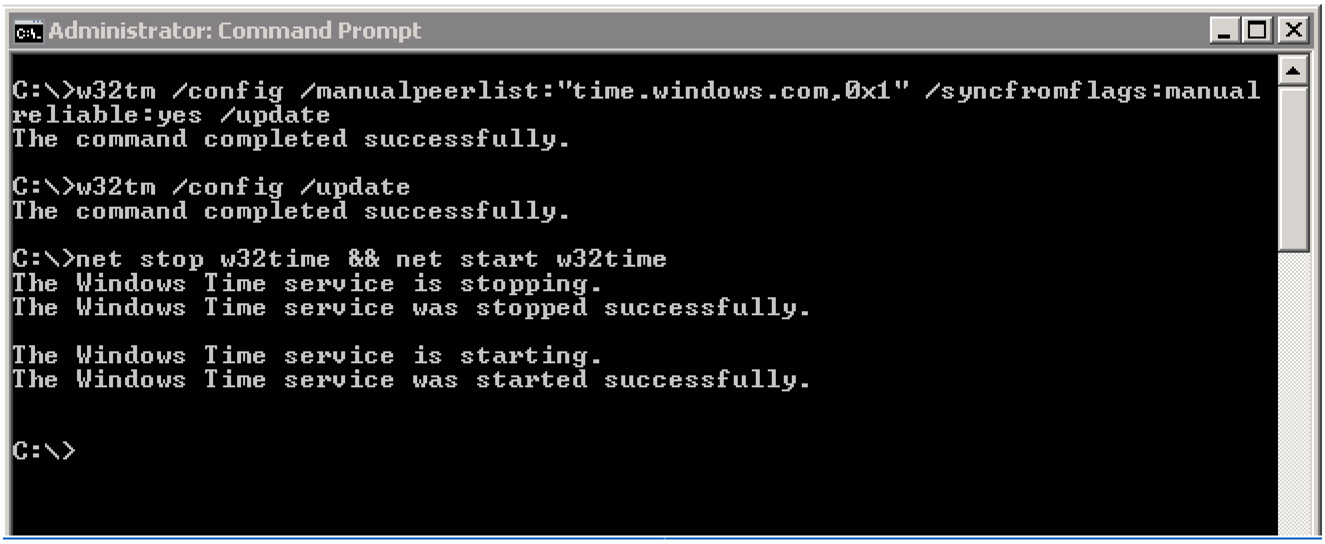
By default, all machines in the domain will sync time from the domain controller which is the internal time server - if you have more than one DC then time will sync from the DC that holds the PDC emulator FSMO role. To check which DC is PDC emulator in your domain you need to run **netdom /query fsmo** command like so:



```
C:\>netdom /query fsmo
Schema master               [redacted].COM
Domain naming master        [redacted].COM
PDC                         [redacted].COM
RID pool manager            [redacted].COM
Infrastructure master       [redacted].COM
The command completed successfully.


C:\>_
```

Once PDC emulator role is established there is few commands we need to run in order for time to sync, these are (run on PDC emulator):

```
w32tm /config /manualpeerlist:"time.windows.com,0x1" /syncfromflags:manual /reliable:yes /update
w32tm /config /update
net stop w32time && net start w32time
```

