Find the source of AD account lockouts

In this post, we will learn how to determine the source of account lockouts in AD. Note that you need domain admin rights to perform the steps mentioned in this post.

Background information

When incorrect password attempts exceed the account lockout threshold configured in your domain, the user account is locked out and an event ID *4740* is recorded in the Security log of the domain controllers. If audit logging is also enabled on client computers, event ID *4625* is recorded on the client computer as well. As you might already know, the event log contains a lot of useful information, such as the name of the user account, the name of the domain controller, the name of the source computer, the timestamp, etc. If you have hundreds or thousands of computers in your AD environment, it isn't feasible to query all client computers. You can first query the domain controllers to find the computer name or IP address of the source computer on which the account lockout occurred. This is what we are going to do in this post.

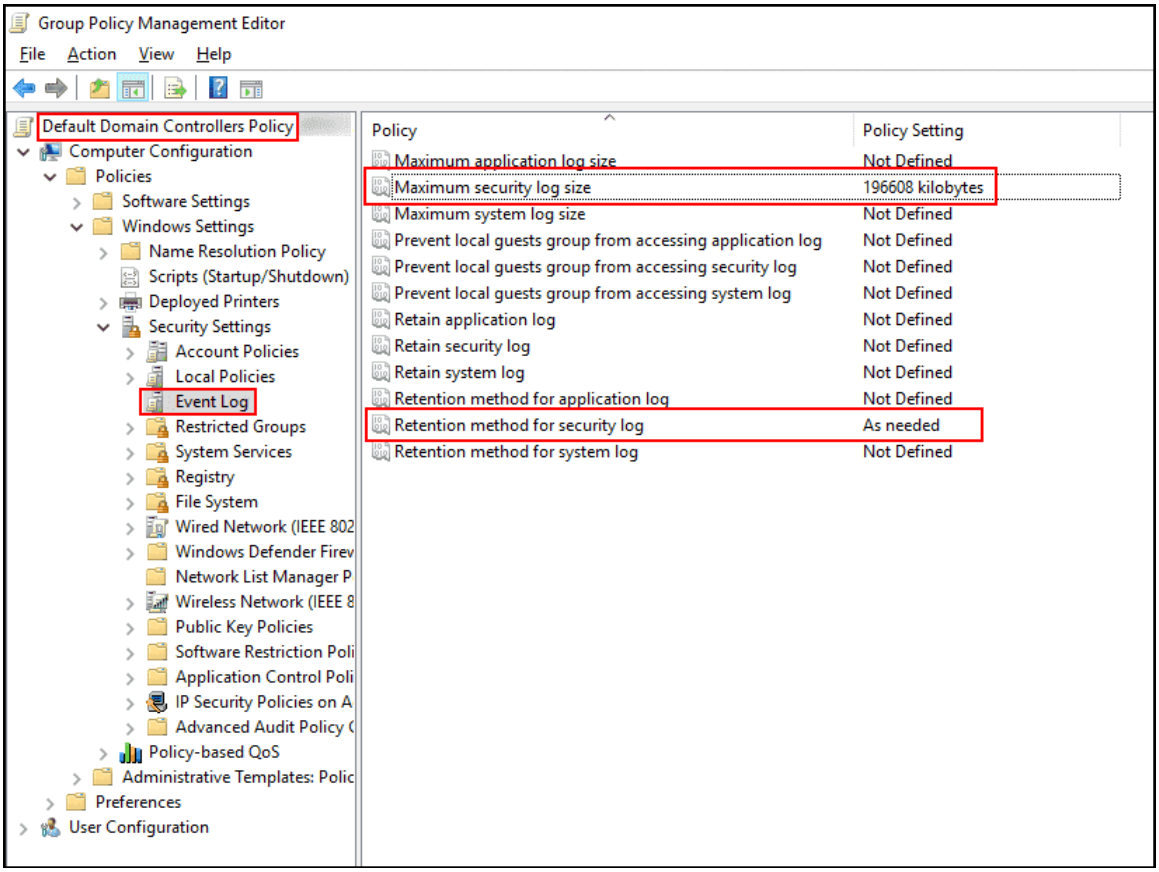
Reasons for account lockouts

It is obvious that account lockout occurs when incorrect password attempts exceed the defined threshold, but there could be various underlying reasons. The most common are as follows:

- Repeated incorrect password attempts
- Drive mapping using old credentials
- Scheduled tasks
- Programs or services using old credentials
- Cached or saved credentials in Windows Credential Manager
- Slow AD replication

Increase the size of the Security log

This step is optional but recommended, particularly if you have a large AD environment. The default size of the Security log on a domain controller is 128 MB, and the old events are overwritten automatically when the log is full. So you need to increase the Security log size to make sure it can store enough logs until you can identify the source of the account lockout. To do so, you can modify the default domain controller policy, as shown in the following screenshot:

