Attackers Can Now Use Mimikatz to Implant Skeleton Key on Domain Controllers & BackDoor Your Active Directory Forest

By Sean Metcalf

Published: 2015-01-19 · Archived: 2026-04-06 01:09:29 UTC

Jan 19 2015

- By Sean Metcalf in Microsoft Security, Technical Reference

Once an attacker has gained Domain Admin rights to your Active Directory environment, there are several methods for keeping privileged access. Skeleton Key is an ideal persistence method for the modern attacker. More information on Skeleton Key is in my earlier post.

Note that the behavior documented in this post was observed in a lab environment using the version of Mimikatz shown in the screenshot. There are likely differences in the Skeleton Key malware documented by Dell SecureWorks and the Mimikatz skeleton key functionality. Mimikatz effectively “patches” LSASS to enable use of a master password with any valid domain user. Rebooting the DC refreshes the memory which removes the “patch”.

Implanting the Mimikatz Skeleton Key on one or multiple Domain Controllers:

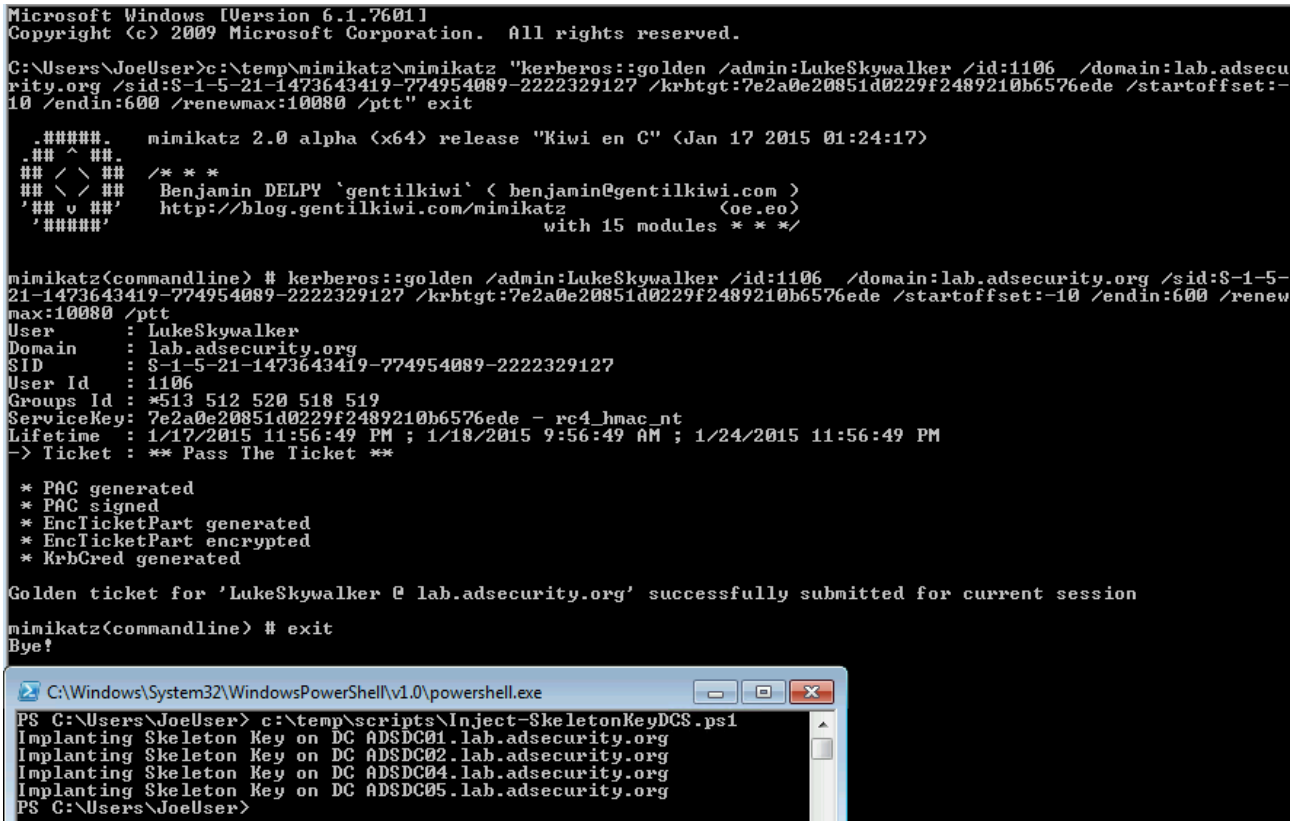
Mimikatz can now inject a skeleton key into LSASS on the Domain Controller by running the following command on the DC:

> mimikatz.exe “privilege::debug” “misc::skeleton” exit

When there are multiple Domain Controllers in an Active Directory site, all of them need the Skeleton Key implant to ensure the skeleton key master password is accepted as the user's valid password.. Since the client discovers a Domain Controller using DCLocator, the DC the client selects is effectively random. If all the DCs don't have skeleton key configured, the master password won't work when the client authenticates to a DC without skeleton key.