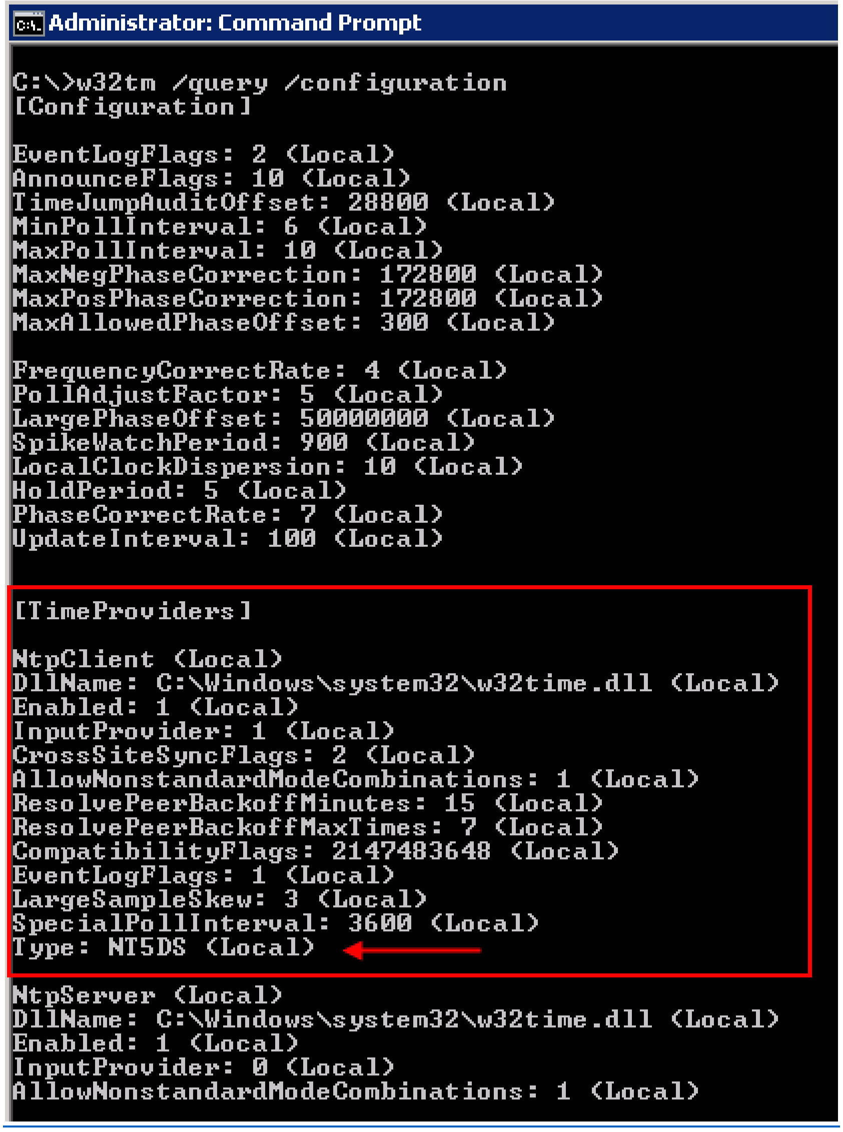


```
C:\>w32tm /config /manualpeerlist:"time.windows.com,0x1" /syncfromflags:manual
reliable:yes /update
The command completed successfully.

C:\>w32tm /config /update
The command completed successfully.

C:\>net stop w32time && net start w32time
The Windows Time service is stopping.
The Windows Time service was stopped successfully.

The Windows Time service is starting.
The Windows Time service was started successfully.


C:\>
```

IF you need to add more than one NTP server then the peer list entries are space delimited, example:

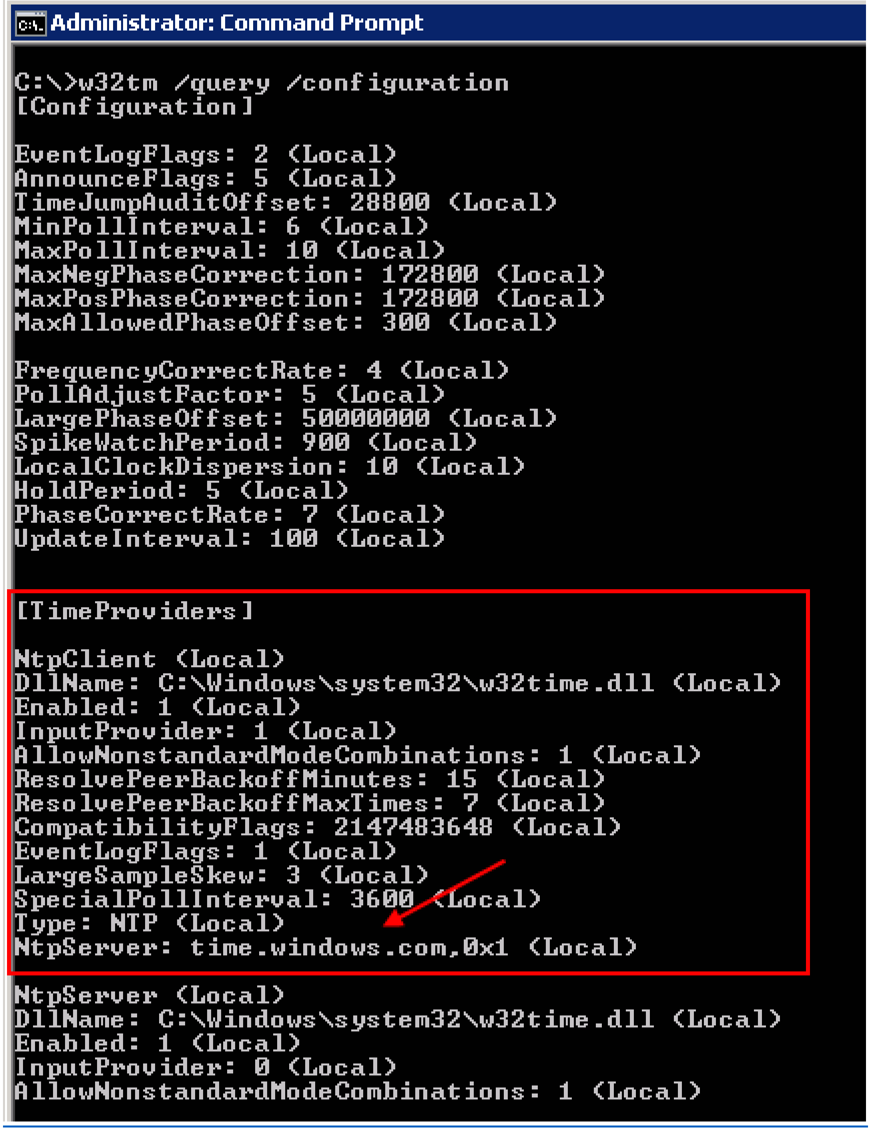
```
"0.ntp.pool.org,0x1 1.ntp.pool.org,0x1 2.ntp.pool.org,0x1"
```

