


Active Directory Attributes: Last Logon

 blog.netwrix.com/2022/11/03/active-directory-logon-attributes

Active Directory user objects possess a number of logon metadata attributes that are valuable for Active Directory audit reporting and administration. For example, they are commonly used to identify user accounts that have been inactive for a significant period, or as “stale” accounts.

However, each logon metadata attribute has some unique behaviors that need to be understood. Otherwise, organizations can end up with reports that are confusing at best and inaccurate or otherwise misleading at worst.

Handpicked related content:

[How to Monitor User Logons in a Domain](#)

This blog post explains the behaviors of each Active Directory user object logon metadata attribute, methods for reviewing them, and the potential uses and misuses of each.

Last Logon AD Attribute

The Last-Logon attribute contains a Windows FileTime representation of the last time a domain controller successfully authenticated the user. It is the granddaddy of user logon metadata, having been around since the first version Active Directory.

Using the PowerShell command below, you can retrieve the last logon time and other user properties on a domain controller:

```
Get-ADUser -Filter * -Properties lastLogon | Select samaccountname,  
@{Name="lastLogon";Expression={[datetime]::FromFileTime($_.lastLogon)}}
```

The Last-Logon attribute is updated every time a domain controller successfully processes a logon request, so it might appear that it provides the perfect way to accurately identify stale user accounts. However, there’s a big caveat that needs to be taken into account.

AD Last-Logon is not a replicated attribute; each domain controller (DCs) maintains its own version of the attribute for any specific user. This behavior is intentional — the increase in replication traffic necessary to keep this attribute in sync across a network’s domain controllers would have been overwhelming, especially at its time of introduction twenty years ago. But this behavior is also the reason that it is necessary to be careful when using this attribute to report on stale user accounts.

Because Last-Logon is not replicated (domain controllers don’t exchange this information), attribute values can be interpreted only in the context of the domain controller being queried. That is, the attribute’s value is not necessarily the last time the user logged in, but rather the last time the user successfully authenticated through the

domain controller being checked. Similarly, the attribute having a value is zero does not necessarily mean that the user has never logged in; it may mean that the domain controller that returned the value has never processed a login request from that user.

In short, while the Last-Logon attribute can be used for login-related auditing, accurate reporting will require querying every domain controller capable of processing login requests to identify the most recently updated value for any specific user account. Alternatively, you can use a third-party reporting solution, as discussed later in this article.

Last-Logon-Timestamp

The Last-Logon-Timestamp contains a Windows FileTime representation of a *recent* time the user logged on to a domain. This user attribute was introduced with Microsoft Windows Server 2003. Unlike the older Last-Logon attribute, the Last-Logon-Timestamp attribute is a replicated attribute; its value for any specific user is synced to every domain controller. This is a big improvement over the Last-Logon attribute. That means the best way to identify stale user accounts is to use the Last-Logon-Timestamp, right? Well, using this attribute comes with its own warning.

The gotcha with the Last-Logon-Timestamp attribute is that it is not always updated when a domain controller successfully processes a logon request. Instead, the attribute has a dynamic update frequency that is limited by the value of the ms-DS-Logon-Time-Sync-Interval attribute, which defaults to NOT SET and is treated as 14 days. It's not common for this attribute to have been changed, but if you're curious, you can easily identify its actual value using following PowerShell script:

```
$lastLogonReplicationInterval = (Get-ADDomain).LastLogonReplicationInterval
if( $lastLogonReplicationInterval )
{
    Write-Host "ms-DS-Logon-Time-Sync-Interval is set to
$( $lastLogonReplicationInterval ) days"
}
else {
    Write-Host "ms-DS-Logon-Time-Sync-Interval is not set and will be treated as
14 days"
}
```

In a domain with the default 14-day maximum update boundary, the Last-Logon-Timestamp is updated only when a domain controller successfully processes a logon request and the period since the attribute's last update is greater than somewhere between 9 and 14 days. The variation in that period is the result of a random percentage that is included in the logic that controls the update frequency. This behavior reflects a compromise between limiting the replication traffic necessary to keep this attribute in sync across a network's domain controllers and limiting the likelihood of having to replicate a significant number of users who had their Last-Logon-Timestamp updated at around the same time.

Here's a simplified example of the logic that controls the update frequency of the Last-Logon-Timestamp attribute:

```
(Current Datetime - Last-Logon-Timestamp) ? (ms-DS-Logon-Time-Sync-Interval - (Random % * 5 days))
```

In practice, the Last-Logon-Timestamp attribute will simplify login-related auditing and reporting. The only significant potential issue involves inactive user reporting. When used to identify inactive users, the threshold for staleness needs to exceed the domain's ms-DS-Logon-Time-Sync-Interval value by enough time to ensure that replication has been able to propagate any meaningful updates.

