TwC Next
Marking a Milestone.
Continuing Our Commitment.

Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques

Mitigating the risk of lateral movement and privilege escalation

Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Authors

Patrick Jungles
Microsoft Trustworthy Computing

Mark Simos
Microsoft Consulting Services

Roger Grimes
Microsoft IT Information Security and Risk Management

Aaron Margosis
Microsoft Consulting Services

Laura Robinson
Microsoft IT Information Security and Risk Management

Contributors

Microsoft Office 365 Security

Joe Bialek

Benjamin Godard

Paul Rich

Justin Hendricks

Microsoft Windows Security and Identity Team

Nathan Ide

Paul Leach

Paul Miller

Michiko Short

Microsoft Trustworthy Computing

Adam Shostack

David Seidman

Ellen Cram Kowalczyk

Georgeo Pulikkathara

Graham Calladine

Ian Hellen

John Lambert

Mike Reavey

Jonathan Ness

Mark Cartwright

Mark Oram

Tim Rains

Matt Thomlinson

Ryan Heffernan

Sean Krulewitch

Microsoft Consulting Services

Al Tieman

Andrew Idell

David Hoyle

Fernando Cima

Janwillem Kok

Jerry Cochran

Jiri Formacek

Matt Kemelhar

Michael Howard

Nate Morin

Patrick Arnold

Sean Finnegan

Interactive Entertainment Business

Mark Novak

Microsoft Server and Tools Business

Dean Wells

Microsoft IT Information Security and Risk Management

Bret Arsenault

Brian Fielder

Eric Leonard

Vexcel

Rich Levy

Contents

Executive Summary

A *Pass-the-Hash (PtH) attack* uses a technique in which an attacker captures account logon credentials on one computer and then uses those captured credentials to authenticate to other computers over the network. A PtH attack is very similar in concept to a password theft attack, but it relies on stealing and reusing password hash values rather than the actual plaintext password. The password hash value, which is a one-way mathematical representation of a password, can be used directly as an authenticator to access services on behalf of the user through single sign-on (SSO) authentication.

To use this technique, an attacker must first obtain local administrative access on a computer in the organization to steal credentials from the computer's disk and memory. This level of privilege allows the attacker to not only obtain password hashes, but also any other credentials stored on the compromised computer. An attacker can obtain local administrative access by either compromising the built-in local administrator account, a domain account with membership in the local administrators group, or another local account that can be used to install drivers, applications, and execute applications that allow direct interaction with the hard disk or volatile memory.

The PtH technique allows an attacker who has compromised a single computer to gain access to connected computers, including domain controllers and other servers storing sensitive information. For this reason, mitigating the risk of PtH attacks and other similar credential theft attacks can significantly improve the security posture of an Active Directory environment. The PtH attack is one specific type of credential theft and reuse attack. While this document focuses on Windows operating systems, other operating systems are vulnerable to similar credential theft and reuse attacks.

These attacks have become common and concern many of our customers. This document is designed to assist your organization with defending against these types of attack. Information about how PtH attacks and related credential theft attack techniques work is provided, as well as how your organization can use security mechanisms in Windows operating systems to mitigate the risk of these attacks.

Introduction

As the tools and techniques for credential theft and reuse attacks like the Pass-the-Hash (PtH) attack improve, malicious users are finding it easier to achieve their goals through these attacks. The PtH attack is one of the most popular types of credential theft and reuse attack seen by Microsoft to date, although this white paper also discusses other similar attacks. Other credential theft attacks include key logging and other plaintext password capture, passing tickets, token impersonation, and man-in-the-middle attacks.

We have recently observed the active use of PtH techniques by determined adversaries in targeted attacks. For more details, see the Microsoft white paper Determined Adversaries and Targeted Attacks[1] which includes information about attacker motivation, goals, and alternative attack methods that are not discussed in this white paper.

Attackers can use multiple tools and techniques to perform a credential theft and reuse attack, some of which are easily available from the Internet. While this paper focuses on Windows operating systems, attackers can perform credential theft and reuse attacks on any operating system and these attacks are a threat to other platforms as well. PtH attacks and similar credential theft attacks take advantage of the same flexibility of single sign-on (SSO) authentication mechanisms that allow users to seamlessly authenticate to network resources. SSO mechanisms require the computer to maintain a copy of authentication credentials to be used on behalf of the user for certain tasks, such as checking email or accessing a remote resource. Without these credentials, the computer would need to prompt the user to enter their authentication credentials every time a network authentication is performed.

A PtH attack can have a significant impact on an environment managed by Active Directory. If successful, the attack may result in the compromise of privileged administrative accounts, such as those that are members of the Domain Admins or Enterprise Admins groups.

For these reasons, it is critical to any organization's security posture to evaluate the risk of PtH attacks and similar credential theft attacks, and to implement mitigations to reduce or manage these risks. The recommended mitigations in this paper are intended to help you significantly minimize the risk and impact of PtH attacks and other credential theft attacks in your organization. We also recommend educating decision makers involved in business risk management and administrative staff with this information. This especially applies to administrators who require Domain Administrator or equivalent accounts for their daily jobs.

The first part of this document discusses PtH attacks against Windows operating systems, how the attack is performed, and recommends mitigations for PtH attacks and

[1] http://www.microsoft.com/en-us/download/details.aspx?id=34793

similar credential theft attacks. More technical details and background information is provided in the "Additional technical information" section. The remainder of this document contains step-by-step instructions on deploying the mitigations described in the first part of the document.

What is the PtH attack?

The Pass-the-Hash (PtH) attack and other credential theft and reuse types of attack use an iterative two stage process. First, an attacker must obtains local administrative access on at least one computer.. Second, the attacker attempts to increase access to other computers on the network by:

1. Stealing one or more authentication credentials (user name and password or password hash belonging to other accounts) from the compromised computer.
2. Reusing the stolen credentials to access other computer systems and services.

This sequence is often repeated multiple times during an actual attack to progressively increase the level of access that an attacker has to an environment.

A *password hash* is a direct one-way mathematical derivation of the password that changes only when the user's password changes. Depending on the authentication mechanism, either a password hash or a plaintext password can be presented as an authenticator to serve as proof of the user's identity to the operating system. Also, an authenticator may be stored in the computer's memory to support single sign-on (SSO) which could be subject to theft.