Scenario:

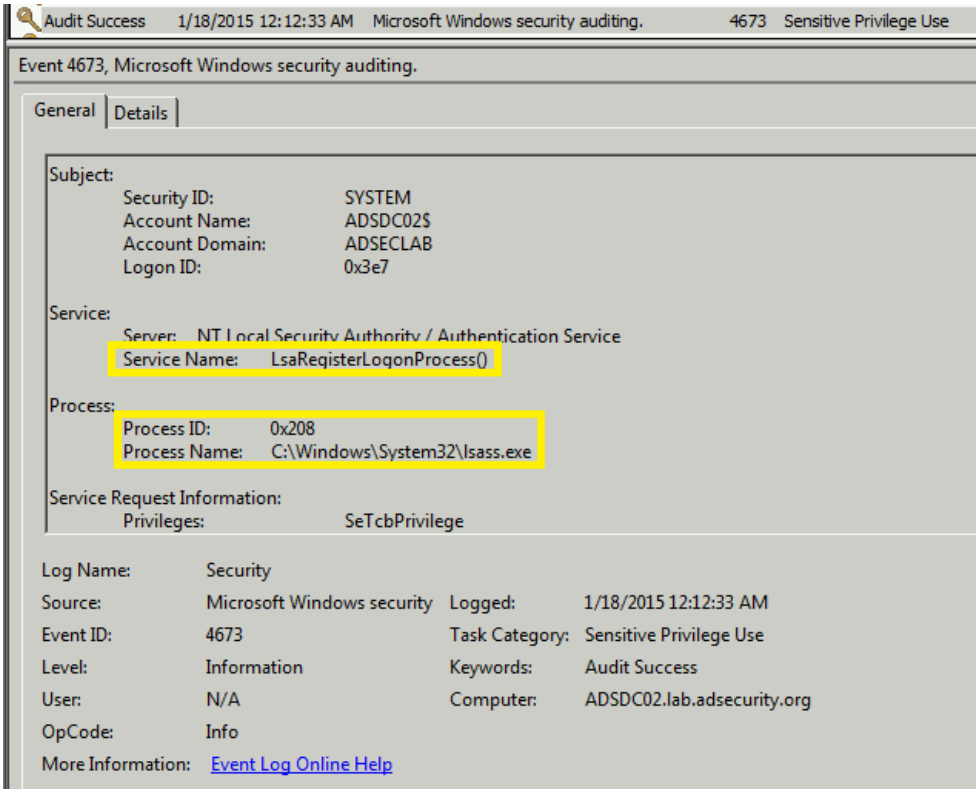
Either the attacker exploits MS14-068 or has the KRBTGT NTLM password hash and uses it to generate a Kerberos Golden Ticket to impersonate a valid Domain Admin account. The attacker leverages the forged Kerberos TGT ticket to access the Domain Controllers via PowerShell remoting. PowerShell remoting runs over WinRM and provides a shell running on the remote computer (much like SSH). In this case, the attacker runs a PowerShell script that uses "invoke-command" to run the mimikatz command on the DCs.