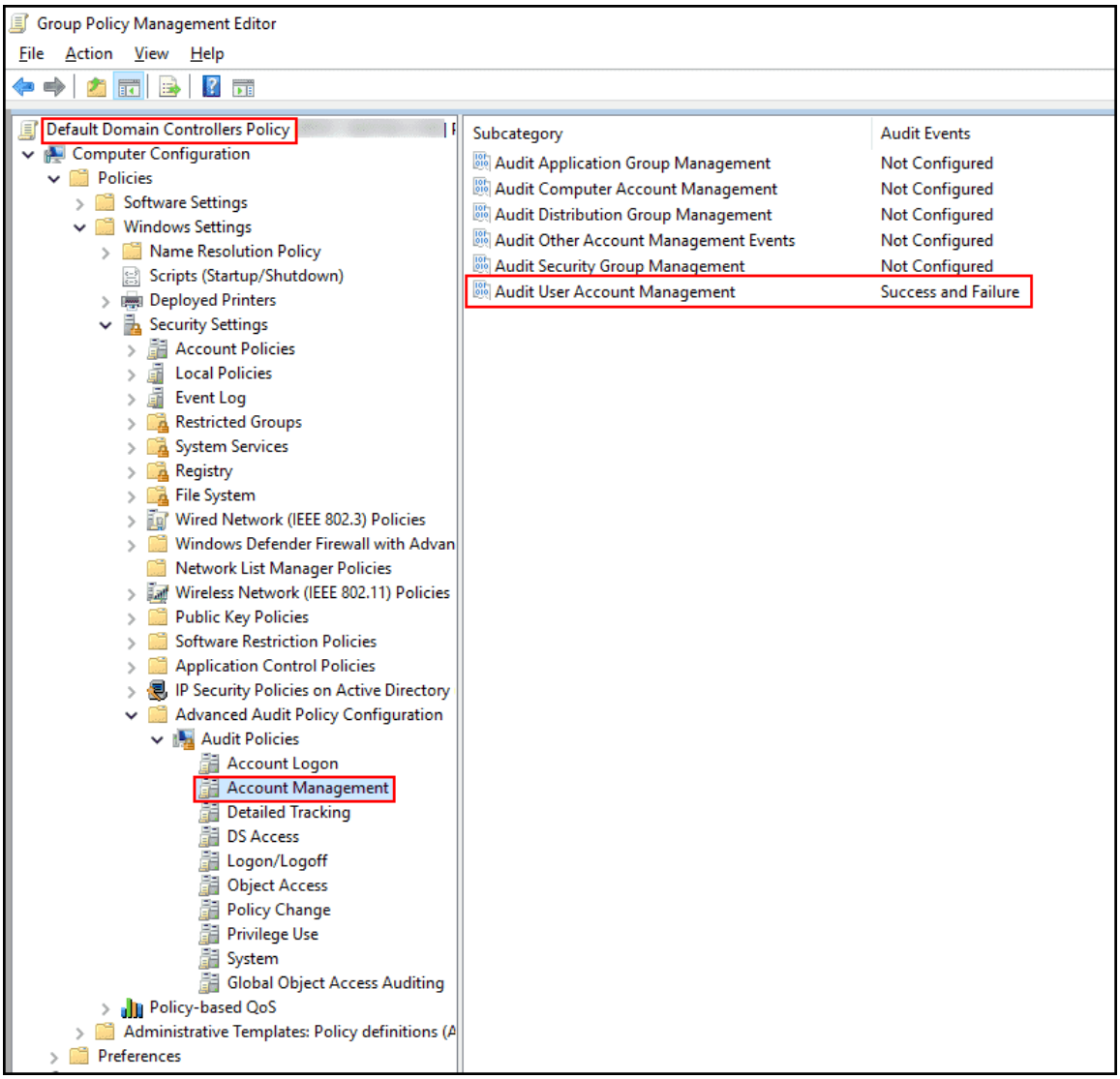


Increasing the Security log size on domain controllers

Enable audit logging on domain controllers

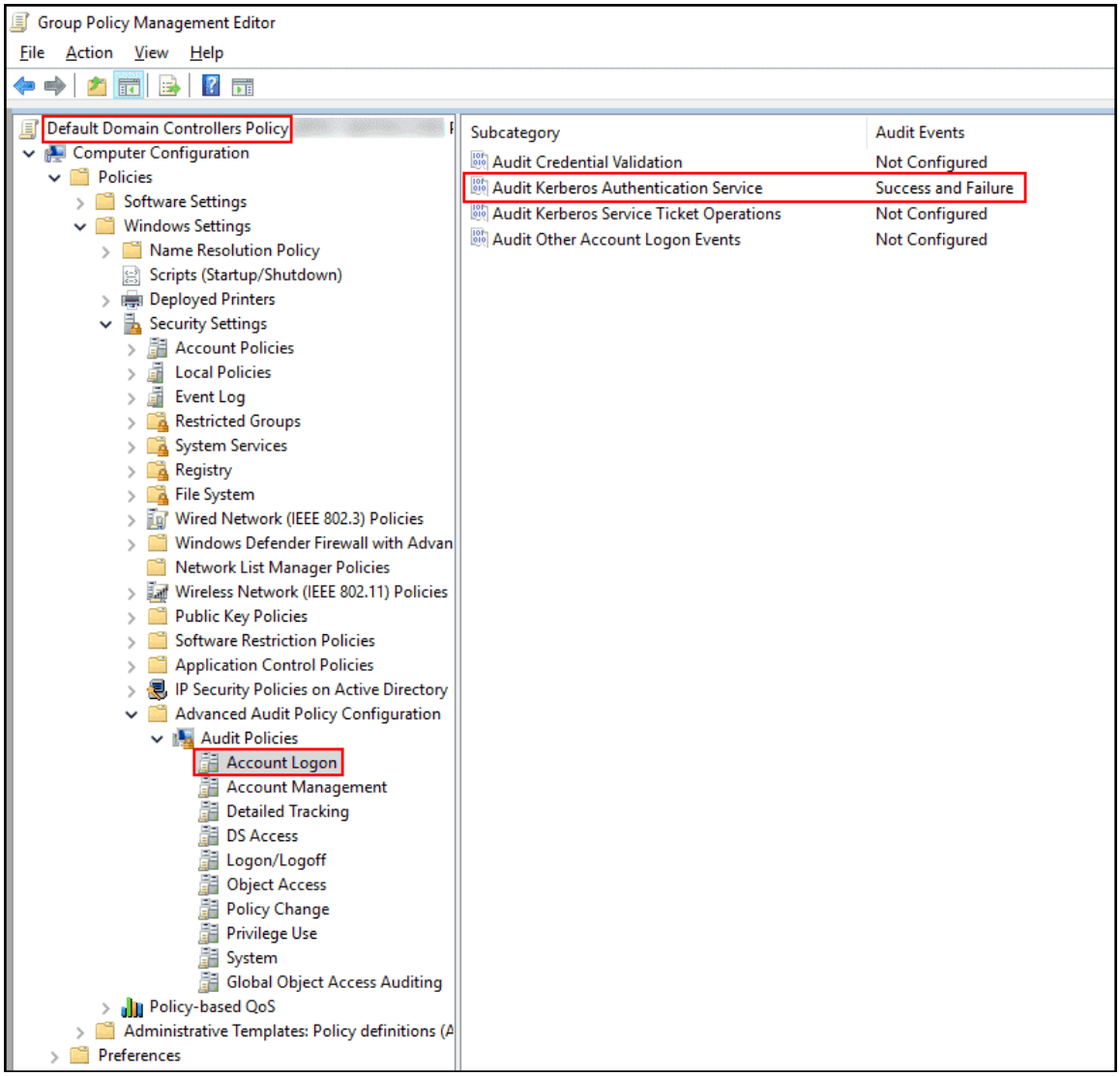
To trace the account lockout source, you need to enable audit logging on your domain controllers. The simplest way to achieve this is to modify the default domain controller policy. To do so, follow these steps:

- Log on to any domain controller and launch the Group Policy Management Console (*gpmc.msc*).
- Navigate to *Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies > Account Management.*
- Now enable the *Audit User Account Management* subcategory for *Success and Failure*, as shown in the screenshot below:



Enabling the Audit User Account Management policy on domain controllers

You might not be able to find event ID 4740 on the domain controllers. If this happens, you can also enable the Audit Kerberos Authentication Service policy available under *Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies > Account Logon.*



