Exploiting Relationship between Active Directory Objects

Boring security is smart security

@junedlive
Contents

• Red Team – Adversary simulation
• Why talk about Active Directory
• Active Directory Structure
• Active Directory Object Relationship
• Bloodhound Introduction
• Case studies!
SamAccountname: Juned.Ansari
company: DarkMatter
title: Senior Consultant
Department: Computer Network Exploitation
initials: AJ
Employeeid: 1337
Description: Hacks for Work, Fun and Learn!, 10+ years of Field Experience, Started as System Admin, Played roles in both Defense and Offense, Primary area of focus is Advance Attack Simulation, Authored two books on Kali Linux
Red Team Engagement vs Other Security Tests

- Vulnerability Assessment
  - Broad Scope, Breath over depth
  - Automated

- Penetration Test
  - Varied scope, Balancing act between depth and breath
  - Prioritized list of vulnerabilities

- Red Team Engagement
  - Goals
  - Measures impacts on an organization
Red Team Phases

- **Get In**: Gain access to network or System. This can be through a Compromised asset or through access granted as part of the scenario. Recon/Enumeration/Exploit

- **Stay In**: Establish persistence or permanent presence. Establish foothold in order to survive the duration of the engagement. Persistence/Lateral Movement/Continued Enumeration

- **Act**: Perform an operational impact, such as Exfiltrate data.
Hunting for Targets

• Lateral Movement
  Moving across client computers
  Site A → Site B

• Vertical Movement
  Moving across server layers
  User VLAN → Server VLAN

• Collectively known in the cyber space as lateral Movement
• Critical piece in the attack chain as local compromise spreads and becomes global
• Increasing difficult for the Incident Response team to scope the breach and perform appropriate containment and remediation.
The Cycle of Lateral movement

Active Directory Pawned

Applications Integrated with Domain
Sharepoint
Microsoft SQL
Microsoft Exchange
UNIX

Windows Dominated Infrastructure

Single Sign on

Multiple privileged accounts
Enterprises have new security solutions integrated in their networks.

- Security Information and Event Management
- Vulnerability scanners
- Next Generation Firewall
- Endpoint Detection and Response solutions
- Database Activity Monitors

*Response to every security issue is a new security solution!*
New security Solutions

Strengthening Active Directory
Reality about your AD infrastructure is...

• Its there since a long time and its working... so leave it that way...

• Organizations have moved from couple of workstations to 1000s of Servers.
Reality about your AD infrastructure is…

• Most Active Directory setups are rich with multiple paths to exploit.

• IT administration is mostly outsourced and contracted.

• Traditional methods of Administration used.
Active Directory Structure
Active Directory Structure

• In AD everything is an object.
  • Users, computers, domains, group, group policy, OU

• Every object has set of properties (attributes) permissions

• Domain Controller is heart of the AD
  Provides Authentication
  Stores all domain objects in a Database
  Replicates Domain data
Active Directory Objects – Attributes

• Each object has a set of attributes common across objects and attributes unique based on the functionality
AD Object Permissions

- Permissions on object can be assigned by OU Delegation
- Each object can have permissions set on it individually.
AD Object Permissions

- Granular permissions to read and write individual attributes can also be assigned
AD Object Permissions

- Domain administrators have permissions over all domain objects
- Enterprise Admins have rights over all domains in the forest
- IT Admins can assign permissions based on roles
  - Dev ops Team have permissions of over their projects
  - IT helpdesk have rights over user machines and objects
  - Assistants have to operate their managers mailboxes
- Nested Permissions is integral to Active Directory
• Nested Group is common in today’s enterprises.
• Biggest Domain Group by default is the Domain Users group.
• Includes every new account created in the Domain

• User A → Group B → Group C → Group D

• Uncareful Group nesting leads to excessive Privileges.
Privileged Group Access

- Most accounts in AD domain are over-permissioned
- Service Accounts in Domain Admin Groups
- Service Accounts having permissions over privileged Users
- Domain accounts as Local Admins
- Computer accounts in Admin Groups
- Domain Admin accounts used on user machines
- Nested groups access

An attacker always on the hunt for such configurations
Object Relationship Graphing
Object Relationship Graphing

- During an adversary simulation we need to map the targeted environment and visualize possible attack path
  - Which domain admins are logged in to which machines
  - Does current domain user has admin privilege on any other machine
  - Which users have change password rights and full Access on which objects

- Home many hops is the Domain Admin workstation
- Are there any service accounts that can help me jump to any database server
- and lots more...... depends on what you are target it.

