BloodHound 1.3 – The ACL Attack Path Update (https://wald0.com/?p=112)

MAY 15, 2017 / 8 COMMENTS

Search this site

Recent Posts

A Red Teamer's Guide to GPOs and OUs (https://wald0.com/?p=179)

BloodHound 1.3 – The ACL Attack Path Update (https://wald0.com/?p=112)

Introducing BloodHound (https://wald0.com/?p=68)

Automated Derivative Administrator Search (https://wald0.com/?p=14)

Recent Comments

Intro & Background

In 2014, Emmanuel Gras and Lucas Bouillot presented their work titled "Chemins de contrôle en environement Active Directory (https://www.sstic.org/2014/presentation/chemins_de_controle_active_directory/)" ("Active Directory Control Paths") at the Symposium sur la sécurité des technologies de l'information et des communications (Symposium on Information and Communications Technology Security), where they used graph theory and Active Directory object permissions to answer the question, "Who can become Domain Admin?" I highly recommend checking out their presentation and whitepaper,

Scanning for Active Directory Privileges & Privileged Accounts - ^B.B^log (https://blog.bbsec.xyz/sec-news/755.html) on BloodHound 1.3 – The ACL Attack Path Update (https://wald0.com/?p=112#comment-4734)
Pgz81 on A Red Teamer's Guide to GPOs and OUs (https://wald0.com/?p=179#comment-3933)
0x90 on A Red Teamer's Guide to GPOs and OUs (https://wald0.com/?p=179#comment-3878)
A Red Teamer's Guide to GPOs and OUs – Information Security Outsider (http://blog.bitflow.studio/red-team/a-red-teamers-guide-to-gpos-and-ous/) on BloodHound 1.3 – The ACL Attack Path Update (https://wald0.com/?p=112#comment-1570)
A Red Teamerâ€™s Guide to GPOs and OUs - Secure Signal NYC (https://securesignal.nyc/wp/blog/2018/04/04/a-red-teameraes-guide-to-gpos-and-ous/) on A Red Teamer's Guide to GPOs and OUs

which we drew initial inspiration from for the BloodHound project, and received very helpful and specific information from for our adding object control paths to the BloodHound attack graph.

Rohan Vazarkar (@CptJesus (https://twitter.com/cptjesus)), Will Schroeder (@harmj0y (https://twitter.com/harmj0y)) and I are very proud to announce BloodHound 1.3, which introduces several new edge types based on Active Directory object control. Additionally, Will and Lee (@tifkin_ (https://twitter.com/tifkin_)) have put considerable work into developing corresponding PowerShell cmdlets which enable a pentester or red teamer to take advantage of these new edges. We believe that ACL-based attack paths will exploit an untapped attack landscape in Active Directory domains.

What are ACLs?

When we talk about ACL-based attacks, we are specifically referring to Access Control Entries (ACEs) which populate Discretionary Access Control Lists (DACLs). DACLs reside within security descriptors, which reside within securable objects. For a list of common securable objects, see https://msdn.microsoft.com/en-us/library/windows/desktop/aa379557(v=vs.85).aspx (https://msdn.microsoft.com/en-us/library/windows/desktop/aa379557(v=vs.85).as

(https://wald0.com/?p=179#comment-1564)