Once completed Windows time service should begin synchronizing time on the domain controller(s) with external source. To view the time configuration you can use **w32tm /query /configuration** command. In my case, my time was not synced with external time server:



```
C:\>w32tm /query /configuration
[Configuration]

EventLogFlags: 2 (Local)
AnnounceFlags: 10 (Local)
TimeJumpAuditOffset: 28800 (Local)
MinPollInterval: 6 (Local)
MaxPollInterval: 10 (Local)
MaxNegPhaseCorrection: 172800 (Local)
MaxPosPhaseCorrection: 172800 (Local)
MaxAllowedPhaseOffset: 300 (Local)

FrequencyCorrectRate: 4 (Local)
PollAdjustFactor: 5 (Local)
LargePhaseOffset: 50000000 (Local)
SpikeWatchPeriod: 900 (Local)
LocalClockDispersion: 10 (Local)
HoldPeriod: 5 (Local)
PhaseCorrectRate: 7 (Local)
UpdateInterval: 100 (Local)


[TimeProviders]

NtpClient (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 1 (Local)
InputProvider: 1 (Local)
CrossSiteSyncFlags: 2 (Local)
AllowNonstandardModeCombinations: 1 (Local)
ResolvePeerBackoffMinutes: 15 (Local)
ResolvePeerBackoffMaxTimes: 7 (Local)
CompatibilityFlags: 2147483648 (Local)
EventLogFlags: 1 (Local)
LargeSampleSkew: 3 (Local)
SpecialPollInterval: 3600 (Local)
Type: NT5DS (Local)

NtpServer (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 1 (Local)
InputProvider: 0 (Local)
AllowNonstandardModeCombinations: 1 (Local)
```

and after I made the changes:



```
C:\>w32tm /query /configuration
[Configuration]

EventLogFlags: 2 (Local)
AnnounceFlags: 5 (Local)
TimeJumpAuditOffset: 28800 (Local)
MinPollInterval: 6 (Local)
MaxPollInterval: 10 (Local)
MaxNegPhaseCorrection: 172800 (Local)
MaxPosPhaseCorrection: 172800 (Local)
MaxAllowedPhaseOffset: 300 (Local)

FrequencyCorrectRate: 4 (Local)
PollAdjustFactor: 5 (Local)
LargePhaseOffset: 50000000 (Local)
SpikeWatchPeriod: 900 (Local)
LocalClockDispersion: 10 (Local)
HoldPeriod: 5 (Local)
PhaseCorrectRate: 7 (Local)
UpdateInterval: 100 (Local)


[TimeProviders]

NtpClient (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 1 (Local)
InputProvider: 1 (Local)
AllowNonstandardModeCombinations: 1 (Local)
ResolvePeerBackoffMinutes: 15 (Local)
ResolvePeerBackoffMaxTimes: 7 (Local)
CompatibilityFlags: 2147483648 (Local)
EventLogFlags: 1 (Local)
LargeSampleSkew: 3 (Local)
SpecialPollInterval: 3600 (Local)
Type: NTP (Local)
NtpServer: time.windows.com,0x1 (Local)

NtpServer (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 1 (Local)
InputProvider: 0 (Local)
AllowNonstandardModeCombinations: 1 (Local)
```

all was set to sync from **time.windows.com**. From workstation point of view to configure a client computer for automatic domain time synchronization:

```
w32tm /config /syncfromflags:domhier /update
```

and to check if its syncing:

```
w32tm /monitor
```

and to re-sync:

```
w32tm /resync
```

If there’re any errors then these will be written to Event Viewer - please check if you’re having issues.