After an attacker has stolen the user name and corresponding authenticator, the attacker is effectively in control of that account and gains access to all the resources, rights, and privileges of that account. If the compromised account is a *privileged account*, such as a domain administrator, the attacker gains that account's privileged access (e.g., domain administrative rights). Any other account credentials stored on a compromised computer can be stolen, including those for local user accounts, domain user accounts, service accounts, and computer accounts. Domain accounts that have never been used to log on to a compromised computer cannot be stolen from that computer.

In order for an attacker to reuse a stolen password hash on another host, the following requirements must be met:

1. The attacker must be able to contact the remote computer over the network, and the computer must have listening services that accept network connections.
2. The account and corresponding password hash value obtained from the compromised computer must be valid credentials on the computer being authenticated to (for example, if both computers are in the same domain, or local accounts with the same user name and password exist on both computers).

3. The compromised account must have the **Network Logon** user right on the remote computer.

> Password hashes may only be used for network logons, but plaintext passwords may be used to authenticate interactively. Plaintext passwords can allow an attacker to access other services and features, such as Remote Desktop.

Table 1, "PtH Attack Activities," lists the types of PtH attack activities that an attacker can perform after the initial compromise.

Table 1. PtH Attack Activities

Attack activities	Description
Lateral movement	In this activity, the attacker uses the credentials obtained from a compromised computer to gain access to another computer of the same value to the organization. For example, the attacker could use stolen credentials for the built-in local Administrator account from the compromised computer to gain access to another computer that has the same user name and password.
Privilege escalation	In this activity, the attacker uses the credentials obtained from a compromised computer to gain access to another computer of a higher value to the organization. For example, an attacker who has compromised a workstation computer could gain administrative access to a server computer by stealing the credentials of server administrators who log on to the compromised workstation.

It is important to reiterate that the attacker must have administrative access on the initial compromised computer in order to steal these credentials. *Administrative Access* to a computer can include the ability to run a program or script with an account in the local Administrators group, but this type of access can also be achieved through the use of "admin-equivalent" privileges, such as those used for "Debug programs," "Load and unload device drivers" or "Take ownership" privileges.

With administrative access, an attacker can steal credentials from several locations on the computer, including:

- The Security Accounts Manager (SAM) database.
- Local Security Authority Subsystem (LSASS) process memory.
- Domain Active Directory Database (domain controllers only).
- The Credential Manager (CredMan) store.
- LSA Secrets in the registry.

For more information about credential storage locations, see Table 4, "Windows Credential Types" in the "Windows authentication" section under "Additional technical information" in this document.

It is very difficult to distinguish activity by attackers using stolen credentials from authorized activity. If System and Event Logging is enabled, all authentication activity, malicious or not, will appear as normal logons. Administrators attempting to detect malicious activities will need to focus on "authorized" activity that is unexpected.

PtH attack and other credential theft attack risk markers

An organization has more risk of a PtH attack and other credential theft attacks if one or more of the following risk factors are present:

- High privilege domain accounts are used to log on to workstations and servers.
- Applications or services run with high privilege accounts.
- Scheduled tasks run with high privilege accounts.
- Ordinary user accounts (Local or Domain) are granted membership to the local Administrators group on their workstations.
- Highly privileged user accounts can be used to directly browse the Internet from workstations, domain controllers, or servers.
- The same password is configured for the built-in local Administrator account on most or all workstations and servers.

  > **Note:** Since the release of Windows Vista, the built-in Local Administrator account is disabled by default in Windows operating systems.

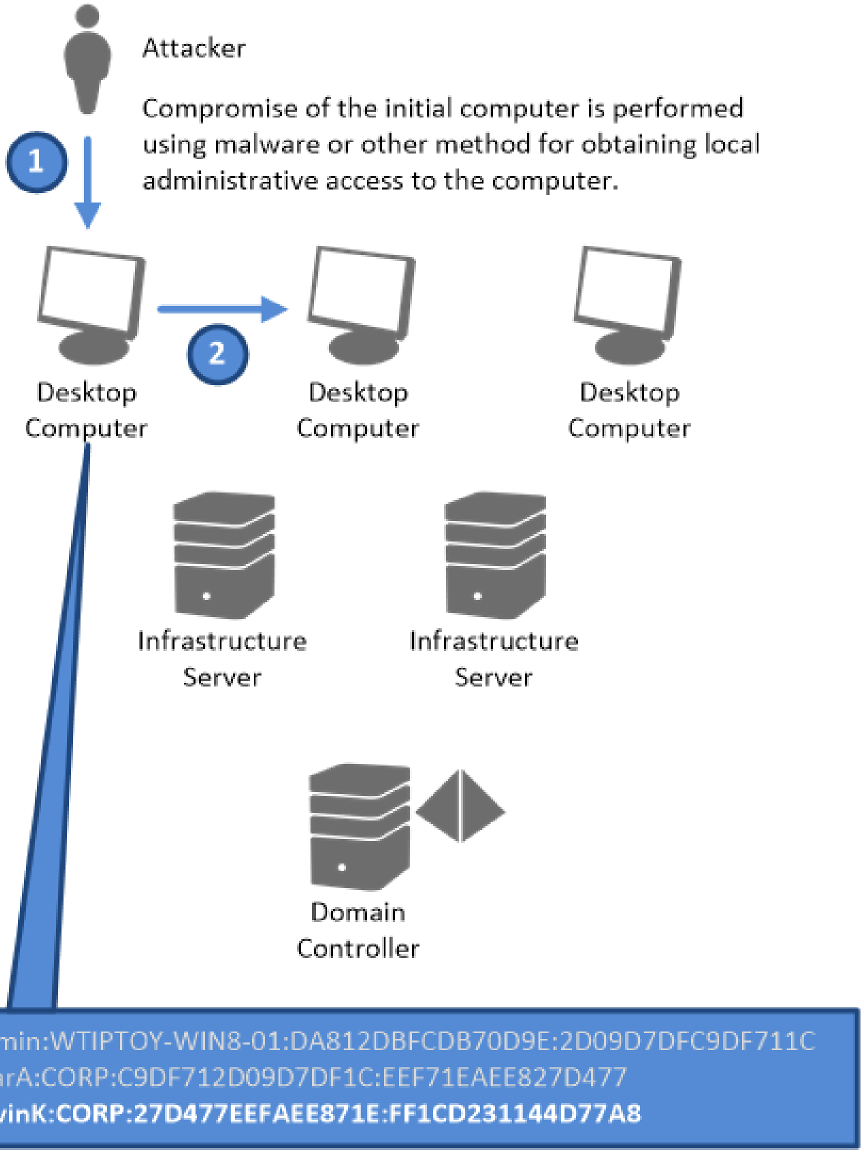
- Account termination is not enforced on accounts in the Domain Admins, Enterprise Admins or other high privileged groups where they are no longer needed.
- Security updates are not applied quickly to operating systems and applications.
- Logons can occur with privileged accounts to less secure computers that are potentially compromised.
- Operations processes and personnel share privileged account credentials.
- Too many administrators use high privileged accounts for administrative tasks.
- Service accounts are granted domain administrative privileges.