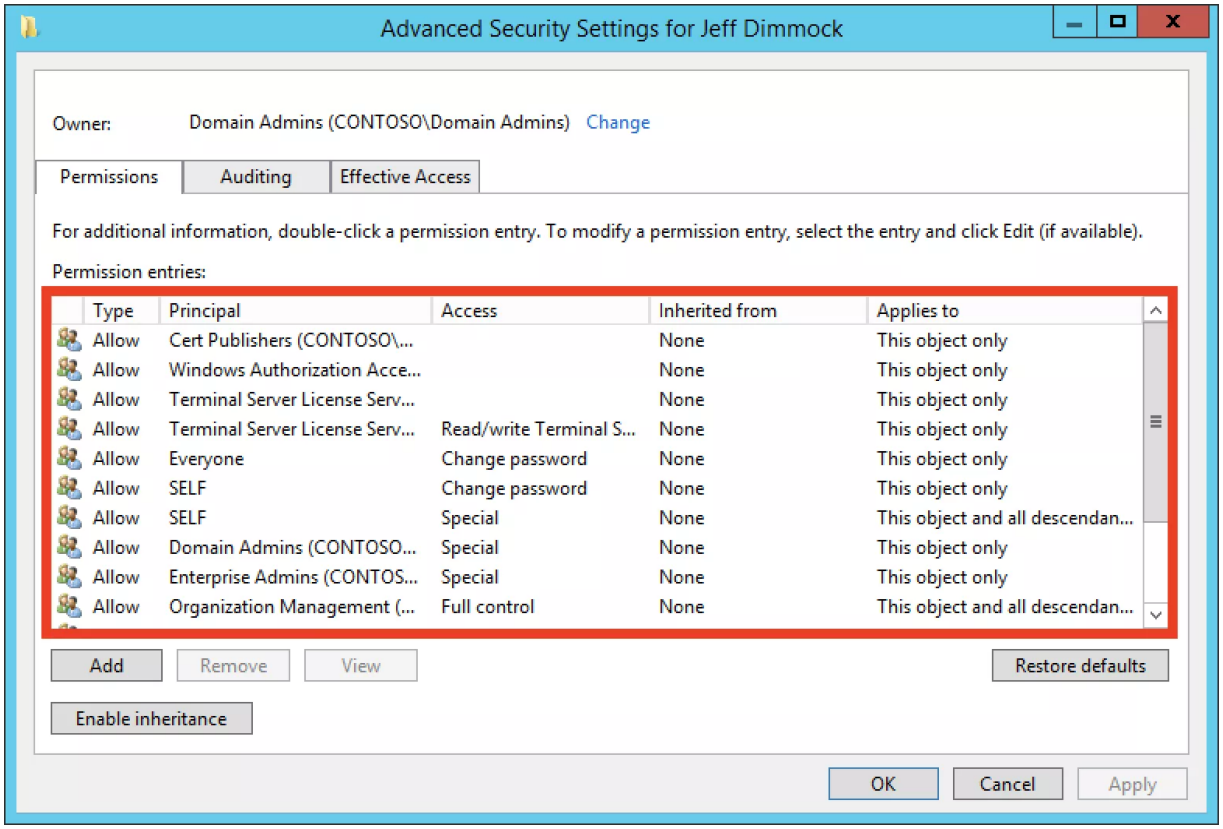
Archives

April 2018 (https://wald0.com/?m=201804)
May 2017 (https://wald0.com/?m=201705)
August 2016 (https://wald0.com/?m=201608)
February 2016 (https://wald0.com/?m=201602)

Categories

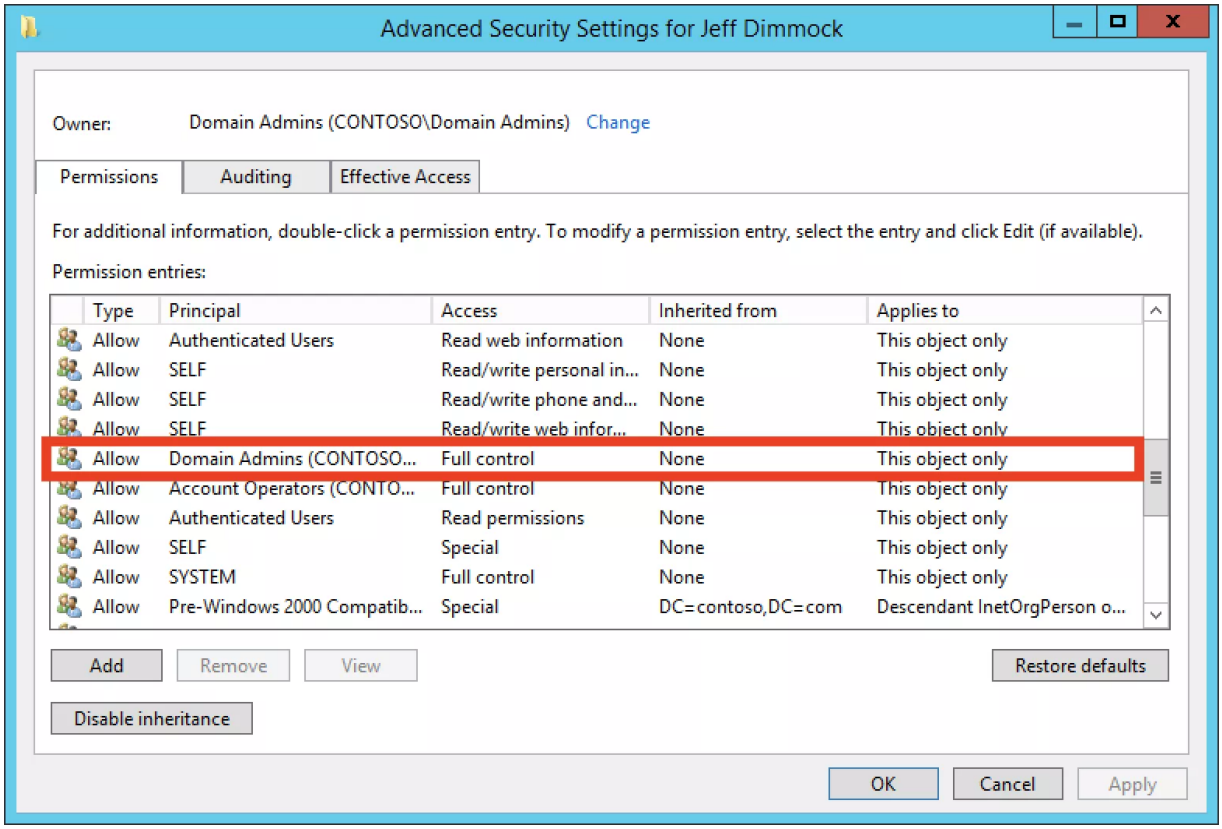
ACL (https://wald0.com/?cat=7)
Active Directory (https://wald0.com/?cat=4)
BloodHound (https://wald0.com/?cat=6)
GPO (https://wald0.com/?cat=8)
Graph Theory (https://wald0.com/?cat=5)
Group Policy (https://wald0.com/?cat=9)
Powershell (https://wald0.com/?cat=3)
Uncategorized (https://wald0.com/?cat=1)

px). Notably, Active Directory users, groups, and computers are securable objects. Access Control Entries describe the allowed and denied permissions for other principals in Active Directory against the securable object.



