Pass-the-Hash II: Admin’s Revenge

Skip Duckwall & Chris Campbell
Do you know who I am?

• Skip
  • Co-presented PTH talk last year at BH, Derbycon
  • http://Passing-the-hash.blogspot.com
  • @passingthehash on twitter
  • Works for Accuvant Labs

• Chris
  • Co-presented PTH talk last year at BH, Derbycon
  • http://www.obscuresec.com
  • @obscuresec
  • Works for Crucial Security (Harris Corp)
Why we are here

• Dispel some FUD going around with PTH

• Provide some practical things everybody can do to defend against credential attacks as well as PTH

• And do it all in less that 80 pages!
The FUD stops here!

• We’re trying to educate everybody about the issues at hand
• Pass-the-hash sounds super sexy but is **NOT** the biggest problem the enterprise faces
• Windows has numerous issues with authentication in addition to PTH…
  • Credential exposure (mimikatz / WCE)
  • Broken protocols still in use (MSCHAPv2 / NTLMv1)
  • Cached credentials
  • Tokens, etc…
PTH-The biggest problem on the network?

- Actually, PTH is only a small subset of the problems with Windows authentication
- What about easy to recover plaintext passwords being kept in memory? Thanks Mimikatz!

Why use PTH when you can use the actual creds?!?
I can fix PTH... With a Patch!

- PTH is by design functionality. There is no fix, there is only mitigation or using some other form of auth.

- Why do you think that the MSV1_0 / NTLMSSP only saves the hash?
Kerberos Solves the PTH problem?

- NTLM hashes are used as the long-term secret keys
- KRBTGT hash is the master key for all Kerberos tickets
- Loss of this hash can completely undermine Kerberos
- Also, TGTs are portable, just like hashes
  - Move from one machine to another...
The Real Problem : Single Sign On

SSO – Ask for the password once, logon everywhere

Microsoft has a term for asking the user for their creds too many times: “Credential Fatigue”

Windows caches credentials in memory for all possible forms of authentication, even if they aren’t being used. Because you know, they *could* be used... sometime... somewhere... somehow... maybe?
Microsoft has a credential problem

Hello
My name is:

MICROSOFT
and I have a credential problem
Easy to recover Plaintext PW in Memory

• Thanks Benjamin Delpy for Mimikatz!
• Multiple SSPs save both username / pw for future use
  • Digest-MD5
  • LiveSSP
  • TSPKG
Even if nobody’s logged in ... 

- There’s at least one plaintext password for a domain in the LSA secrets.
- Computer’s domain account
  - Can be used to gather info from the domain
    - Usernames
    - Group memberships
  - Can be used to browse file shares
    - There aren’t any recoverable creds on any shares are there?
    - Group Policy Preferences... we’ll touch on this later...
- There might be more accounts
  - Creds for accounts with saved passwords (service accounts)
Don’t forget local account hashes

• Stored in the SAM (Security Account Manager)
• Local 500 account is a dangerous account
  • Has separate UAC settings from regular user which are **DISABLED** by default
  • Often enabled (despite being disabled by default)
  • Often has the same password across large number of machines
  • Has access to the domain via the computer’s domain account
For Brevity . . .

• We aren’t going to talk about Domain Cached Credentials
• Or token impersonation
• Or Services that store passwords in files
• Or plaintext password files on file servers
• Or keyloggers
• Or Phishing
• Or any of the other ways that an attacker can get creds
Bottom Line

• An attacker has multiple ways to gain access to valid (or usable) network credentials
• Attackers take the path of least resistance
• Once an attacker has SYSTEM on one box, it’s usually a matter of time until they have SYSTEM on your domain controller
How do you make PtH worse?

does Microsoft

- Ensure that every local admin password is the same
  - Introducing a feature that makes PtH easier to exploit:
    - Group Policy Preferences

- Make sure that hashes never change
  - Over-selling an expense as a security mitigation:
    - Smart Cards
clean up slide
awe, 7/11/2013
Group Policy Preference Settings

- Easy way to enforce settings on every workstation
- Popular with administrators to set passwords
GPP Passwords: Making it Easy
GPP: Its Even Worse

- Passwords are obfuscated on the domain controller
- Easily decrypted by anyone on the network
- Demo

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

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

PS C:\demo> Get-EncryptedCpassword "9KQYhHxSxrrZjFo8Frt/nExdMLKsQM+ThhWOJkajaRc" Recyling*3ftw!
PS C:\demo>
Smart Cards

• Who needs passwords?