For details and other practices that can decrease the risk of PtH attacks, see the "Additional recommendations" section.

How is a PtH attack performed?

While the tools and methods of obtaining administrative rights on the initial computer vary, the subsequent Pass-the-Hash (PtH) attack steps that take place are fairly consistent. The initial steps in this sequence are illustrated in

Figure 1 and Figure 2 at a high level. Other credential theft and reuse attacks, such as stealing and passing Kerberos Ticket Granting Tickets (TGTs) or plaintext passwords, would typically follow a similar process after the credential has been stolen.



Compromise of the initial computer is performed using malware or other method for obtaining local administrative access to the computer.

```
Admin:WTIPTOY-WIN8-01:DA812DBFCDB70D9E:2D09D7DFC9DF711C
PilarA:CORP:C9DF712D09D7DF1C:EEF71EAEE827D477
KevinK:CORP:27D477EEFAEE871E:FF1CD231144D77A8
```

Figure 1. Initial high-level PtH attack sequence with lateral movement

The following describes a high-level example of a typical PtH attack using commonly available PtH tools based on the illustrations in

Figure 1 and Figure 2:

1. An attacker obtains local administrative access to a computer on the network by enticing a victim into executing malicious code, by exploiting a known or unpatched vulnerability, or through other means. The attacker then takes advantage of this administrative access to obtain password hashes from the local SAM database on disk, and by reading or injecting hashes into process memory where credentials are stored. The attacker will use these newly obtained password hashes to perform lateral movement or privilege escalation in subsequent steps.

   After the password hashes are captured, the attacker typically replaces the password hash of the currently running Windows session with the newly captured credentials. Other methods are also available for the attacker to use the obtained password hash.

   > **Note:** An attacker is limited to the logon credentials that they can obtain from the compromised computer. Accounts the attacker cannot harvest locally cannot be used in further attacks. If a Domain Admin account is never used for authentication to workstations, this account will not be available to an attacker that has compromised these workstations.

2. The attacker uses the stolen credentials to connect to other computers on the network using built-in Windows commands, such as net use, or net view, or by downloading and executing utilities like psexec.exe.

   > **Note:** Windows built-in tools by default only support plaintext passwords or the use of current session credentials for authentication through network logon. Attack tools can allow the attacker to use any credential type by either creating a new session command prompt or overwriting the hashes for the current session with these newly obtained credentials to impersonate the target user.

   If local privileged accounts, such as the built-in local Administrator account, have the same password on the compromised computer as other computers on the network, the attacker can log on to those computers using the stolen password hashes. This can be done because NT password hashes are created using an *unsalted* MD4 algorithm, so they are identical on each computer. This allows the attacker to match the username and password hash required on network logons.

   The attacker then continues to perform lateral movement by compromising other computers on the network until the attacker can compromise a computer with a privileged domain account. (

   Figure 1 previously illustrates the first two steps of this attack: initial compromise and lateral movement).

Figure 2 illustrates the later high-level stages of a PtH attack.

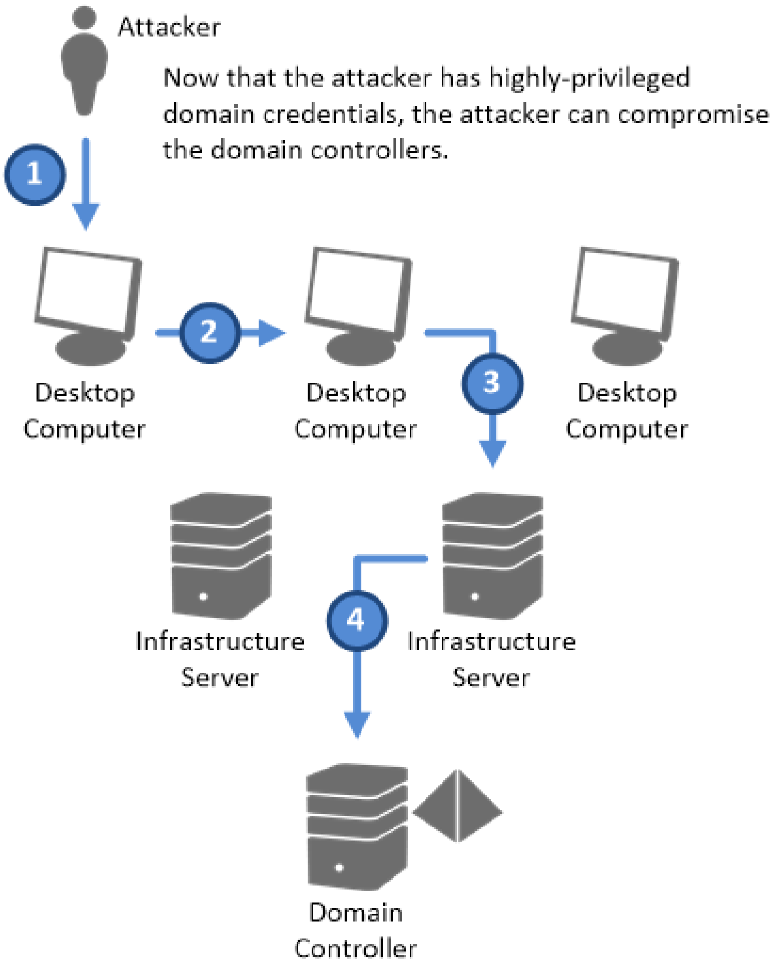


Figure 2. High-level later stages of a PtH attack with both lateral movement and privilege escalation

3. The attacker compromises a computer containing a higher privileged domain account or a service account using the same techniques. This account allows the attacker to compromise a server resource resulting in privilege escalation. The attacker may also continue to perform lateral movement within the server environment to compromise other servers until a server with Domain administrator credentials is compromised.
4. If the attacker obtains the credentials for a domain administrator or an equivalent account with privileged access to Active Directory, then the attacker can compromise all of the computers in the Active Directory forest. The attacker may also compromise other domains that trust the compromised domain.

