Version 1.1
Beau Bullock
6-15-2016

Pentest Preparation Guide

This document is designed to assist organizations in preparing for a network-based penetration test. Many of the items listed here are things that can allow an attacker to escalate privileges within an environment with relative ease. This list is by no means a comprehensive list but contains many of the common items found in recent penetration tests performed by the author.

Below is a basic checklist that can be used for tracking these issues.

	Missing Patches
	Group Policy Preference Passwords
	Widespread Local Administrator Accounts
	Weak Password Policy
	Overprivileged Users (admin of local host)
	Overprivileged Users (admin of other hosts)
	Sensitive Files on Shares
	Information Disclosure on Intranet Sites
	NetBIOS and LLMNR Poisoning
	Local Workstation Privilege Escalation

1. Missing Patches

Summary

Systems that are patch delinquent may be vulnerable to a number of attacks. It is important to ensure all systems and software are updated in a timely manner after security patches are released. Many times there may not be an exploit publically available for a given vulnerability however this does not mean that an exploit does not exist.

Identification

Run a vulnerability scanner on your network. Perform authenticated scans against every host.

Or…

Run the Get-ExploitableSystem module from the PowerView PowerShell script

1. Download the script located here: https://github.com/PowerShellEmpire/PowerTools/blob/master/PowerView/powerview.ps1
2. Run cmd.exe as a standard low-privilege user.
3. Run the following command from the Windows command line and press enter: powershell.exe –exec bypass
4. Import-Module [full path to powerview.ps1].
5. Run the command ‘Get-ExploitableSystem’

Remediation

Develop a process for ensuring all systems are being patched. This should include the patching of both the operating system as well as any third-party software within the environment. Use a patch management software to deploy patches to every system on your organization's network on a frequent schedule. Just as important is a patch verification process to ensure patches are being applied correctly.

2. Group Policy Preference Passwords

Summary

In a Windows environment Group Policy Preference files may contain encrypted passwords for accounts set using GPP. Passwords of these accounts are trivially decrypted as Microsoft's AES key for encrypting them is publically documented. Most of the time this is used to set up local administrator accounts for systems in a domain. Another typical use case is configuring scheduled tasks to run as a specific user.

Identification

Option 1: PowerSploit – Get-GPPPassword PowerShell Script

1. Download the script here: https://github.com/mattifestation/PowerSploit/blob/master/Exfiltration/Get-GPPPassword.ps1
2. Open a Windows command terminal.
3. Type "powershell.exe –exec bypass" and press enter.
4. Type "Import-module [full path to Get-GPPPassword.ps1]" and press enter.
5. Type "Get-GPPPassword" and press enter.

Option 2: Metasploit GPP Module
http://www.rapid7.com/db/modules/post/windows/gather/credentials/gpp

Option 3: Windows Command Line

1. Login to a Windows workstation as a domain user
2. Run the following command from the command line:
    - findstr /S cpassword %logonserver%\sysvol*.xml
3. If a result containing a "cpassword" field is returned a password is possibly being exposed by GPP. The PowerSploit Get-GPPPassword PowerShell script can decrypt this field or the Ruby script located here: http://carnal0wnage.attackresearch.com/2012/10/group-policy-preferences-and-getting.html.

Remediation

1. Ensure Microsoft Patch MS14-025 is installed.
2. Delete any Group Policy Preference files that contain account credentials.
3. Change the passwords of any exposed accounts.

3. Widespread Local Administrator Accounts

Summary

It is often that organizations use a local administrator account that includes the use of the same credentials across an environment. When this happens, if the account hasn't

been properly secured it may be used by an attacker to pivot around the environment effectively permitting the attacker to act as a domain administrator.

Identification
Option 1: Organizational Knowledge
If you do not already know if a local administrator account is widespread in use across your environment it may be easiest to ask someone within your organization with this level of information. Often times members of the IT or Helpdesk department may be able to inform you whether a widespread local administrator account is in use. One specific question to ask is if a default administrator account is set up in their current laptop and workstation imaging process.

Option 2: Metasploit SMB_login Module
This module can be used to authenticate to multiple hosts on a network quickly. Attempt to authenticate to multiple hosts using the same local administrator credentials.