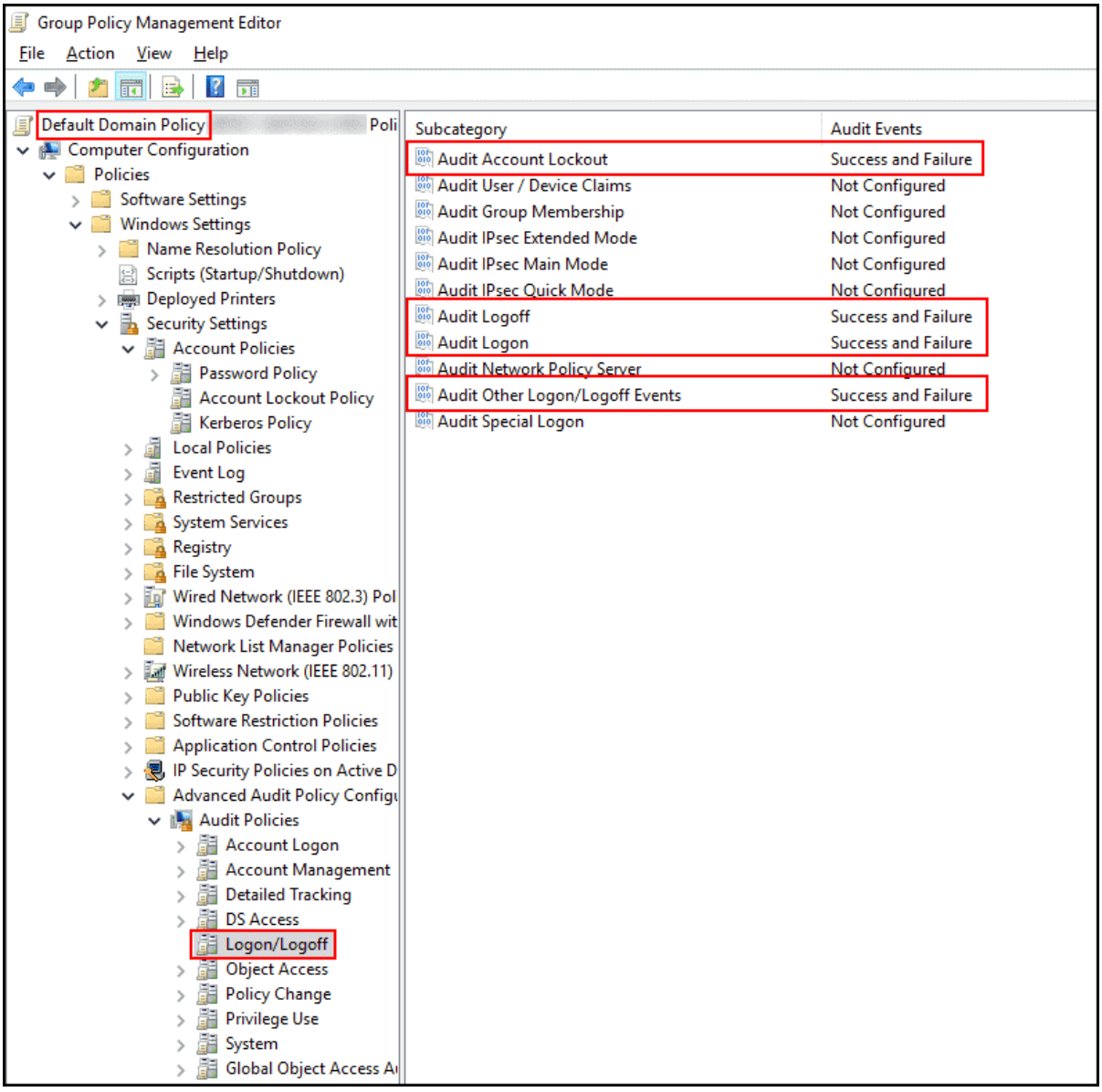
Enabling the Audit Kerberos Authentication Service policy on domain controllers

Once enabled, it will generate event ID 4771 with the Kerberos pre-authentication failed message. This event shows you the IP address of the source computer that failed Kerberos authentication.

Enable audit logging on client computers

Now modify the default domain policy to enable audit logging on client computers, as shown below:

- Navigate to *Computer Configuration* > *Policies* > *Windows Settings* > *Security Settings* > *Advanced Audit Policy Configuration* > *Audit Policies* > *Logon/Logoff*.
- Now enable *Audit Account Lockout, Audit Logoff, Audit Logon,* and *Audit Other Logon/Logoff Events*, as shown in the following screenshot:

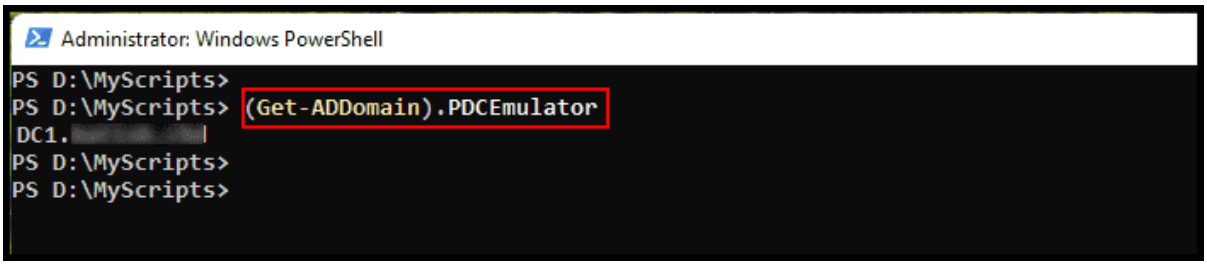


Enabling audit logs on client computers

Find account lockout source

Now that you have enabled auditing on both domain controllers and client computers, here comes the most interesting part. AD account lockouts are processed on the PDC emulator role holder domain controller, so most account lockout events will be available on it for you. You can use the following PowerShell command to determine the PDC role holder in your domain:

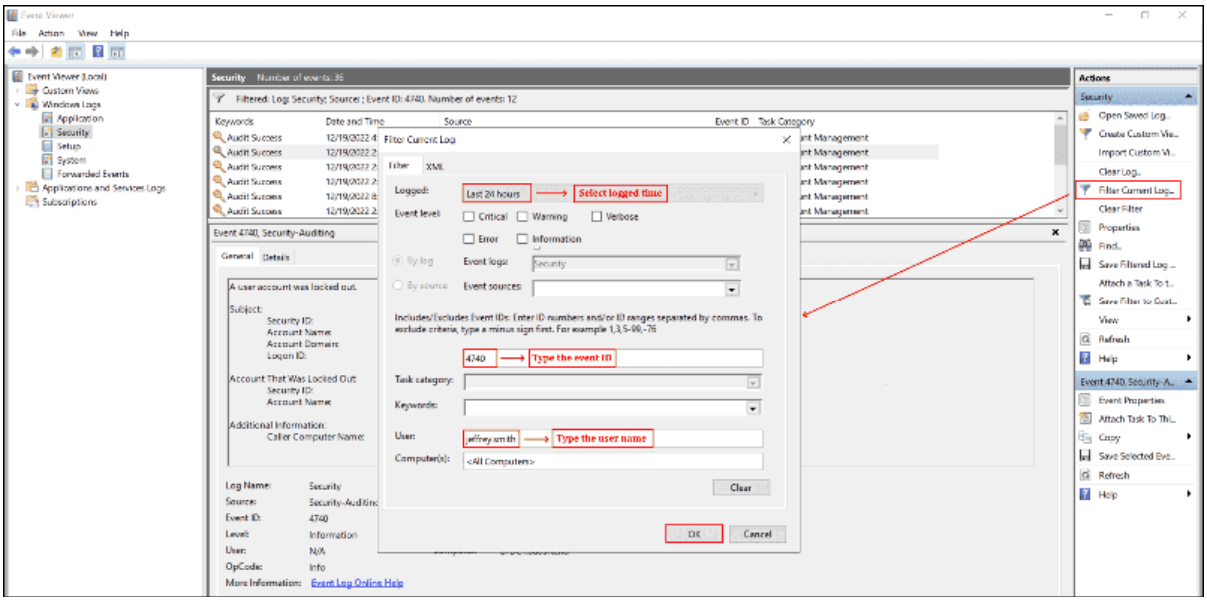
```
(Get-ADDomain).PDCEmulator
```



Determine PDC emulator role holder domain controller with PowerShell

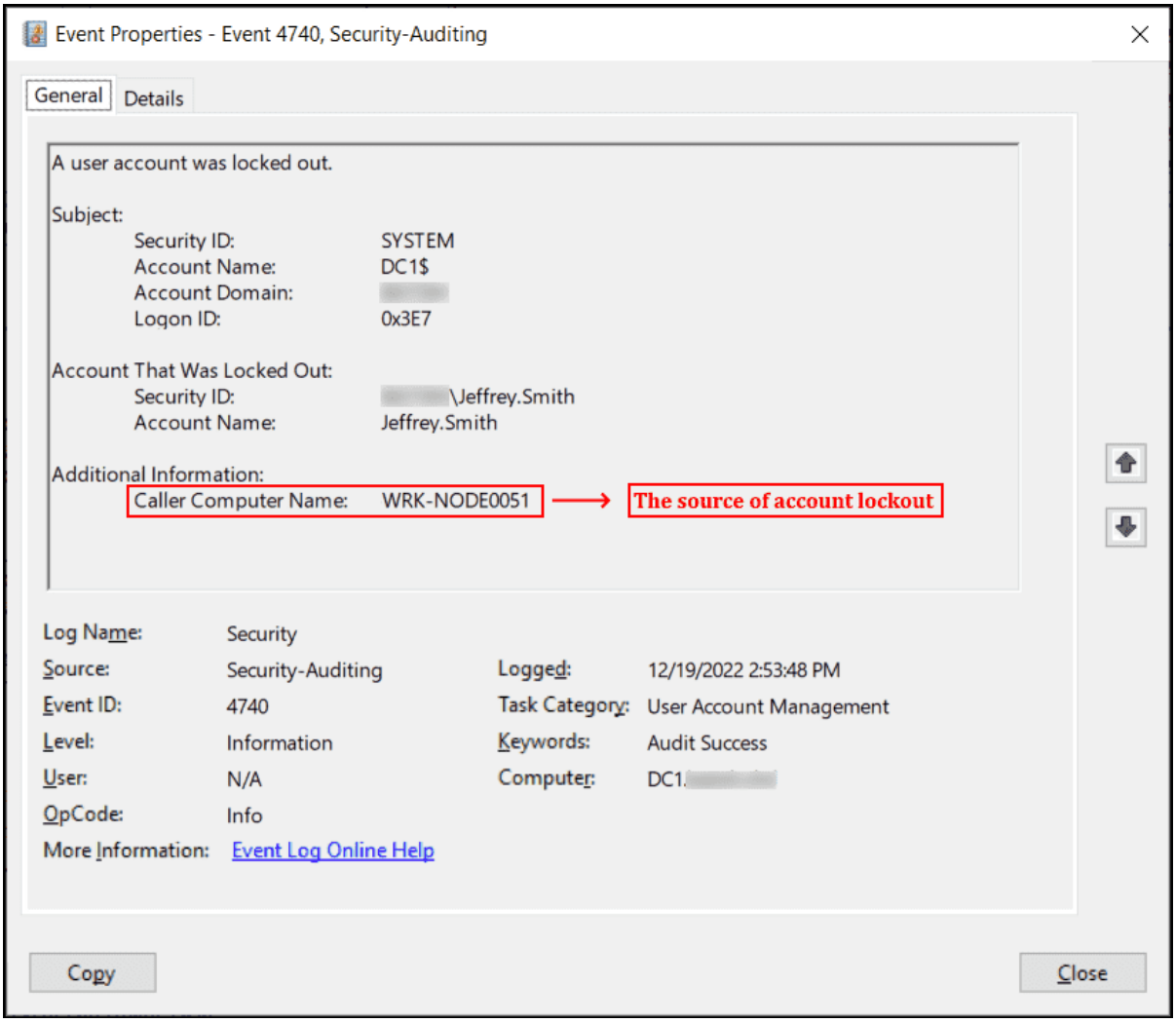
Once you know the name of the PDC emulator, follow these steps to find the source computer responsible for the account lockout:

- Log on to the PDC emulator and launch the event viewer.
- Expand Windows Logs and select *Security*.