Even if the attacker cannot compromise an account that is a member of the Domain Admins group or another highly privileged group, the attacker can often obtain significant access to the domain infrastructure, including the ability to steal, alter, and destroy data stored on compromised servers and workstations. Attackers are also likely to entice administrators to log on to compromised computers with privileged credentials.

If an attacker obtains credentials for an account that is a member of the Domain Admins group or an equivalent privileged account, that attacker can gain effective control of all computers and services under the administrative scope of that account.

An attacker can perform a complete compromise of an infrastructure after the first attack or after carrying out several lateral movements and privilege escalations. This attack sequence can happen very quickly, often in a matter of minutes.

Why can't Microsoft release an update to address this issue?

Credential theft and reuse is not a problem that can be addressed with a simple software update. For a product change to be effective in mitigating PtH and similar attacks, any change must deny attackers the ability to perform one or all of the following:

- **Find where credentials are stored**: The current security research community and attack landscape are very knowledgeable about Windows internals. If changes to the encryption or obfuscation methods (or both) are engineered and implemented, it is unlikely to be effective as it can be discovered and reverse-engineered within a relatively short time. Security by obscurity will not deter attackers in the long term.
- **Extract credentials:** PtH attacks and other credential theft attacks exploit the access that an attacker gains by compromising an account in the local Administrators group. These accounts have complete control over the computer's memory, disks, and processor resources.

  While the methods used to encrypt and hide credentials can be changed, the operating system still must have the ability to retrieve them. An attacker who can execute code as the local administrator has the same security privileges as the operating system and can retrieve credentials in the same way that the operating system does. A significant step in the right direction is to prevent attackers from obtaining control of these accounts by restricting local administrative access from standard users, a mitigation that is available today.
- **Reuse credentials:** The same single sign-on (SSO) mechanism that brings significant benefits to the user experience also increases the risk of a PtH attack if an operating system is compromised. Credentials must be stored or cached to allow the operating system to perform actions on behalf of the user to make the system usable. If credentials that a user typed at logon are not available or cannot be reused, the user must retype them countless times in a distributed environment that uses Active Directory. Additionally, keystroke logging and other attack techniques to capture credentials can still be performed. Limiting delegation or where credentials can be used are positive steps toward preventing PtH attacks. The mitigation recommendations in this document address these challenges.

While we will continue to investigate platform modifications to enhance the security of Windows operating systems, this is not an attack that can be addressed with a single fix or update. For example, changing how the Windows Local Security Authority Subsystem (LSASS) stores credentials only requires attackers to update existing tools to support such modifications. We are actively investigating the optimal means to help our customers mitigate these risks with product updates and releases.

How can your organization mitigate the risk of a PtH attack?

This section provides mitigation strategies that you can use in your organization to help prevent both lateral movement and privilege escalation by decreasing the impact of credential theft or illicit reuse on computers running Windows operating systems in your environment. These mitigations have been chosen from a larger list of considerations because they are effective, practical, and broadly applicable to different domain configurations. These recommended mitigations also don't have significant prerequisites, so they can be deployed relatively quickly to mitigate PtH attacks and other related threats. The sections "Additional recommendations" and "Analysis of other potential mitigations" are also included in this portion of the document.

Table 2, "Mitigations, More Recommendations, and Other Mitigation Analysis," provides a summary of these areas and their effectiveness, as well as the perceived effort required to implement each solution, and the applicability of each mitigation to lateral movement or privilege escalation as it relates to PtH attacks and credential theft and reuse.

Table 2. Mitigations, More Recommendations, and Other Mitigation Analysis

Mitigation	Effectiveness	Effort required	Privilege escalation	Lateral movement
Mitigation 1: Restrict and protect high privileged domain accounts	Excellent	Medium	√	-
Mitigation 2: Restrict and protect local accounts with administrative privileges	Excellent	Low	-	√
Mitigation 3: Restrict inbound traffic using the Windows Firewall	Excellent	Medium	-	√

More recommendations	Effectiveness	Effort required	Privilege escalation	Lateral movement
Remove standard users from the local administrators group	Excellent	High	√	-
Limit the number and use of privileged domain accounts	Good	Medium	√	-
Configure outbound proxies to deny Internet access to privileged accounts	Good	Low	√	-
Ensure administrative accounts do not have email accounts	Good	Low	√	-
Use remote management tools that do not place reusable credentials on a remote computer's memory	Good	Medium	√	-
Avoid logons to less secure computers that are potentially compromised	Good	Low	√	√
Update applications and operating systems	Partial	Medium	-	-

Secure and manage domain controllers	Partial	Medium	-	-
Remove LM hashes	Partial	Low	-	-
Other mitigation	**Effectiveness**	**Effort required**	**Privilege escalation**	**Lateral movement**
Disable the NTLM protocol	Minimal	High	-	-
Smart cards and multifactor authentication	Minimal	High	-	-
Jump servers	Minimal	High	√	-
Rebooting workstations and servers	Minimal	Low	-	-

> **Note:** Although the recommended mitigations should have a minimal negative impact for most organizations, we strongly recommend testing your systems before implementing any mitigation in a production environment. Ensure to test each of these mitigations before implementing them, identify relevant rollback plans, and gradually deploy any changes to minimize the impact of daily IT operations in your organization. These recommendations are not a substitute for updating and securing your computers against compromise by attackers. These mitigations are defense-in-depth measures designed to ensure that your environment is protected even if these measures fail.

Mitigation 1: Restrict and protect high privileged domain accounts

Some organizations allow high privilege accounts like those that are members of the Domain Admins group to perform general administration tasks, or to log on to user desktops or other systems used for email and Internet browsing, potentially exposing these credentials to attackers. We recommend restricting highly privileged accounts so that they can only be used to log on to sufficiently secured systems that require them. In addition, allowing the use of Kerberos delegation with privileged accounts can make it easier for an attacker to reuse them to access additional network resources. For more details on delegation, see Delegating Authentication.

Main objective: This mitigation reduces the risk of administrators from inadvertently exposing privileged credentials to higher risk computers.

> Domain administrators logging onto a compromised computer may still briefly expose their credentials even if the recommended tasks bellow are implemented. Attackers can capture these credentials during logon despite the account not being authorized to successfully logon.