Option 3: Use the below command line script. Input the IP addresses of the systems you would like to test into a file called systems.txt. Modify the “Administrator” username and passwords to reflect those in your environment.

```
@FOR /F %s in (systems.txt) DO @net use \\%s\C$ /.\Administrator
AdminPass 1>NUL 2>&1 && @echo %s>>admin_access.txt && @net use
/delete \\%s\C$ > NUL
```

Remediation
Disable the local administrator account if possible. If the account is needed, add it to the “Deny access to this computer from the network” setting in the Local Security Policy. It is possible to do this for every host in the domain with GPO.

4. Weak Password Policy
Summary
While brute force attacks against domain user accounts within an environment may not be a risk due to strict lockout thresholds it may still be possible to perform password spraying attacks. Password spraying is where an attacker will use a list of passwords lower in number than the domain account lockout policy and attempt to authenticate against every user account in the domain with that list. So, if the lockout policy is five (5) attempts, and attacker may try three (3) passwords against every account. This allows an attacker to potentially find a valid password for a user without locking out any accounts on the domain. The chances of an attacker being successful with this attack are greatly increased with a weak password policy because users will typically choose as weak of a password as the enforcement policy allows.

Identification
Password Policy
To identify the domain password policy the following command can be run from the Windows command line:

- net accounts /domain

Password Spraying – **BE EXTREMELY CAREFUL WHEN PASSWORD SPRAYING. THIS CAN POTENTIALLY LOCKOUT ACCOUNTS**.
Option 1: PowerShell
Powerspray - https://github.com/lukebaggett/powerspray

Option 2: Windows command line

1. Create a list of all the usernames within your organization and call it users.txt
2. Create a list of passwords lower in number than the domain account lockout policy. Example: (Password123, Companyname123, Passw0rd)
3. The following command can be ran from the Windows command line. Modify the "DOMAINCONTROLLER" and "DOMAIN" fields to reflect your own organization:

```
@FOR /F %n in (users.txt) DO @FOR /F %p in (pass.txt) DO @net use
\\DOMAINCONTROLLER\IPC$ /user:DOMAIN\%n %p 1>NUL 2>&1 && @echo [*]
%n:%p && @net use /delete \\DOMAINCONTROLLER\IPC$ > NUL
```

Remediation

Implement a stronger password policy. Increase the length requirement for passwords, and ensure passwords are being changed often. Consider requiring passwords be at least sixteen (16) characters in length. Train your users to choose length over complexity.

Also, perform password spraying attacks against your user base to find any credentials that may be easily sprayed before an attacker does. This will also help you tune any detection devices as password spraying generates many failed logins.

5. Overprivileged Users (admin of local host)

Summary

In some environments it may be organizational policy to grant administrative access to every user. This gives an attacker much greater freedom in what they can do to attack other systems and avoid detection.

Identification

Organizational Knowledge
It may be easiest to ask someone within your organization with this level of information. Oftentimes members of the IT or Helpdesk department may be able to inform you whether it is policy to grant all users administrative access or not.

Remediation

Only grant administrative access where necessary, not globally.

6. Overprivileged Users (admin of other hosts)

Summary

User groups that include everyone in the domain have a greater potential for granting additional privileges that most users may not need. An example of this would be the "Domain Users" group. If this group is added to the "Local Administrators" group of a system it effectively grants everyone in the domain administrative access to it. If a non-

administrative attacker can find one of these systems it may create a node for privilege escalation within the environment.

Identification
Option 1: Powershell - Veil-PowerView Find-LocalAdminAccess
https://github.com/Veil-Framework/PowerTools/blob/master/PowerView/powerview.ps1
http://www.harmj0y.net/blog/penetesting/finding-local-admin-with-the-veil-framework/

Option 2: Powershell - Veil-PowerView Invoke-ShareFinder
1. Run cmd.exe as a standard low-privilege user.
2. Download powerview.ps1 from the link above.
3. Run the following command from the Windows command line and press enter: powershell.exe –exec bypass
4. Import-Module [full path to powerview.ps1].
5. Invoke-ShareFinder –CheckShareAccess –Verbose | Out-File –Encoding ascii found_shares.txt
6. Check the share list for ADMIN$ shares that your current user has access to.
7. Potentially, these systems may have misconfigured privilege settings that you will want to assess.

Remediation
Only grant administrative access where necessary. User groups should be tightly restricted when setting up administrative access.

7. Sensitive Files on Shares
Summary
Employees are human and can make mistakes. Attackers can take advantage of this by gathering information that employees place in locations available to them. Employees sometimes place very sensitive information on shares including usernames, passwords, PCI data, PII, or PHI.