Get List of Locked Out Users and their Clients and Kill the Sessions

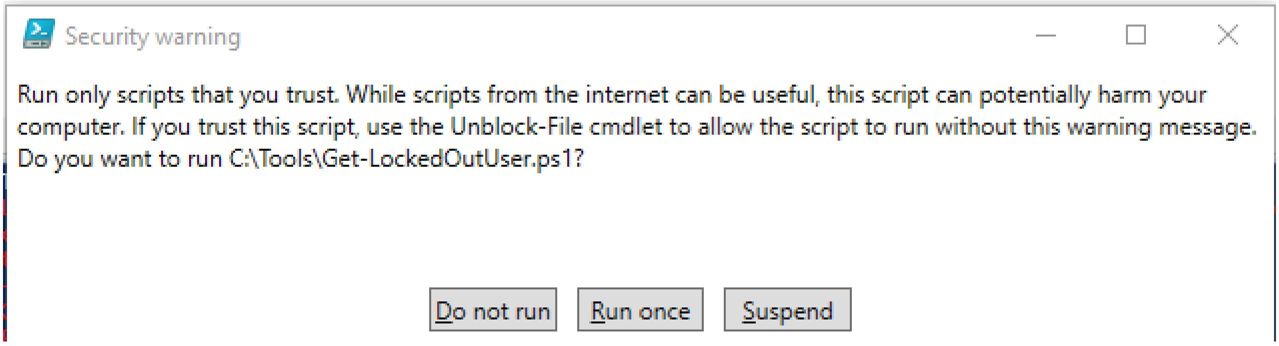
Get List of Locked Out Users

Download the Get-LockedOutUser.ps1 script and run it in an administrator powershell prompt:

```
PS C:\Tools> .\Get-LockedOutUser.ps1
```

In the **Security Warning**, click on **Run once** (**Figure 1**):

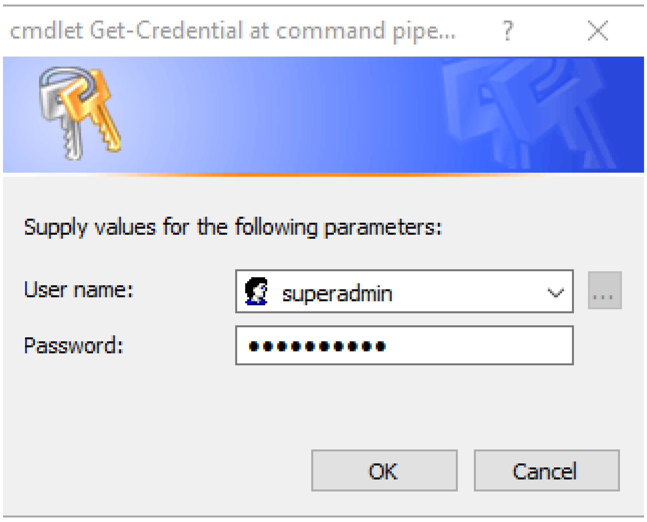
Figure 1



Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning message. Do you want to run C:\Tools\Get-LockedOutUser.ps1?

In the **cmdlet Get-Credential at command pipeline** prompt, supply Administrator credentials (**Figure 2**):

Figure 2



Supply values for the following parameters:

User name: superadmin

Password: ●●●●●●●●●●●

Wait for a bit while it parses the domain controller logs and you should see an output similar to below where the **UserName** field reflects the locked out username and the **ClientName** field reflects the client machine the lockout was generated:

```
TimeCreated         UserName       ClientName
-----------         --------       ----------
1/27/2021 9:15:39 AM user1□□□ wkstation1
1/27/2021 8:20:47 AM user2□□    wkstation25
1/27/2021 8:15:27 AM user3 □□□ wkstation11
```

Kill the Sessions

From a administrator command prompt, get the session ID of the logged in user from the machine name from the powershell output above by using **qwinsta** to query the user session on the client machine you wish to kill the session from:

```
qwinsta /server:wkstation1
```

You should get an output similar to below:

```
 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
 services                                  0  Disc
 console           user1□                  1  Disc
 rdp-tcp                               65536  Listen
```

If we were to kill the session for **user1**, then the session ID would be **1**. We can kill the session by using **rwinsta**:

```
rwinsta 1 /server:wkstation1
```