How: Completing the following tasks is required to successfully implement this mitigation:

- Restrict domain administrator accounts and other privileged accounts from authenticating to lower trust servers and workstations.
- Provide admins with accounts to perform administrative duties that are separate from their normal user accounts.
- Assign dedicated workstations for administrative tasks.
- Mark privileged accounts as “sensitive and cannot be delegated” in Active Directory.
- Do not configure services or schedule tasks to use privileged domain accounts on lower trust systems, such as user workstations.

Outcome: An attacker cannot steal credentials for an account if the credentials are never used on the compromised computer. Using this mitigation significantly reduces the risk of attackers compromising privileged accounts.

For more information about how to configure your environment with the recommendations for this mitigation, see the section "Mitigation 1: Restrict and protect high privileged domain accounts" in Appendix A, "Step-by-step instructions to mitigate PtH attacks."

Mitigation 2: Restrict and protect local accounts with administrative privileges

Accounts with administrative access on a computer can be used to take full control of the computer. And if compromised, an attacker can use the accounts to access other credentials stored on this computer.

In addition, many organizations have deployment and operational processes that result in defining the same administrative local account and password on many computers. Maintaining identical passwords makes it significantly easier for attackers to compromise all computers that use them and obtain all credentials stored on these computers. IT support processes typically do not require the built-in local administrator account to log on over a network connection, which is a common attack vector for lateral movement using credential theft.

> **Note:** If all administrative local accounts are already disabled, the steps in Mitigation 2 are not required. Specific instructions for disabling accounts is not included in Mitigation because implementing this strategty requires a design tailored to how your organization supports local and remote users.

Main objective: This mitigation restricts the ability of attackers to use administrative local accounts for lateral movement PtH attacks.

How: Completing one or a combination of the following tasks is required to successfully implement this mitigation on all computers in the organization:

1. Enforce the restrictions available in Windows Vista and newer that prevent local accounts from being used for remote administration.
2. Explicitly deny network and Remote Desktop logon rights for all administrative local accounts.
3. Create unique passwords for local accounts with administrative privileges.

Outcome: An attacker who successfully obtains local account credentials from a compromised computer will not be able to use those credentials to perform lateral movement on the organization's network.

For more information, see "Mitigation 2: Restrict and protect local accounts with administrative privileges" in Appendix A, "Step-by-step instructions to mitigate PtH attacks."

Mitigation 3: Restrict inbound traffic using the Windows Firewall

One of the most important prerequisites for an attacker to conduct lateral movement or privilege escalation is to be able to contact other computers on the network.

Main objective: This mitigation restricts attackers initiating lateral movement from a compromised workstation by blocking inbound connections on all other workstations with the local Windows Firewall.

How: This mitigation restricts all inbound connections to all workstations except for those with expected traffic originating from trusted sources such as helpdesk workstations, security compliance scanners and management servers. Applications that do not directly accept authentication credentials may also be allowed through the Windows firewall without incurring the risks of credential theft and reuse.

Outcome: Enabling this mitigation will prevent an attacker from connecting to other workstations on the network using any type of stolen credentials.

For more information on how to configure your environment with this mitigation, see the section "Mitigation 3: Restrict inbound traffic using the Windows Firewall" in Appendix A, "Step-by-step instructions to mitigate PtH attacks."

Additional recommendations

This section discusses additional recommendations for protecting computers against PtH attacks and other credential theft attacks. These recommendations may not directly protect against PtH attacks or be as effective, practical and broadly applicable in different domain configurations. However, we strongly encourage using them because they significantly increase the security posture of organizations, as well as indirectly protect organizations against these types of attacks.

Do not allow browsing the Internet with highly privileged accounts

Internet activities, such as browsing the Internet and reading email, are inherently high risk activities because they process content accessed from the Internet that is potentially malicious or dangerous. If user accounts with administrative rights are used to perform these activities, a potential compromise on the computer or application can lead to immediate attacker control of those administrative rights. For these reasons, we recommend separating administrative rights from Internet access where possible by doing the following:

- Remove standard users from the local Administrators group.
- Configure outbound proxies to deny Internet access to privileged accounts.
- Ensure administrative accounts do not have email accounts or mailboxes associated with them.

Remove standard users from the local Administrators group

We recommend not granting membership in the local Administrators group of the organization's workstations to standard user accounts that run Internet applications, such as those used for web browsing and email. Many organizations have already implemented this configuration, and others are implementing it as they deploy the latest Windows operating systems.

This strategy strengthens an organization's resilience to a PtH attack by increasing the barrier that an attacker must overcome to obtain the local administrative access required to start a credential theft attack. An attacker who has compromised a standard domain user account must overcome the additional operating system security boundary to elevate to the administrator level in order to steal credentials. If the user is not a member of the local Administrator group, attackers attempting to compromise a user account must find a different way to elevate their privileges locally.

While restricting administrative rights is a strong defense against PtH attacks and credential theft, it may not be feasible to apply this mitigation in some organizations. Examples include organizations that do not have a robust management infrastructure designed to handle administrative tasks that users can no longer perform, or those that depend on legacy applications that do not work correctly without administrative rights.

> **Note:** The latest Windows operating systems include a set of technologies known as User Account Control (UAC) that are designed to help users run tasks without administrative privileges and mitigate the impact of malicious programs. For more information about UAC, see the User Account Control Technical Reference.

If a large number of standard users in your organization are currently operating with local administrative privileges, converting these users to standard privileges should include the following activities:

- Application compatibility testing to ensure that legacy applications continue to operate correctly for standard users.
- Using deployment processes and tools to deploy new software and updates without administrative rights.
- Updating helpdesk and support processes to ensure support is available for users without local administrative rights.

Configure outbound proxies to deny Internet access to privileged accounts

Many products on the market that proxy user Internet traffic offer the capability to authenticate users and allow or block access using groups in Active Directory. We recommend blocking Internet access for domain accounts that are members of highly privileged groups.

Ensure administrative accounts do not have email accounts

Ensure that the domain privileged accounts are not associated with mailboxes in Microsoft Exchange or any other email system.

Use remote management tools that do not place reusable credentials on a remote computer's memory

Some remote authentication methods allow you to perform administrative tasks on the remote computer without storing the administrator account password hash, Kerberos ticket granting tickets (TGTs), or other reusable credentials on the remote computer's memory. Therefore, using only management tools with these authentication mechanisms can reduce the risk of PtH attacks.

This mitigation has maximum effect when using a dedicated administrative workstation, as described in "Task 2: Create specific administrative workstation hosts for administrators" in the section "Mitigation 1: Restrict and protect high privileged domain accounts" of Appendix A, "Step-by-step instructions to mitigate PtH attacks."