(https://i0.wp.com/wald0.com/wp-content/uploads/2017/05/JeffDimmockSecurityDescriptor.png?ssl=1)*Above: The graphical representation of the security descriptor for the user "Jeff Dimmock". Highlighted in red is the Discretionary Access Control List (DACL), comprised of Access Control Entries (ACEs).*

The best example of this is when one object has "full control" over another object. Consider the "Domain Admins" group, for example. It makes sense that the "Domain Admins" group would have full control over every other object in a domain:



(https://i1.wp.com/wald0.com/wp-content/uploads/2017/05/DomainAdminsFullControlACE.png?ssl=1)*Above: The ACE granting the "Domain Admins" group* full control *of the "Jeff Dimmock" user is highlighted in red.*

Now, of course the Domain Admins group has full control of every other object in Active Directory; however, as attackers, we are interested in how we can abuse ACEs to gain control of a domain admin or a user or group that gets us closer to our target objective. Additionally, the owner of an object has complete control (GenericAll equivalent) of the object, regardless of any explicit deny ACEs.

Abusable ACEs

This update adds seven new edges to the BloodHound attack graph schema, based on direct object-to-object control situations that we have verified are abusable. Additionally, Will Schroeder (@harmj0y (https://twitter.com/harmj0y)) and Lee Christensen (@tifkin_ (https://twitter.com/tifkin_)) have put considerable effort into creating easy-to-use PowerShell cmdlets to abuse each associated ACE:

- **ForceChangePassword:** The ability to change the target user's password without knowing the current value. Abused with Set-DomainUserPassword (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L4967).
- **AddMembers:** The ability to add arbitrary users, groups or computers to the target group. Abused with Add-DomainGroupMember (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L9893).
- **GenericAll:** Full object control, including the ability to add other principals to a group, change a user password without knowing its current value, register an SPN with a user object, etc. Abused with Set-DomainUserPassword (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L4967) or Add-DomainGroupMember (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L9893).
- **GenericWrite:** The ability to update any non-protected target object parameter value. For example, update the "scriptPath" parameter value on a target user object to cause that user to run your specified executable/commands the next time that user logs on. Abused with Set-

DomainObject (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L6122).

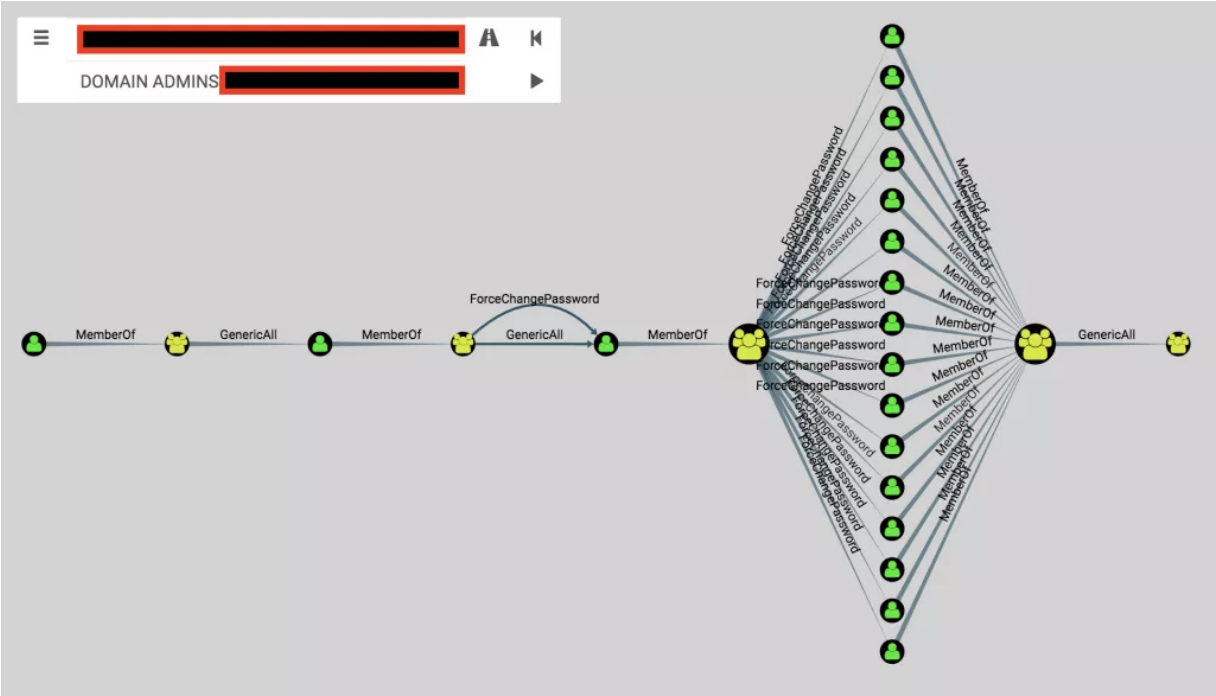
- **WriteOwner:** The ability to update the owner of the target object. Once the object owner has been changed to a principal the attacker controls, the attacker may manipulate the object any way they see fit. Abused with Set-DomainObjectOwner (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L6727).

- **WriteDACL:** The ability to write a new ACE to the target object's DACL. For example, an attacker may write a new ACE to the target object DACL giving the attacker "full control" of the target object. Abused with Add-NewADObjectAccessControlEntry (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L6496).