- Tedious manual way to tracing each permission across the domain
  Powershell module of Active Directory
  Get-ACL, Get-Netsession, Get-GPO etc

OR... automate and map the path to your target into a graph
Manual Mapping of Access Control Entry

```
PS C:\Tools> Get-DomainObjectAcI -Identity "akhan" -Reso{}veGUIDs | wher{$_objectAcType -like "User-Force-Change-Password"}

AceQualifier : AccessAllowed
ObjectDN : CN=Armman Khan,OU=HR,DC=LabinDistress,DC=local
ActiveDirectoryRights : ExtendedRight
ObjectAceType : S-1-5-21-3844510068-2928011426-106315059-1129
ObjectSId : ContainerInherit
InheritanceFlags : 72
AceType : AccessAllowedObject
isCallBack : False
PropagationFlags : None
SecurityIdentifier : S-1-5-21-3844510068-2928011426-106315059-1128
AccessMask : 256
AuditFlags : None
AceFlags : True
InheritedObjectAceType : ContainerInherit, Inherited
InheritedObjectFlags : User
OpaqueLength : 0

PS C:\Tools> Get-DomainObject "S-1-5-21-3844510068-2928011426-106315059-1128"

userprincipalname : helpdesk@LabinDistress.local
countrycode : 0
displayname : Helpdesk
samaccounttype : USER_OBJECT
samaccountname : helpdesk
objectSId : S-1-5-21-3844510068-2928011426-106315059-1128
objectClass : {top, person, organizationalPerson, user}
codepage : 0
givenname : Helpdesk
cp : Helpdesk
primarygroupuid : 513
distinguishedname : CN=helpdesk,OU=Service-accounts,DC=LabinDistress,DC=local
name : helpdesk
objectguid : 6068a3c2-c818-4cf6-83bf-68d4147e6a35
objectCategory : CN=Person, CN=Schema, CN=Configuration, DC=LabinDistress, DC=local
```
Graph Theory

- Study of graphs, which are mathematical structures used to model pairwise relations between objects.

- A graph is made up of *vertices* (also called nodes or points) which are connected by *edges*.

- The most basic form of graph that we use and see every day is Google Maps.

- It’s just one big graph! Where Edges represent streets and *vertices represent crossings*. 
BloodHound to our rescue

• Javascript web application that uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment.

• Heavily relies on the PowerShell based Domain reconnaissance tools called Powerview by harmjoy

• Retrieves Domain Object and ACE’s data from Domain Controller using LDAP queries.

• Identifies session information and local Administrators from computers in the domain to find out currently logged on users.

• Massages the information and display the information in a graph format.

• Developed by @_wald0, @CptJesus, and @harmj0y.
Bloodhound terminology

- **Nodes →** Users, Computers, Groups and Domains
- **Edges →** Type of relationship between two AD objects.
  - MemberOf
  - HasSession
  - AdminTo
  - ACL – GenericAll
  - ACL – ForceChangePassword
  - ACL -- AddMember
- **Paths →** Series of Nodes connected by Edges which is the attack path. Each edge can be abused to reach the next node.
Group membership:

• Domain Security Group membership – Directly from Domain controller

• Group, user, domain and computer properties

• Local Admin and RDP Group membership – From each computer

• Session Information: Using the NetSessionEnum function against each computer

• Abusable ACEs from objects

Few more like GPO Links, DCOM, Domain trusts etc..
Bloodhound Data Collection

PS C:\Tools> Import-Module \SharpHound.ps1
PS C:\Tools>
PS C:\Tools> Invoke-BloodHound -CollectionMethod All
Initializing BloodHound at 9:37 PM on 4/2/2019
Resolved Collection Methods to Group, LocalAdmin, Session, Trusts, ACL, Container, RDP,
Starting Enumeration for LabinDistress.Local
Status: 91 objects enumerated (+91 Infinity/s --- Using 157 MB RAM )
Finished enumeration for LabinDistress.local in 00:00:00.6656977
4 hosts failed ping. 0 hosts timedout.

Compressing data to C:\Tools\20190402213752_BloodHound.zip.
You can upload this file directly to the UI.
Finished compressing files!
PS C:\Tools> _
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.

Bloodhound hunts for 7 different Abusable ACEs
• ForceChangePassword
• AddMembers
• GenericAll
• GenericWrite
• WriteOwner
• WriteDACL
• AllExtendedRights

By default, all authenticated users can read all ACEs on all objects!
Case Studies
From unprivileged user to a local admin on 2 systems in the Domain!
Get Started!

• Bloodhound GitHub Repository
  https://github.com/BloodHoundAD/BloodHound

• Ingestor aka Data collection script/tool
  https://github.com/BloodHoundAD/BloodHound/tree/master/Ingestors

• BloodHound User Interface
  https://github.com/BloodHoundAD/BloodHound/releases
NOW WE’RE COMPLIANT!

Boring security is Smart security
Questions