You can use Table 7, "Connection Methods and Where the Credentials Are Created and Cached" in this document to identify common administrative tools and how much risk of credential exposure they may incur.

Avoid logons to less secure computers that are more likely to be compromised

When a highly-privileged domain account is used to log on to workstations or member servers that may be compromised, attackers who have compromised that computer may harvest those credentials. See "Mitigation 2: Restrict and protect high privileged domain accounts" in Appendix A, "Step-by-step instructions to mitigate PtH attacks" for information about how to restrict privileged account usage by location.

You can investigate the computer using a number of online or offline techniques. How your organization performs its investigation should always take into account legal considerations for evidence preservation, regulatory reporting requirements, and any potential operational impacts. You may also want to consider consulting a professional incident response or forensics team to assess your organization's level of compromise and develop the most effective mitigation plan for your situation.

Update applications and operating systems

Application or operating system vulnerabilities that have not been remedied contribute to credential theft attacks by providing an avenue to use well-known published exploits to circumvent security controls or elevate privileges. Applying security updates to operating systems and applications forces attackers to find unknown vulnerabilities or other means of attack that require user interaction.

Limit the number and use of privileged domain accounts

Granting membership in the Administrators, Domain Admins, and Enterprise Admins groups in a domain or forest creates high value targets for attackers. The greater the number of members in these groups, the greater the likelihood that a privileged user may inadvertently misuse these credentials and expose them to attackers.

Every workstation that a privileged domain user logs on to provides another location where privileged credentials can be stolen. We strongly advise organizations to reduce membership in privileged groups, and stringently control where and how privileged accounts are used. For more information, see "Mitigation 2: Restrict and protect high privileged domain accounts" in Appendix A, "Step-by-step instructions to mitigate PtH attacks."

Secure and manage domain controllers

Because domain controllers store credential password hashes of all accounts in the domain, they are a high value target for attackers. If your domain controllers are not stringently updated and secured, attackers may also compromise them and the domain (and forest) through a vulnerability that has not been addressed. We recommend ensuring that the domain controllers in your environment do not run unnecessary software, are promptly and regularly updated, and are configured with appropriate security settings.

Installed applications and management agents on domain controllers may provide a privilege escalation path for attackers to compromise the management service or administrators of that service. Consider the management tools and services that your organization uses to manage domain controllers and their administrators equally important to the security of the domain controllers and domain administrator accounts. Ensure to secure these services and administrators with equal effort.

You can obtain Microsoft recommendations for domain controller configurations that you can distribute using the Security Compliance Manager (SCM) tool. For more information, see the Microsoft Security Compliance Manager page on TechNet.

Remove LM hashes

You should disable and remove LAN Manager (LM) hashes in the computer’s local SAM and Active Directory domain databases to reduce the risk of attackers obtaining these legacy password hashes. You may have LM hashes for one or more user accounts, if either of the following conditions is true:

- Your domain was created with a version of Windows released prior to Windows Server 2008.
- You have explicitly disabled the Group Policy setting **Network security: Do not store LAN Manager hash value on next password change** on a group policy object applying to domain controllers.

When a user changes a password, Active Directory always stores a copy of the NT hash and it can also store a LM hash if the password is compatible with LM and the setting **Network security: Do not store LAN Manager hash value on next password change** is disabled. This setting is enabled by default in Windows operating systems, starting with the release of Windows Vista and Server 2008. However, using a Group Policy with this setting disabled may cause it to persist in a domain upgraded from Windows 2003 or earlier. Additionally, any user who has not changed a password since the setting was enabled still has an LM hash in the user's account if the password is LM compatible.

To ensure that your Active Directory and SAM databases no longer stores LM hash values, do the following:

1. Ensure this setting is enabled in the Default Domain Policy: **Network security: Do not store LAN Manager hash value on next password change in the group policy**.
2. Ensure that all users change their passwords.

For more information about this Group Policy Object (GPO), see Network security: Do not store LAN Manager hash value on next password change.

> **Note:** Some older applications, operating systems and services may still rely on LM hashes to be present for authentication, so we recommend testing this change before implementing it. Testing for incompatibility can typically be accomplished by configuring an account with a password or passphrase that is more than 15 characters long. This prevents storage of the LM hash for the account, which you can use to test applications for compatibility.

Analysis of other potential mitigations

This section discusses other commonly proposed mitigations that do not directly provide a meaningful mitigation of credential theft and reuse. Nonetheless, these may have other positive security or operational impacts on an Active Directory domain environment.

Disable the NTLM protocol

Restricting NTLM completely in an environment mitigates PtH attacks and offers added security benefits. However, this does not qualify as a mitigation that we recommend because it cannot be easily implemented by most organizations and it does not mitigate theft and reuse of Kerberos tickets or passwords.

The requirements for most organizations to restrict and effectively disable NTLM include at a minimum the following tasks:

- Extensive discovery analysis for incompatible devices and applications.
- Discovery of non-Windows operating system dependencies (if applicable).
- Planning, testing, and implementing changes to address all discovered compatibility issues (potentially including hardware and software replacements).
- Ensuring that all Kerberos prerequisites are completely met and configured for all applications and services in the environment.

Even with extensive NTLM restrictions in the environment that mitigate PtH attacks, attackers may still be able to steal and reuse other credentials including Kerberos TGTs and plaintext passwords. While this does not constitute a proposed mitigation, users are still encouraged to implement Kerberos if possible as Microsoft does not plan to enhance the NTLM protocol.

For more information about how to restrict NTLM, see the Auditing and restricting NTLM usage guide.

Smart cards and multifactor authentication

Multifactor authentication methods, such as smartcards, can greatly enhance the strength of the proof of the user's identity if the host is secure, but these methods do not provide immunity from credential theft attacks. While multiple factors are required for initial logon, the Windows operating system communicates with other domain computers using standard Kerberos and NTLM authentication protocols that exchange single factor authenticators, as required by the protocol standards when accessing network resources. When a computer in the domain is compromised and a user logs on to it with multifactor authentication, these single-factor secondary authenticators may be stolen from LSASS process memory, and reused in exactly the same way as the user logged on with a password.

> **Note:** If the account is enabled for smartcard use and still has a valid password, the NT hash in LSASS process memory is the hash of the user's password. If the account has been configured with the attribute **Smart Card required for interactive logon**, then the NT hash is a random value calculated when that attribute was enabled for the account. This password hash is provided to the client computer during the smartcard logons process by the domain controller. This password hash that is automatically generated when the attribute is set does not change. For more information, see [MS-PAC]: Privilege Attribute Certificate Data Structure.