• Authentication still works the same
Smart Cards = Persistence

- NT Hashes still look the same with smart cards

- NTLMSSP and SSO push PtH to the perimeter
MS Mitigation for PtH

- Microsoft PtH whitepaper has 3 main mitigations
  - Restrict and protect high privileged domain accounts
  - Restrict and protect local accounts with admin privs
  - Restrict inbound traffic using Windows Firewall
Our Mitigations

• Don’t let the attacker get SYSTEM
  • Most of the tools don’t work w/o SYSTEM privs on the workstation
  • Users almost never need admin access to their workstation (no matter how high up they are in the org)
  • ACL off unneeded command line utilities
    • CMD.EXE
    • NET.EXE
  • Patch all local privilege escalation bugs
    • Don’t always show up as ‘critical’ or ‘high’ in patch software
Don’t make it easy for them

• Use UAC, even for the local 500 account
• Don’t use GPP to set the passwords
• Disable the local 500 if you can get away with it
• Don’t use elevated creds in startup scripts
• Don’t save service account passwords on workstations
Protect the crown jewels

• The loss of your DC means the loss of your network
• Make sure your VM environment is safe (if your DC is virtualized)
• Encrypt backups
• Don’t store backups on shares that regular users can access
Too Many EA/DA accounts

• You probably don’t need 50 enterprise (or domain) admins

• These accounts are given out too freely because they easily can solve problems

• Focus on what your specific needs are
  • Lots of tools only need specific privileges rather than full EA/DA
  • Work with the vendors for your tools to figure out what’s needed

• Most service accounts don’t need to be EA/DA
Issue multiple accounts to admins

- Regular accounts
  - Use for day to day activity
  - Email, web, etc

- Privileged accounts
  - Only to be used for tasks requiring their privilege
  - Don’t give elevated accounts email addresses
  - Most admin tasks can be handled by right-click->runas and elevating that way while logged into a normal account
Manage Tokens

• Try to minimize running tools that leave tokens lying about as admins
  • PsExec
  • McAfee scan run from EPO leaves tokens too...
• Log out of RDP sessions: Start -> logout
• Reboot periodically to get rid of tokens
  • Yes, even servers
  • Especially servers, where are admins more likely to log into?
The Goodies

• We aren’t going to leave you hanging

• We’ve got some scripts to hopefully help you guys out
Prevent Persistence

- Reset password hash on every smart card account

```powershell
# Import AD Module
Import-Module ActiveDirectory

# Create an array of all accounts that have smart card logon enforced
$ADUsers = (Get-AdUser -Filter * -Properties 'SmartcardLogonRequired' | Where-Object {($_.SmartcardLogonRequired)}).SamAccountName

if ($ADUsers -eq $null) {Write-Error "No Accounts Require Smart Cards for logon"}

# Iterate through each account that has the setting enabled and toggle
ForEach ($User in $ADUsers) {
    Get-AdUser -Identity $($User) | Set-AdUser -SmartcardLogonRequired $false
    Start-Sleep 1
    Get-AdUser -Identity $($User) | Set-AdUser -SmartcardLogonRequired $true
}
```
Different Pwds, Different Hashes

• Don’t use GPP, ensure passwords are unique
  • Set-UniquePassword PowerShell Function

```
EXAMPLE 1
C:\PS>Get-Content c:\demo\computers | Set-UniquePassword -Random | ConvertTo-Csv | Out-File c:\demo\password_update.csv

EXAMPLE 2
C:\PS>\'localhost','test','127.0.0.1' | Set-UniquePassword -Random

EXAMPLE 3
C:\PS>Set-UniquePassword -Random

EXAMPLE 4
C:\PS>Set-UniquePassword -UserName "mspresenters" -Position "Prepend" -Phrase "Recycling#3ftw!" -Token "Serial"
```
Detection is More Realistic

- Write “tools” for every tool out there?
  - Look for signatures in open-source tools
  - Try to stay on top of every new tool
    - Find-PSEncService
    - Find-MsfPsExec
    - Find-WinExec
Why Not Monitor Activity?

- NTLM Network Logons
- Find-NTLMNetworkLogon

```powershell
Function Find-NTLMNetworkLogon {
    $Filter = "*[EventData[Data = 'NtlmSsp ']]"
    $Events = Get-WinEvent -Logname "security" -FilterXPath $Filter | `
    Where-Object {$_._ID -eq 4624}
    if ($Events) {$Events | ForEach-Object {
        $ObjectProps = @{
            'IPAddress' = $_.Properties[18].value;
            'UserName' = $_.Properties[5].value;
            'Domain' = $_.Properties[6].value;
            'Time' = $_.TimeCreated;
            'Workstation' = $_.MachineName
        }
        $Results = New-Object -TypeName PSObject -Property $ObjectProps
        Write-Output $Results
    }
}
```
Find PtH and Insider Threat

- Schedule it and use Send-MailMessage
- Don’t just catch Pen-testers, detect real incidents
Questions?