- Now click *Filter Current Log* in the Actions pane, configure the filter criteria as shown in the screenshot, and click OK.



Filtering the event log to view only lockout events

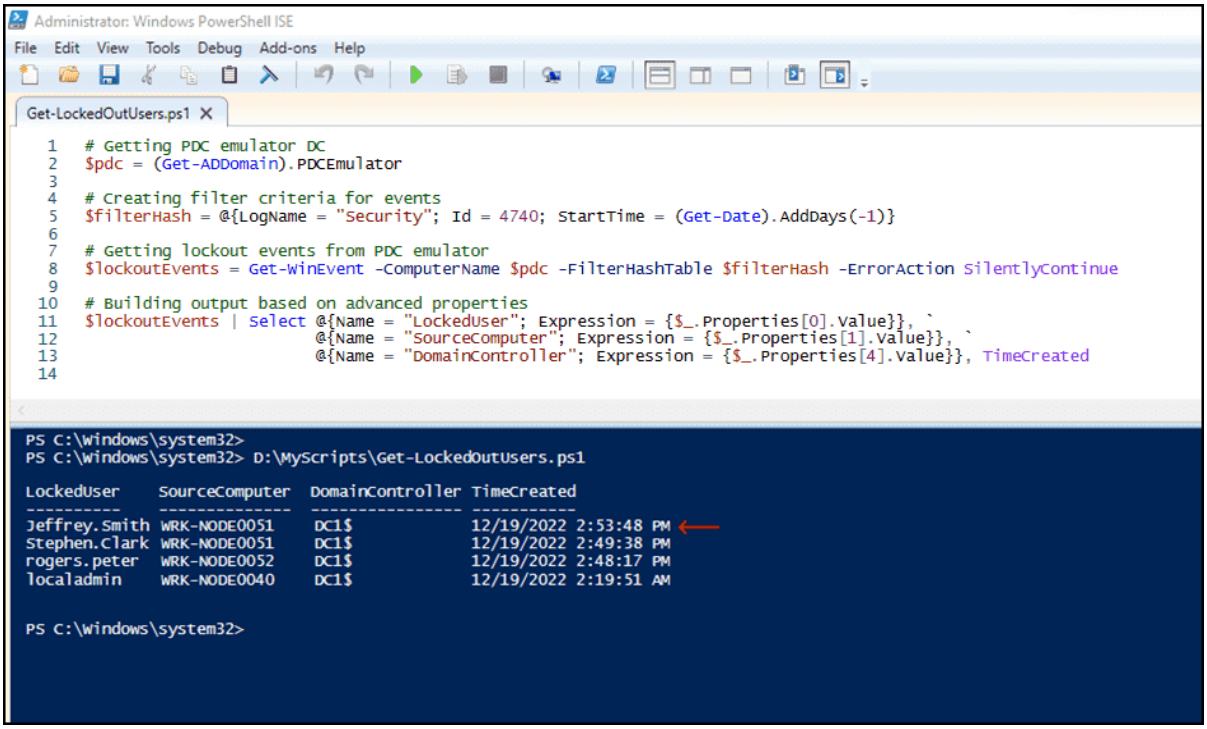
- The log will now show account lockout events for the specified user.
- Double-click the event to view complete information about it. You will see *Caller Computer Name* under the Additional Information section. This is the source computer on which the account lockout occurred.