LastLogonDate (PowerShell)

Those familiar with PowerShell may recognize LastLogonDate, but you won't be able to find it anywhere in the Active Directory global catalog schema. This is because LastLogonDate is actually a locally calculated value that will display the replicated value of the Last-Logon-Timestamp attribute in a user-friendly format. Unsurprisingly, LastLogonDate has all of the benefits and all of the drawbacks of the Last-Logon-Timestamp attribute. However, since it does not require conversion from Windows DateTime, it is the best option for most user login-related audit reporting.

Active Directory Last Successful Interactive Logon Attribute

The attribute ms-DS-Last-Successful-Interactive-Logon-Time was introduced in Windows Server 2008, but many people are unfamiliar with it because it's disabled by default. When enabled, it contains the date and time of a user's last successful interactive logon. While this seems like an incredibly useful thing to enable, there are some compelling reasons for leaving it disabled, which I'll get to in a moment.

If you have a lab environment and want to play around with the ms-DS-Last-Successful-Interactive-Logon-Time attribute, you can enable the following: Computer Configuration ? Policies ? Administrative Templates ? Windows Components ? Windows Logon Options ? Display information about previous logons during user logon GPO. Then force a Group Policy update. Do not enable this setting in a production domain for fun; you'll have a bad time.

The first issue with the ms-DS-Last-Successful-Interactive-Logon-Time attribute is that its value is updated only when an *interactive* logon is authenticated (think "Ctrl-Alt-Del" logons). This means important authentication activities like network share logins and LDAP authentications will not trigger an update. Accordingly, if you use this attribute to generate logon-related audit reports, you're likely to get some inaccurate results. For example, reports identifying inactive user accounts are likely to list domain service accounts, which are generally very active — just in non-interactive ways. In short, this attribute makes stale user reporting really simple and reliable, but only for user accounts that are used for interactive sessions.

Summary

If you need to generate Active Directory login audit reports, the best approach is probably to aggregate your domain controller event logs and process them. While event logs are incredibly noisy, they're also incredibly reliable and provide historical information that Active Directory cannot.

If that's not feasible, use LastLogonDate. Or, even better, use the Search-ADAccount cmdlet baked into the Active Directory PowerShell module to get necessary information and output it to CSV file:

```
Search-ADAccount -AccountInactive -DateTime ((Get-Date).AddDays(-30)) -UsersOnly |  
Select Name,LastLogonDate,DistinguishedName| Export-CSV c:psinactive_users.csv
```

The Search-ADAccount cmdlet actually leverages LastLogonDate behind the scenes. Its inactivity period defaults to 15 days, which should be fine in most environments. The example above includes the syntax necessary to override the inactivity period with a value of 30 days. For those who prefer a systematized approach, there are helpful free tools such as Netwrix Inactive User Tracker to retrieve this information without having to convert values and analyze csv files.

How can Netwrix help?

Regularly reviewing each user's last logon date in Active Directory can help your domain admin detect and remove stale accounts that adversaries would love to compromise and abuse. But that's only one small part of a comprehensive cybersecurity strategy.

The Netwrix Active Directory Security Solution provides detailed information about not just the last logon time for every Active Directory user account, but all activity in Active Directory. Its comprehensive pre-built reports streamline logon monitoring. In particular, the User Accounts – Last Logon Time report lists all user accounts, both enabled and disabled, with the last logon time for each account. Using the report subscription function, IT admins can have the report delivered by email automatically on the schedule they specify, facilitating regular review in accordance with best practices and enabling them to eliminate system vulnerabilities more efficiently.

FAQ

What is difference between Lastlogon and lastLogonTimeStamp?

Unlike the Last-Logon attribute, the Last-Logon-Timestamp attribute is a replicated attribute; its value for any specific user is synced to every domain controller.

How is lastLogonTimeStamp calculated?

Last-Logon-Timestamp (or lastLogonTimeStamp) is registered by DC on user login; however, this attribute is not always updated across all DCs when one domain controller successfully processes a logon request.

What is interactive authentication?

Interactive authentication is a process in which a user is prompted to enter their user ID and password to log in to a device. The most common places where interactive login happens are the Windows login screen and the Windows lock screen.

Joe Dibley

Security Researcher at Netwrix and member of the Netwrix Security Research Team. Joe is an expert in Active Directory, Windows, and a wide variety of enterprise software platforms and technologies, Joe researches new security risks, complex attack techniques, and associated mitigations and detections.