- **AllExtendedRights:** The ability to perform any action associated with extended Active Directory rights against the object. For example, adding principals to a group and force changing a target user's password are both examples of extended rights. Abused with Set-DomainUserPassword (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L4967) or Add-DomainGroupMember (https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1#L9893).

Attack Path Planning with BloodHound

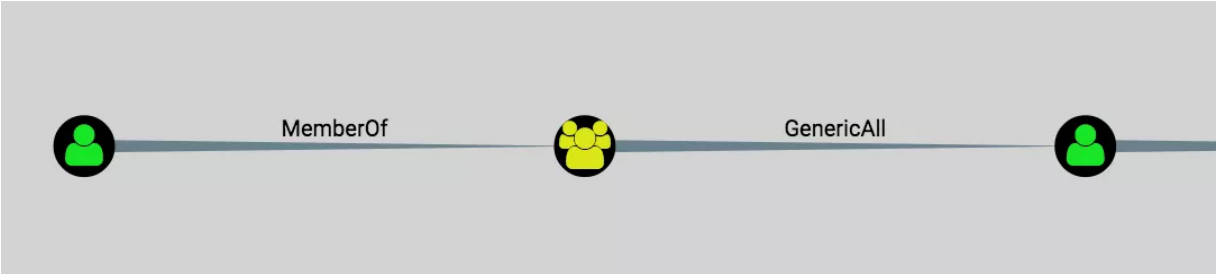
After completing BloodHound data collection activities (read: by default, all authenticated users can read all ACEs on all objects!), we can use the BloodHound interface to plan an attack to compromise our target. Let's take a look at an example based on real data from a real environment:



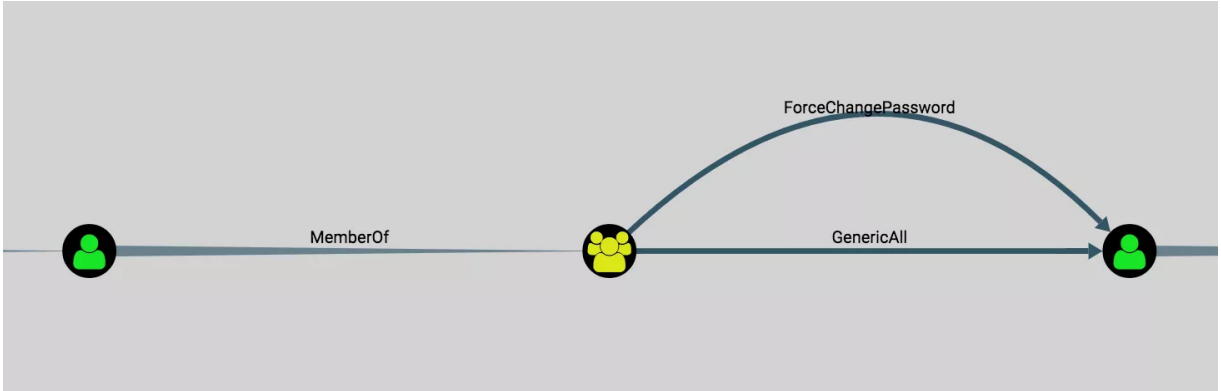
(https://i0.wp.com/wald0.com/wp-content/uploads/2017/05/FullACLAttackPath.png?ssl=1)*Above: An ACL attack path identified by BloodHound, where the target group is the "Domain Admins" group.*

In this instance, we have a relatively low-privileged user on the far left with an ACL-only attack path ending up in control of the Domain Admins group. Unfortunately, from an OPSEC perspective, we are forced to perform a password reset against one of the many users in the 7th step of the attack path. Along the way, we may choose to perform password resets on the two other users we identified; however, we have other options up our sleeve as

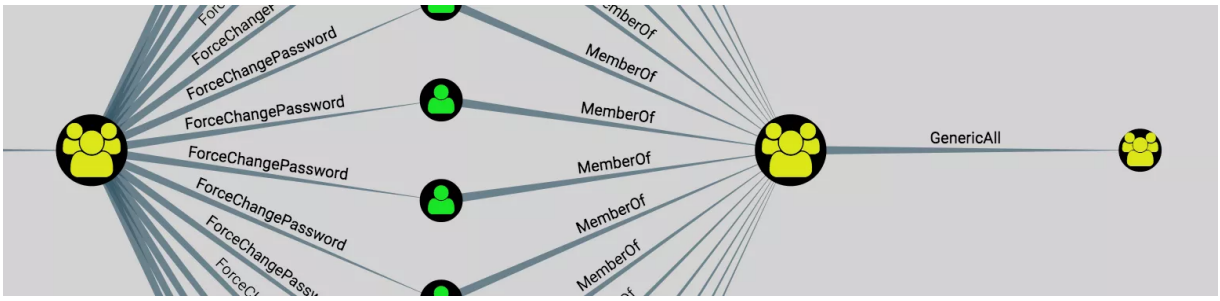
well, including altering the user's scriptPath attribute or registering an SPN with the target user, as Will (@harmj0y (https://twitter.com/harmj0y)) outlines in his blog post, "Targeted Kerberoasting (http://www.harmj0y.net/blog/activedirectory/targeted-kerberoasting/)":



(https://i2.wp.com/wald0.com/wp-content/uploads/2017/05/AttackPathStep1.png?ssl=1)*Above: Detail of step 1 of our attack path. The user on the left is a member of the security group in the center. That group has full control of the user on the right; therefore, so does the user on the left.*