Event detail showing the caller computer name as the account lockout source

Alternatively, to get the events with PowerShell, you can use the following code snippet:

```
# Getting the PDC emulator DC
$pdc = (Get-ADDomain).PDCEmulator
# Creating filter criteria for events
$filterHash = @{LogName = "Security"; Id = 4740; StartTime = (Get-Date).AddDays(-1)}
# Getting lockout events from the PDC emulator
$lockoutEvents = Get-WinEvent -ComputerName $pdc -FilterHashTable $filterHash -ErrorAction SilentlyContinue
# Building output based on advanced properties
$lockoutEvents | Select @{Name = "LockedUser"; Expression = {$_.Properties[0].Value}}, `
@{Name = "SourceComputer"; Expression = {$_.Properties[1].Value}}, `
@{Name = "DomainController"; Expression = {$_.Properties[4].Value}}, TimeCreated
```

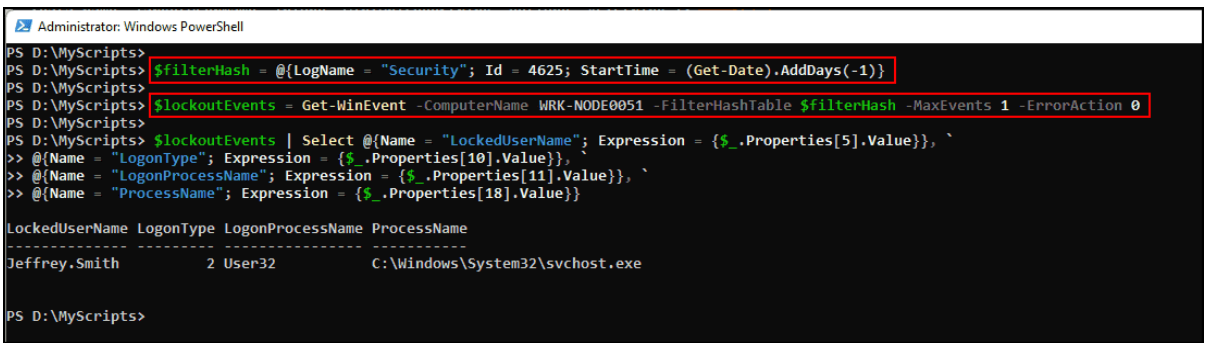


Finding the source computer responsible for AD account lockouts with PowerShell

We first created the filter criteria to search for *event ID 4740* and a log time of 24 hours in the *Security* log. Then, we used the *Get-WinEvent* cmdlet to pull the logs based on the filter hash and used calculated properties to build the output. I will discuss these properties later in this post. This code snippet gave us the locked-out user name, source computer name, DC name, and the timestamp of when the event was created.