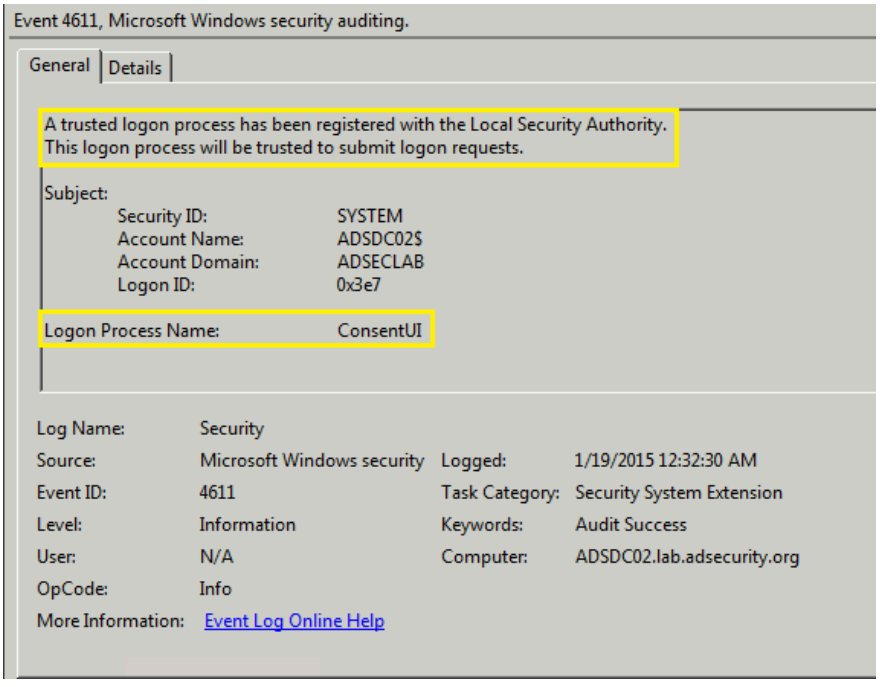
Domain Controller Security Events When Implanting the Mimikatz Skeleton Key:

When implanting the skeleton key remotely using Mimikatz the following events are logged on the Domain Controller.

Event Id 4673 Sensitive Privilege Use,

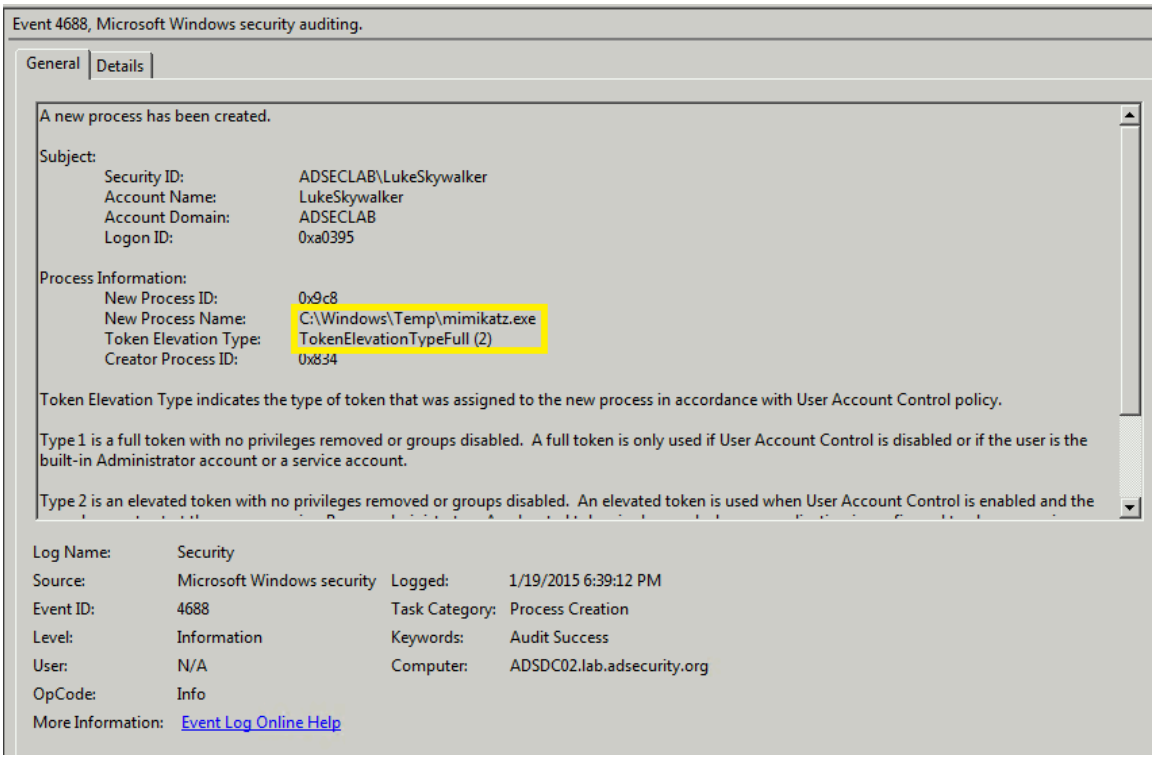


Event 4611: A trusted logon process has been registered with the Local Security Authority.