(https://i0.wp.com/wald0.com/wp-content/uploads/2017/05/AttackPathStep2.png?ssl=1)*Above: Detail of the second step in the attack path. The user on the left belongs to the group in the middle. That group has both full control via "GenericAll", and (redundantly) the "ForceChangePassword" right against the user on the right.*

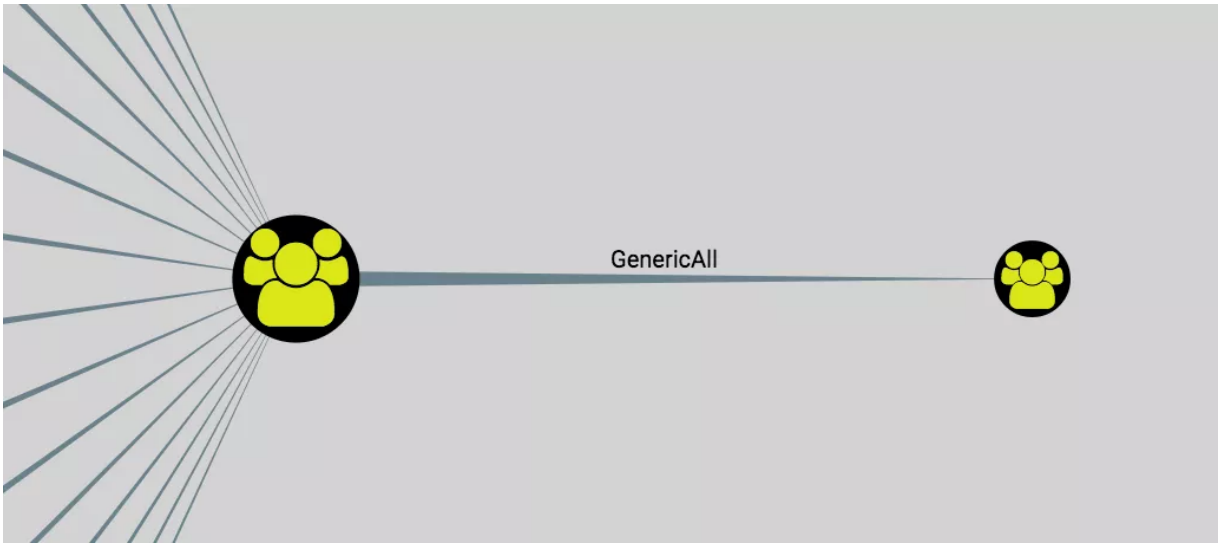


(https://i2.wp.com/wald0.com/wp-content/uploads/2017/05/AttackPathStep3and4.png?ssl=1)*Above: Detail of the final two steps in the attack path. The group on the left has "ForceChangePassword" right against several users who all belong to the group in the middle right. That group in the middle right has full control of the group on the right, which is the "Domain Admins" group.*

The second-to-last step of our attack path requires us to change an active user's password. The OPSEC considerations of this action should not be taken lightly: if an attacker changes a service account password, for example, and the associated actions of that service account start to fail, the SOC may be alerted. Or, if we change the password of an admin user and that user can't log on next time, they may suspect that their password was changed by someone else. A great solution to this would be the ability to directly inject NT hashes into the NTDS after we get DA, and reset the user's password to what it was before; however, I haven't found a way to do this (@gentilkiwi (https://twitter.com/gentilkiwi) please!). This will require that you retrieve the target user's NTLM hash or plaintext password at some point in the attack chain, so if you can't do that or you can't escalate to domain admin and fetch the user's NT hash from their password history in NTDS, you're in the dog house.

Another option would be to change that user's password, pivot to a machine that user is currently actively logged onto, grab the clear-text password that user used to authenticate to that machine with mimikatz, and reset the user's password to that value. If done quickly enough (and if password changes are not closely audited), the user should be none-the-wiser. This is where domain recon is critical. The more you know about the environment, user behaviors, monitoring capability, etc., the more likely you are to be able to execute this step of the attack path without getting caught.

In the last step of the attack path, we gain full control over our target node, the "Domain Admins" group:



(https://i0.wp.com/wald0.com/wp-content/uploads/2017/05/AttackPathStep4.png?ssl=1)

Note that having full control over a group does *not* automatically give you any control over users joined to that group. In this instance, the option we have is simple: add an arbitrary principal that we control to the domain admins group. Once the attacker has done this, they may DCSync the krbtgt hash, remove their arbitrary principal from the domain admins group, and

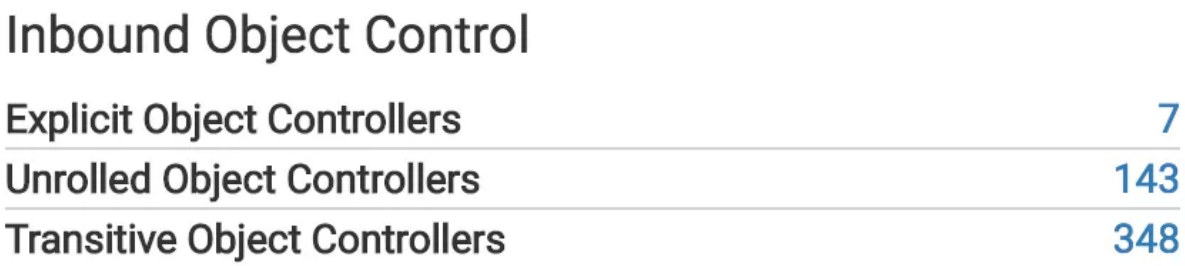
then set up their persistence using the krbtgt hash. For more information about that, see Sean Metcalf's (@PyroTek3 (https://twitter.com/PyroTek3)) blog post, "Kerberos & KRBTGT: Active Directory's Domain Kerberos Service Account (https://adsecurity.org/?p=483)."