Once you know the source computer, you can query that computer and pull the events based on event ID *4625,* which will show you the name of the actual process causing the account lockout. See the updated code snippet below.

```
# Creating filter criteria for events
$filterHash = @{LogName = "Security"; Id = 4625; StartTime = (Get-Date).AddDays(-1)}
# Getting lockout events from the source computer
$lockoutEvents = Get-WinEvent -ComputerName WRK-NODE0051 -FilterHashTable $filterHash -MaxEvents 1 -
ErrorAction 0
# Building output based on advanced properties
$lockoutEvents | Select @{Name = "LockedUserName"; Expression = {$_.Properties[5].Value}}, `
@{Name = "LogonType"; Expression = {$_.Properties[10].Value}}, `
@{Name = "LogonProcessName"; Expression = {$_.Properties[11].Value}}, `
@{Name = "ProcessName"; Expression = {$_.Properties[18].Value}}
```



Finding the process name responsible for AD account lockouts on a remote computer with PowerShell

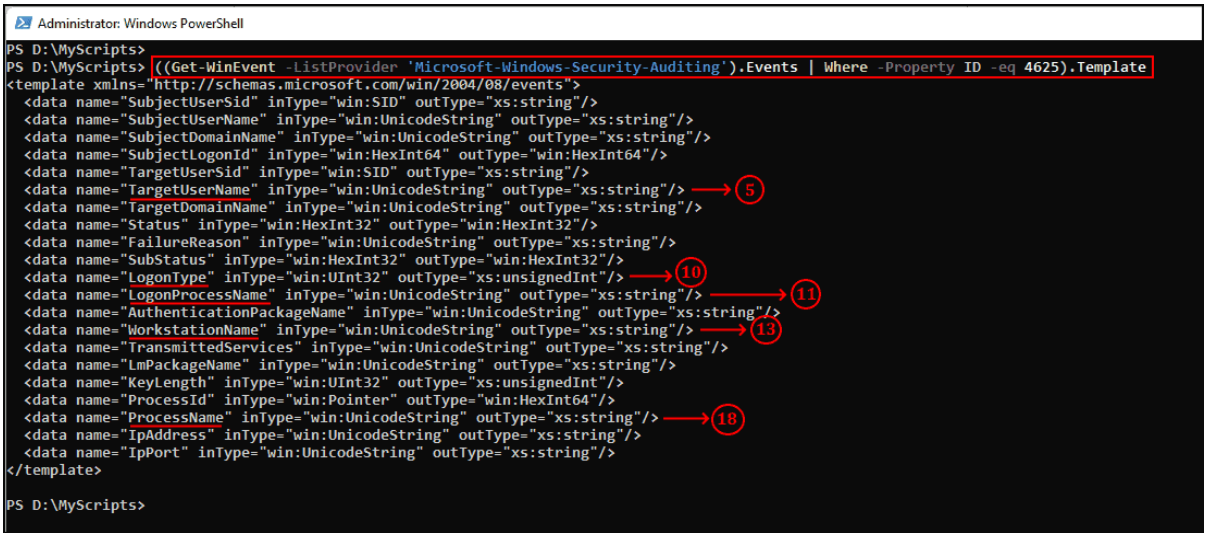
The above screenshot shows that LogonProcessName is *User32*, the ProcessName is *svchost.exe*, and LogonType is *2*, which means it was an interactive logon. This indicates that the user was trying to log on interactively with a bad password, which caused the user account to lock out. The following table shows the possible values for the *LogonType* field:

Logon Type	Title	Description
2	Interactive	An interactive logon to a local computer.
3	Network	A logon from the network to a local computer.
4	Batch	A batch logon type means a process is executed on behalf of a user account.
5	Service	A service was started by a service control manager.
7	Unlock	A local workstation is unlocked.
8	NetworkCleartext	A user logged on to a local computer from the network, and the password was passed in cleartext.
9	NewCredentials	A cloned token was used so the new session has the same user identity locally but uses a different credential for remote network connections. This logon type is recorded when you use *RunAs* with the */netonly* switch.
10	RemoteInteractive	A user was logged on using Remote Desktop or Terminal Services.
11	CachedInteractive	A user was logged on using cached credentials without contacting the domain controller to verify credentials.

So, you can determine what might be causing account lockouts just by looking at the LogonType field. If you are wondering what these properties are that I used to build the output, let me briefly explain it now.

These properties are defined in the security auditing XML template used by event logs. To view these properties, you can use the following command:

```
((Get-WinEvent -ListProvider "Microsoft-Windows-Security-Auditing").events | Where -Property ID -eq 4625).template
```



Understanding the properties of the XML template used by event logs

The above command displays an XML template for event ID 4625, as you can see in the screenshot. To see the template for event ID 4740, you would use the *((Get-WinEvent -ListProvider "Microsoft-Windows-Security-Auditing").events | Where -Property ID -eq 4740).template* command instead.

The Get-WinEvent cmdlet that we used in our snippet essentially stored these properties in an array, and we called them by their index number. For example, to get the target username from the event log property,

I used *{$_.properties[5].value}*, since it was located at index 5 (as marked in the screenshot above). To get the logon type, I used *{$_.properties[10].value},* and so on. The same idea was used with event ID 4740, which we pulled earlier from the PDC emulator.