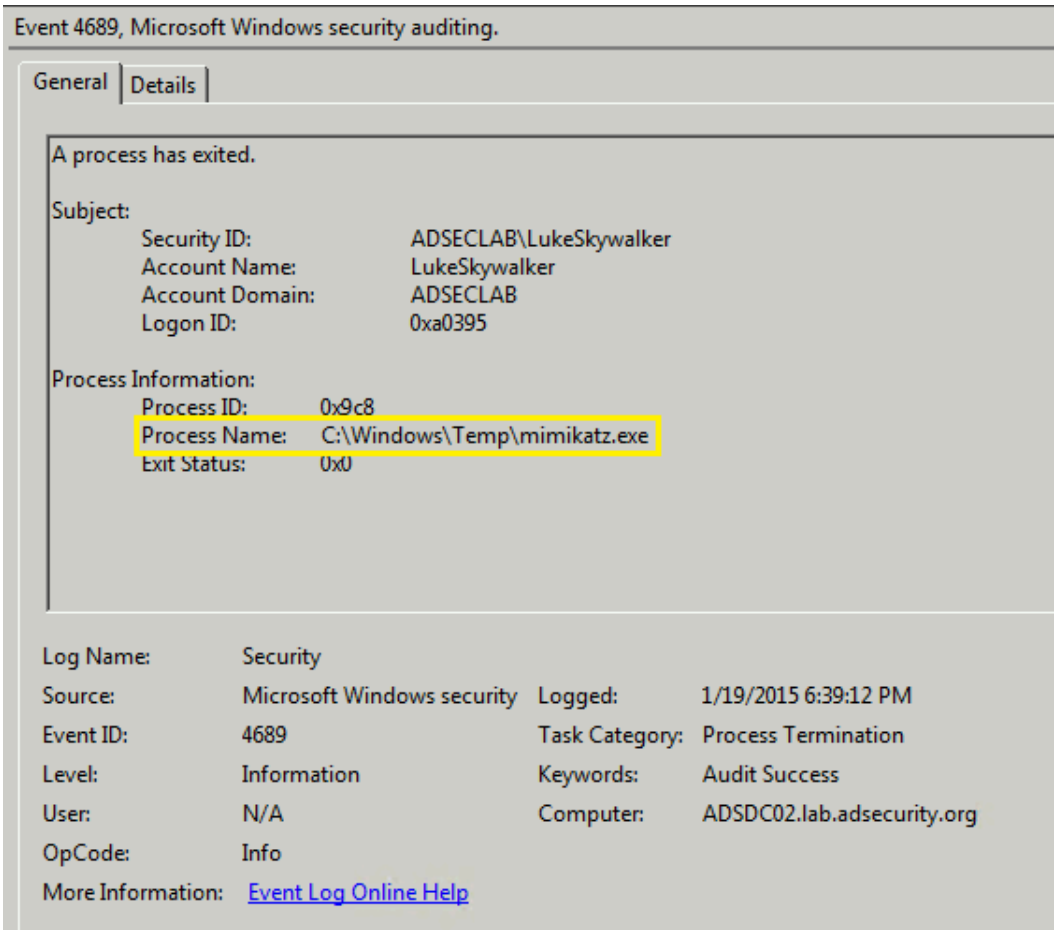


If Process Tracking (logging) is enabled, there are two events that are logged reliably.

Event 4688: A new process has been created.



Event 4689: A new process has exited.



Authenticating with the Mimikatz Skeleton Key:

Testing user password and user account with skeleton key password.
Note that both passwords are accepted – the valid user password and the skeleton key master password!

Testing Domain Admin account with password & skeleton key password.
Note that both passwords are accepted – the valid user password and the skeleton key master password!