For a bit of fun, here's a video showing the YOLO method of executing this attack path. Because we're just manipulating AD objects via LDAP/ADSI, we can actually execute these attack paths with considerable speed:

Auditing ACLs with BloodHound

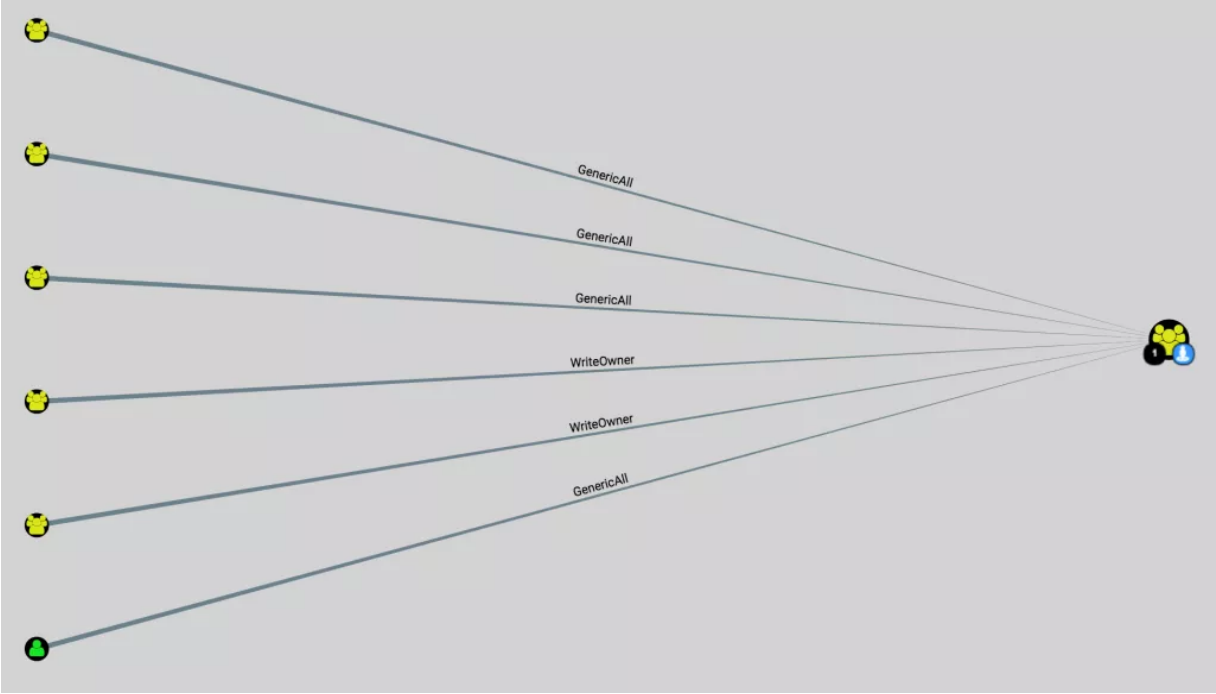
Effectively auditing ACLs in Active Directory has historically been a confusing, frustrating, and painfully slow process. BloodHound now enables quick, easy auditing of ACLs, with two important caveats: first, the only ACLs we collect information on are those that can be used to take control of another object, and we still have some work to do on including OUs, GPOs, and other attacks (https://github.com/sensepost/ruler) which can benefit from misconfigured ACLs to the graph schema. Second, we only ingest "Allow" type ACEs, and do not account for effective access as determined by how the security reference monitor reads ACEs in canonical order; however, in most environments, we have noticed very little usage of "Deny" type ACEs.

By clicking on a group node, for example, and scrolling to the bottom of the group info tab, we see the "**Inbound Object Control**" section:



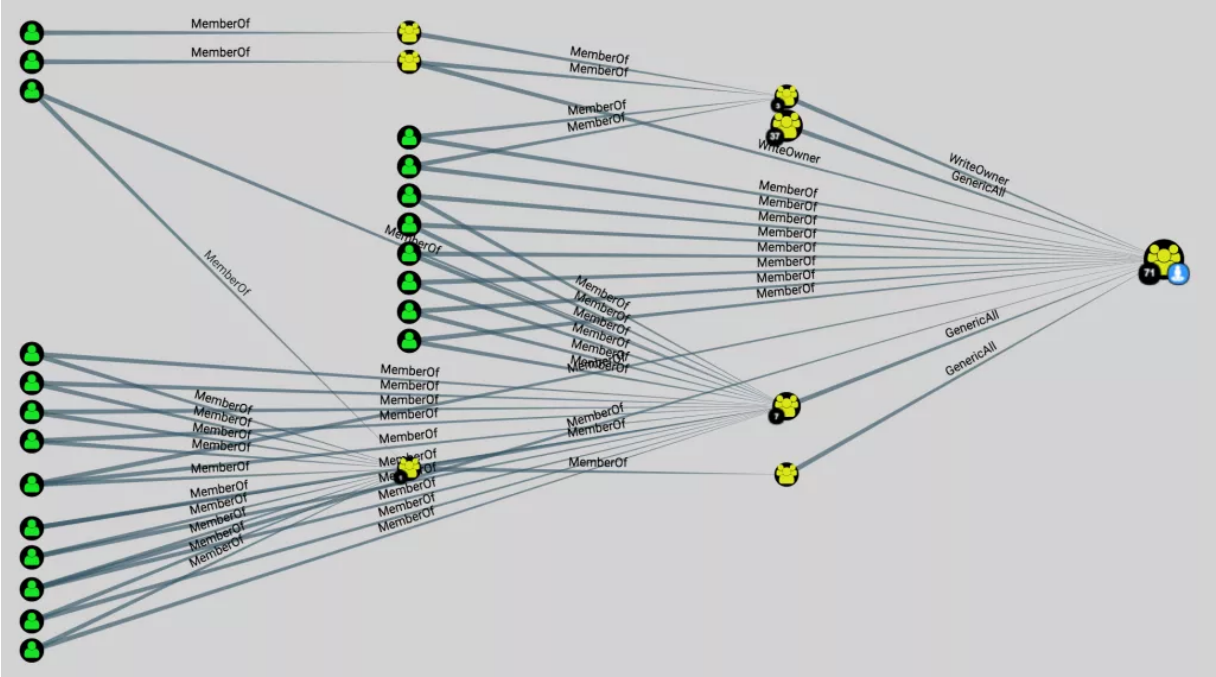
(https://i2.wp.com/wald0.com/wp-content/uploads/2017/05/InboundControl.png?ssl=1)

"**Explicit Object Controllers**" tells us who the first degree controllers of this object are. Note that this is different than non-inherited rights, and may include ACEs which are inherited from parent objects. By click on the number, we can see what those objects are:



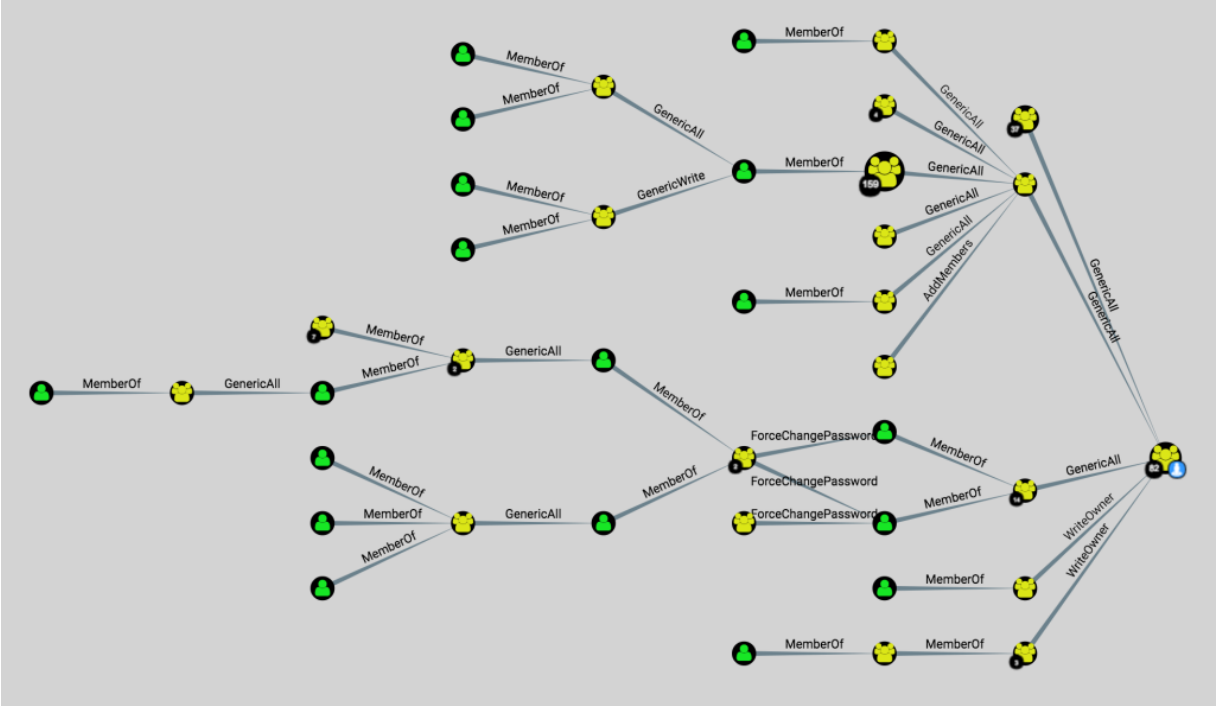
(https://i2.wp.com/wald0.com/wp-content/uploads/2017/05/ExplicitControllers.png?ssl=1)

"**Unrolled Object Controllers**" takes every group with privileges against this object and unrolls them out, showing the effective principals who have that right via security group delegation:



(https://i2.wp.com/wald0.com/wp-content/uploads/2017/05/GroupDelegatedControllers.png?ssl=1)

Finally, "**Transitive Object Controllers**" draws out all the possible attack paths based on the collected ACL data. If there is an ACL-only attack path to compromise this object, BloodHound will find it for you:



(https://i0.wp.com/wald0.com/wp-content/uploads/2017/05/TransitiveControllers.png?ssl=1)

These different views will give you instant insights on what other objects in AD have the ability to gain control of any other node.