Another factor to consider is that multifactor authentication is typically only available for interactive logons, including local logons (Interactive) and Remote Desktop Protocol (RDP, RemoteInteractive) logons, so the account attribute can only enforce smartcard multifactor authentication on those types of logons.

Jump servers

Jump servers are special purpose computers typically used for administrative access to isolated or segmented networks. Jump servers consolidate administrative tools and activities, and organizations can use them to restrict access to different security zones.

While jump servers can provide utility in security architecture, they do not directly mitigate credential theft and reuse attacks. Security integrity cannot be maintained if a user connects to an administrative jump server from a lower trust workstation. If the host connecting to a jump server is already sufficiently trusted, the jump server does not provide additional security. Jump servers can provide value as part of a more comprehensive security architecture. For example, using Jump servers as part of a strategy for monitoring unauthorized activity. If administrators are required by policy to perform all administrative tasks from jump servers, authentication not originating from jump servers would be immediately suspicious.

Rebooting workstations and servers

Rebooting computers after privileged administrators log off may have a positive mitigating effect prior to a PtH attack. Rebooting computers after use is the only way to ensure that credentials from stale or leaked logon sessions are removed from memory.

This is useful to limit risk in the event an attacker later compromises a running computer, but rebooting is not a recommendation in this document, because it has no meaningful effect on an already compromised computer. Attackers can capture credentials as soon as a logon has succeeded, and the process of capturing credentials can easily be automated. For these reasons, limiting the duration the logon session or any potential lingering stale session will have a limited effect on preventing a PtH attack.

Additional technical information

This part of the document contains additional technical information related to Pass-the-Hash (PtH) attacks and other credential theft attacks. While this information is not required to understand the impact of PtH attacks or how to implement the recommended mitigations, it provides additional details that may answer common questions, and background information about PtH attacks and other credential theft reuse attacks.

Trust levels and credential theft

A trusted computer or system (for example, a domain controller) should not depend on a lower trust computer, such as a workstation with Internet access, for its security. This section describes practical implications derived from this important principle that are focused on credential theft and reuse attacks.

An administrator is effectively entrusted with the security of any computer they control. Because any account that has administrative access to a computer can be used to steal the credentials of logged on or stored accounts, administrators must not log on to a computer administered by lower trust accounts and that could be potentially compromised.

One implication of this principle is that an administrator who logs on to a lower trust computer with higher-trust administrative credentials effectively creates a privilege escalation for that lower trust administrator. For example, an account in the Domain Admins group used to log on to a standard workstation is entrusting the security of the domain to that workstation and its security.

Another implication of this principle is that it is not possible gain security by connecting to a higher trust computer from a lower trust computer. For example, if you log on to a workstation as a standard user and then connect to a domain controller as a domain administrator using Remote Desktop Services (RDS) or some other means, you may have compromised the security of the domain. At this point, the domain administrator credentials have been typed into a keyboard that is under the control of the local workstation, which could be compromised.

Credential theft and reuse attacks exploit weaknesses in an organization's trust model and operational practices. Ensuring that Active Directory security architecture and administrative practices are designed with this in mind will greatly increase an organization's resilience to this class of credential theft and reuse attacks.

Other credential theft attacks

We have discussed attacks that rely on capturing and passing credentials already stored on a compromised computer without manipulating these credentials. There are also a number of other attack techniques not yet discussed in this paper in great detail, but that are worth mentioning in this section because they can potentially expose credentials to attackers or enhance their ability to steal credentials.

Compromised computers or inadvertent user actions can allow an attacker to steal plaintext passwords using the following attack techniques:

- **Keystroke loggers**: These are malicious applications that capture credentials while they are typed by the user to submit them to attackers.
- **Stored passwords:** Passwords stored by applications installed on the operating system can be obtained by an attacker.
- **Brute force attacks:** Attackers can apply computing resources to try to crack captured password hashes to obtain plaintext passwords.
- **Man-in-the-middle attacks**: This is a broad attack classification that can allow an attacker to intercept communication and capture credentials from network traffic. NTLM Relay attacks are an example of a Man-in-the-middle that may be addressed through Extended Protection for Authentication.
- **Local Security Authority Subsystem (LSASS):** These are passwords stored on the local computer that can be reversed to plaintext using available attack tools.

These types of attack introduce similar threats to the organization because they may allow attackers to obtain plaintext passwords which can be used during interactive logons.

Social engineering attacks originating from compromised computers should also be recognized as significant threats. Attackers may be able to send phishing email as a legitimate user or lure privileged users into authenticating to a compromised computer and exposing privileged credentials are another significant risk.

Password hashes can also be stolen if an attacker can gain physical access to the computer's hard drive. Accessing the hard drive of a domain member workstation or server can allow an attacker to steal the credentials of the stored local accounts. Accessing a domain controller's hard drive also allows an attacker to steal the password hashes for all accounts in the domain, including those of domain administrators.

An attacker can gain access to a hard drive if they obtain access to:

- The physical computer.
- Virtual disk files (VHD, VHDX, VMDK) for virtual hosts stored on a Virtual Host Hard Drive, Storage Area Network (SAN) device, or backup drive/tape.
- The backup files of physical or virtual servers or workstations.
- Backup applications where the server backups can be restored to a system under the attacker's control.
- Access to Remote Control through hardware features or remote Keyboard/Video/Mouse (KVM) device can provide the physical equivalent of access to a server.

An attacker can directly steal data from the computer using these means or they can use the access they gain to steal the NT hashes stored in the local SAM database or service account passwords. The hashes or service account passwords can also be used to attack the compromised computer when online to steal more credential information. All these attack techniques enhance the ability of the attacker to capture some form of credential that can be used for lateral movement or privilege escalation.

Kerberos Pass the Ticket attacks

We have not observed Kerberos attacks as frequently as PtH attacks, but proof-of-concepts and tools dedicated to them have already been published. This type of attack is referred to as a *Pass the Ticket attack*, and it resembles a PtH attack in its execution steps. As with a PtH attack, this type of credential theft and reuse attack requires the attacker to obtain local administrative access to capture the stored Ticket Granting Tickets (TGTs) before they can reused with the Kerberos protocol.