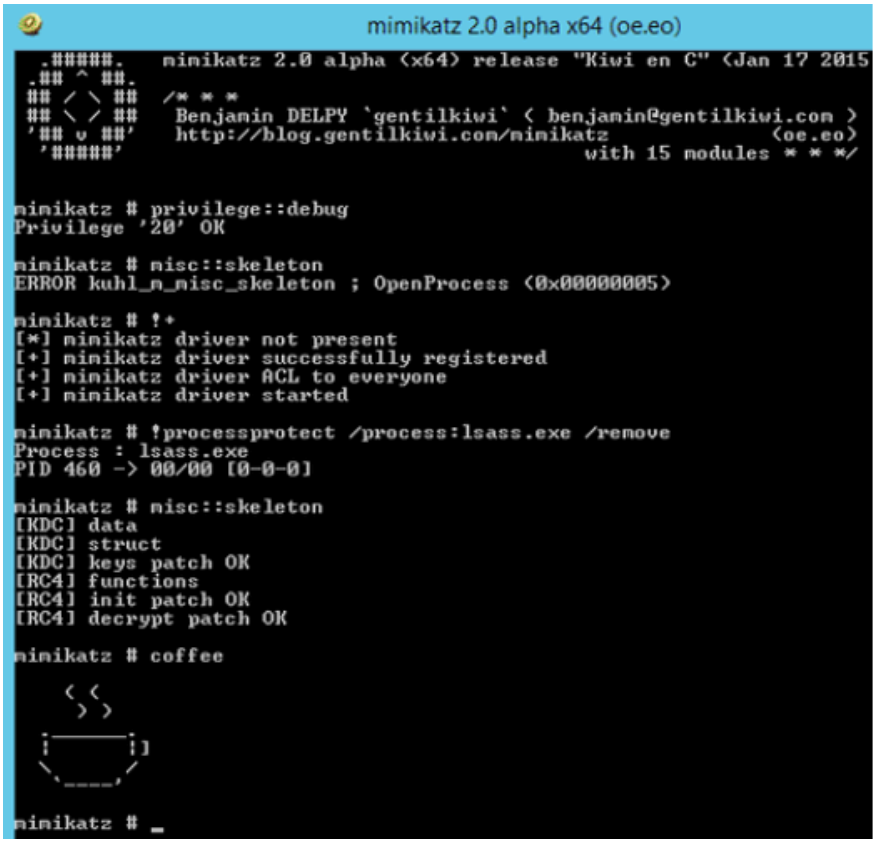
Skeleton Key Mitigation:

- Protect domain-level admin (DLA) accounts (Domain Admin, Administrators, etc) which reduces the risk of attackers gaining access to these credentials. Don't let DLA accounts logon to systems at a different security level from Domain Controllers. Don't let services run as Domain Admin on member servers that aren't protected at the same level as DCs.
- Enable smart card authentication for all users.
- Ensure Domain Controllers have limited connectivity to the network until MS14-068 is patched (kb3011780). The challenge is that the patch has to be applied after DCPromo is complete.
- Security software that prevents LSASS patching may mitigate the issue.
- Application whitelisting (ex. AppLocker) can prevent unapproved applications from running on Domain Controllers.
- Enabling Process Logging on Domain Controllers provides additional data on what applications (exes) are executed on Domain Controllers.
- Enable LSASS as a protected process on Windows Server 2012 R2 (Mimikatz can bypass with a driver, but that should make some noise in the event logs):

> The LSA, which includes the Local Security Authority Server Service (LSASS) process, validates users for local and remote sign-ins and enforces local security policies. The Windows 8.1 operating system provides additional protection for the LSA to prevent reading memory and code injection by non-protected processes. This provides added security for the credentials that the LSA stores and manages.

> **To enable LSA protection on a single computer**
>
> 1. Open the Registry Editor (RegEdit.exe), and navigate to the registry key that is located at: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa.
> 2. Set the value of the registry key to: “RunAsPPL”=dword:00000001.
> 3. Restart the computer.
>
> **To enable LSA protection using Group Policy**
>
> 1. Open the Group Policy Management Console (GPMC).
> 2. Create a new GPO that is linked at the domain level or that is linked to the organizational unit that contains your computer accounts. Or you can select a GPO that is already deployed.
> 3. Right-click the GPO, and then click **Edit** to open the Group Policy Management Editor.
> 4. Expand **Computer Configuration**, expand **Preferences**, and then expand **Windows Settings**.
> 5. Right-click **Registry**, point to **New**, and then click **Registry Item**. The **New Registry Properties** dialog box appears.
> 6. In the **Hive** list, click **HKEY_LOCAL_MACHINE**.
> 7. In the **Key Path** list, browse to **SYSTEM\CurrentControlSet\Control\Lsa**.
> 8. In the **Value name** box, type **RunAsPPL**.
> 9. In the **Value type** box, click the **REG_DWORD**.
> 10. In the **Value data** box, type **00000001**.
> 11. Click **OK**.

Mimikatz bypassing LSA Protection:



(Visited 22,640 times, 1 visits today)

Sean Metcalf

I improve security for enterprises around the world working for TrustedSec & I am @PyroTek3 on Twitter.
Read the About page (top left) for information about me. :)
https://adsecurity.org/?page_id=8

Source: http://adsecurity.org/?p=1275