Identification
Option 1: Powershell - Veil-PowerView Invoke-FileFinder
1. In the previously described issue above we used Veil-PowerView Invoke-ShareFinder to create a list of shares available to a user. We will need this list to use with FileFinder.
2. Follow the guide above for importing the powerview.ps1 module into PowerShell.
3. Run: Invoke-FileFinder -Verbose -ShareList .\found_shares.txt -OutFile found_files.csv
4. This will create a list of files available to the current user that contain the following in their titles: ‘*pass*’, ‘*sensitive*’, ‘*admin*’, ‘*secret*’, ‘*login*’, ‘*unattend*.xml’, ‘*.vmdk’, ‘*creds*’, or ‘*credential*’.
5. The “-Terms” flag can be passed to the FileFinder command above to search for custom terms such as “payroll”, “scada”, ”pci”, etc.

Option 2: Windows Explorer
1. Open Windows Explorer to any commonly used internal file share that a large number of people have access to.

2. Using the built-in search bar search for files containing “password”, “credential”, or other similarly sensitive phrase.
3. Repeat with all other permissive file shares.

Remediation

Locate sensitive files on network shares available to low-privileged users. Remove any files deemed sensitive from these locations. Train users about the dangers of storing sensitive information in unencrypted files. Assess the privilege settings of these file locations to determine if they align with your organizational security policy.

8. Information Disclosure on Intranet Sites

Summary

Similar to the previous issue this vulnerability arises when sensitive information is placed in a location accessible to many users in the environment. Good examples of this are Intranet sites like SharePoint.

Identification

Utilize search functions built into Intranet sites like SharePoint, and IT Helpdesk applications. Search for terms like “password”, “mainframe”, “scada”, “credit card”, “social security number”, etc.

Remediation

Ensure sensitive data is not accessible by simply searching Intranet sites. Remove or restrict access to any sensitive data so that users who need access can view the information.

9. NetBIOS and LLMNR Poisoning

Summary

Both NetBIOS and LLMNR help computers identify other hosts on a network when DNS fails. If an attacker can respond to these types of requests it may be possible for them to intercept NTLM authentication hashes. The attacker can then attempt to crack these hashes and potentially gain access to the network as that user.

Identification

Option 1: Assess GPO settings, and local security policies on domain systems to determine if these are enabled.

Option 2: SpiderLabs Responder

1. Can be used to poison NetBIOS and LLMNR to obtain NTLM hashes.

This program must be run from a Linux machine.
https://github.com/Spiderlabs/Responder
There is also a Windows version that has limited functionality.
https://github.com/lgandx/Responder-Windows

Remediation

Use GPO to disable LLMNR. Unfortunately, there isn't an option in Group Policy for disabling NetBios. However, a VB script located at the following URL can be pushed down via GPO to disable NetBios:
https://technet.microsoft.com/en-us/library/ee692589.aspx

LLMNR

1. In the Local Group Policy editor navigate to Local Computer Policy>Computer Configuration>Administrative Templates>Network>DNS Client
2. Click on "Turn Off Multicast Name Resolution" and set it to "Enabled".

NetBios

1. Open Control Panel
2. Under "Network and Internet", Click "View network status and tasks"
3. Click "Change adapter settings"
4. Double-click "local area connection"
5. Double-click "Internet Protocol Version 4"
6. Click "Advanced"
7. Click "WINS"
8. Click on "Disable NetBIOS over TCP/IP"

10. Local Workstation Privilege Escalation

Summary

There are many methods attackers can use to elevate their privileges on a system. Ensuring that as many of the known local privilege escalation vulnerabilities are eliminated from a system is very important.

Identification

Option 1: Powershell - Veil PowerUp
https://github.com/Veil-Framework/PowerTools/blob/master/PowerUp/PowerUp.ps1

1. Download the PowerUp PowerShell script from the link above.
2. Run cmd.exe as a non-administrative user.
3. Load PowerShell – powershell.exe -exec bypass
4. Import PowerUp – Import-Module [full path to powerup.ps1]
5. Invoke all checks and write it to a file - Invoke-AllChecks | Out-File -Encoding ASCII checks.txt

Option 2: Windows-Privesc-Check
http://pentestmonkey.net/tools/windows-privesc-check

Remediation

If PowerUp or Windows-Privesc-Check find any potential privilege escalation vectors each finding should be assessed to determine exploitability. Eliminate all privilege escalation vectors.