Future Work and Conclusion

Soon, we will update the schema to reflect GPO edit rights as they apply to OUs and the children of those OUs. We are also continually researching methods of controlling objects in AD without setting off alarms. Additionally, Will and I will be speaking at Black Hat USA 2017 (https://www.blackhat.com/us-17/briefings.html#an-ace-up-the-sleeve-designing-active-directory-dacl-backdoors) on designing very sneaky backdoors in AD using ACLs, so expect that to come out at the time of that talk.

BloodHound is available free and open source on GitHub at https://github.com/BloodHoundAD/BloodHound (https://github.com/BloodHoundAD/BloodHound)

You can join us on Slack at the official BloodHound Gang Slack by clicking here: https://bloodhoundgang.herokuapp.com/ (https://bloodhoundgang.herokuapp.com/)

You can find the deck we used to present the BloodHound 1.3 ACL Attack Path update at Paranoia17 here: https://www.slideshare.net/AndyRobbins3/bloodhound-13-the-acl-attack-path-update-paranoia17-oslo (https://www.slideshare.net/AndyRobbins3/bloodhound-13-the-acl-attack-path-update-paranoia17-oslo)

Also published on Medium (https://medium.com/@_wald0/bloodhound-1-3-the-acl-attack-path-update-74aa56c5eb3a).

Share this:

Related

Introducing BloodHound (https://wald0.com/?p=68)
August 29, 2016
In "Active Directory"

A Red Teamer's Guide to GPOs and OUs (https://wald0.com/?p=179)
April 2, 2018
In "ACL"

Automated Derivative Administrator Search (https://wald0.com/?p=14)
February 15, 2016
In "Active Directory"

Older Post (https://wald0.com/?p=68)

Next Post (https://wald0.com/?p=179)

8 thoughts on "BloodHound 1.3 – The ACL Attack Path Update"

1. Attacking Active Directory Permissions with BloodHound | The Insider Threat Security Blog (https://blog.stealthbits.com/attacking-active-directory-permissions-with-bloodhound/)
   June 13, 2017
   Reply (https://wald0.com/?p=112&replytocom=513#respond)

   […] These are the rights that let a user take over another account, or add themselves to a group, to increase their access rights. If you tie many of these rights together, there may be the ability to move from an account with no privileges to an account with Domain Admin rights. BloodHound makes it extremely easy to explore these attack paths. For a more complete overview of the supported permissions by one of the authors of BloodHound, you can read more here. […]

2.

Scanning for Active Directory Privileges & Privileged Accounts » Active Directory Security (https://adsecurity.org/?p=3658)
June 15, 2017
Reply (https://wald0.com/?p=112&replytocom=517#respond)

[…] Andy Robbin's (@_Wald0) post covers ways these rights can be abused. […]

3.

Active Directory Access Control List – Attacks and Defense – Enterprise Mobility and Security Blog (https://blogs.technet.microsoft.com/enterprisemobility/2017/09/18/active-directory-access-control-list-attacks-and-defense/)
September 18, 2017
Reply (https://wald0.com/?p=112&replytocom=649#respond)

[…] BloodHound 1.3 – The ACL Attack Path […]

Andreas Kasidis
October 1, 2017
Reply (https://wald0.com/?p=112&replytocom=680#respond)

What is the procedure to manually check to which user does the forcechangepassword acl apply?

I am doing a get-adobjectacl against a specific user and I see it has the forcechangepassword ACE. How can I check to which user does this ACE apply?

5.

Active Directory Access Control List – Attacks and Defense – Enterprise Mobility + Security (https://cloudblogs.microsoft.com/enterprisemobility/2017/09/18/active-directory-access-control-list-attacks-and-defense/)
October 18, 2017
Reply (https://wald0.com/?p=112&replytocom=708#respond)

[…] ACL's to the front (literally…) with announcing BloodHound 1.3. Though the example in their blog post (on how to plan an ACL-only attack path ending as a domain admin) may prove difficult in practice, […]

6.

A Red Teamer's Guide to GPOs and OUs – wald0.com (https://wald0.com/?p=179)
April 2, 2018
Reply (https://wald0.com/?p=112&replytocom=1540#respond)

[…] BloodHound 1.3 – The ACL Attack Path Update […]

7.

A Red Teamer's Guide to GPOs and OUs – Information Security Outsider (http://blog.bitflow.studio/red-team/a-red-teamers-guide-to-gpos-and-ous/)
April 4, 2018
Reply (https://wald0.com/?p=112&replytocom=1570#respond)

[…] them. The initial release of BloodHound focused on the concept of derivative local admin, then BloodHound 1.3 introduced ACL-based attack paths. Now, with the release of BloodHound 1.5, pentesters and […]

8.

Scanning for Active Directory Privileges & Privileged Accounts - ^B.B^log (https://blog.bbsec.xyz/sec-news/755.html)
August 20, 2018
Reply (https://wald0.com/?p=112&replytocom=4734#respond)

[…] have had this post in draft for a while and with Bloodhound now supporting AD ACLs (nice work Will @harmj0y & Andy @_Wald0!), it's time to get more information out about AD […]

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.