A Kerberos TGT and the associated session key together comprise a reusable credential for the Kerberos protocol. TGTs have a default lifespan of about 10 hours, and a default total lifetime of 7 days, if that TGT is repeatedly renewed before it expires. Attackers can steal TGTs and associated session keys and request a new session ticket at will until the renewal lifetime is reached.

When smartcards are used for authentication and the TGT has expired, users must insert their smart cards and then type their corresponding PINs. Otherwise, the TGT is renewed automatically using the same credentials for single sign-on (SSO) authentication. Kerberos attacks are currently less popular than attacks on NTLM, but they are equally possible if the attacker has compromised a computer and obtained local administrator access.

A significant difference in the attack value between NT hashes used in NTLM authentication and TGTs, is that password hashes are reusable until the user's password changes, while TGTs expire in a matter of hours according to their lifetime.

While Kerberos authentication is vulnerable to a similar attack, it is not likely to displace PtH attacks until NTLM becomes unavailable in organizations targeted by attackers. Unless the use of NTLM is explicitly disabled, password hashes are still created and stored in the LSASS process memory, and they are valid for authentication. NTLM also remains the most commonly used authentication protocol, because of the current level of NTLM support and compatibility with existing devices and software. For a discussion of this potential mitigation, see the "Disable NTLM" section.

Kerberos delegation

One additional risk of Kerberos authentication may arise if sensitive domain accounts are trusted for delegation. If the particular service or server being authenticated to is trusted for unconstrained delegation, the client sends a TGT and session key to the server. An attacker that has compromised the target computer can impersonate clients with that TGT.

You can mitigate this particular delegation risk by doing the following:

- Enable the setting **Account is sensitive and cannot be delegated** attribute on all privileged accounts to protect them from this attack.
- Use constrained delegation to set limits on which accounts can be impersonated by which service.

For more information about delegation mitigation, review the section "Task 4: Disable the account delegation right for privileged accounts" in "Mitigation 1: Restrict and protect high privileged domain accounts" of Appendix A, "Step-by-step instructions to mitigate PtH attacks."

For more information about Kerberos constrained delegation, see How to Configure the Server to be Trusted for Delegation.

For information about additional features in Windows Server 2012 to further constrain delegation, see What's New in Kerberos Authentication.

Windows authentication protocols and credential types

Windows supports a number of different types of credentials and authentication protocols, depending on the operating system version and configuration.

Windows authentication protocols

The following table provides information on Windows authentication protocols and a brief description of each supported protocol.

Table3. Windows Authentication Protocols

Protocol	Description
Kerberos	Kerberos is the default and preferred authentication protocol for domain authentication on current Windows operating systems. Kerberos relies on a system of keys, tickets, and mutual authentication in which keys are normally not passed across the network. (Direct use of the key is permitted for some application clients under certain circumstances). While a full description of the Kerberos authentication protocol is outside the scope of this document, certain Kerberos-specific objects that are used in the authentication process are stored as LSA secrets in memory, such as Ticket Granting Tickets (TGT) and Service Tickets (ST). TGTs are Single sign-on (SSO) authentication credentials that can be reused for lateral movement or privilege escalation, while STs are not credentials that can be used for lateral movement or privilege escalation. For more information about Kerberos authentication, see the Kerberos Authentication Technical Reference.

Protocol	Description
NTLM	NTLM protocols are authentication protocols that use a challenge and response method to make clients mathematically prove that they have possession of the NT hash. Current and past versions of Windows support multiple versions of this protocol, including NTLMv2, NTLM, and the LM authentication protocol. **Note:** All current versions of NTLM are vulnerable to relay attacks without the software upgrades required to enable Extended Protection for authentication. How to best configure the **LMCompatibilityLevel** setting that controls protocol version negotiation and resulting compatibility issues has been the subject of a significant amount of security guidance over the past decade and this is not addressed in detail in this document. For a recommended reference on the technical details involving this subject, see the Security Watch article, "The Most Misunderstood Windows Security Setting of All Time."
Digest	Digest is a standards-based protocol typically used for HTTP and Lightweight Directory Access Protocol (LDAP) authentication. Digest authentication is described in RFCs 2617 and 2831.The current implementation of digest authentication in Windows was introduced in Windows XP and Server 2003. For more information about digest authentication, see the Digest Authentication Technical Reference and Store passwords using reversible encryption

Windows authentication

This section includes background information about Windows authentication as it relates to credential theft and reuse attacks.

Terminology: authentication, credentials, and authenticators

This section defines some terminology that appears throughout the document. When a user wants to access a computing resource, they must provide information that identifies who they are, their *identity*, and proof of this identity in the form of secret information that only they are supposed to know. This proof of identity is called an *authenticator*. An authenticator can take various forms, depending on the authentication protocol and method. The combination of an identity and an authenticator is called an *authentication credential* or *credential*.

The process of creation, submission, and verification of credentials is described simply as *authentication*, which is implemented through various authentication protocols, such as NTLM and Kerberos authentication. Authentication establishes the identity of the user, but not necessarily the user's permission to access or change a computing resource, which is handled by a separate *authorization* process.

Credentials in Windows operating systems

Credentials are typically created or converted to a form required by the authentication protocols available on a computer. Credentials may be stored in LSASS process memory for use by the account during a session. Credentials must also be stored on disk in authoritative databases, such as the SAM database and the Active Directory database.

> **Note:** Some authentication protocols present secret information in its original form, such as protocols that can transmit a user name and password in plaintext. These authentication protocols are inherently insecure, are not used by default settings in Windows, and should not be used unless they are encapsulated within another protocol that provides session security, such as SSL or TLS.

Identities – usernames

In Windows operating systems, a user’s identity takes the form of the account’s username, either the "user name" (SAM Account Name) or the User Principal Name (UPN).

Windows authenticators

Table 4, "Windows Credential Types," lists the credential authenticator types in Windows operating systems and provides a brief description of each type.

Table 4. Windows Credential Types

Credential type	Description
Plaintext credentials	When a user logs on to a Windows computer and provides a username and credentials, such as a password or PIN, the information is provided to the computer in plaintext. This plaintext password is used to authenticate the user's identity by converting it into the form required by the authentication protocol. Current versions of Windows also retain an encrypted copy of this password that can be decrypted back to plaintext for use with authentication methods such as Digest authentication. **Note:** Windows operating systems never store any plaintext credentials in memory or on disk, only reversibly encrypted credentials. When later access to the plaintext forms of the credentials are required, Windows stores the passwords in encrypted form that can only be decrypted by the operating system to provide access in authorized circumstances. These protections cannot prevent an attacker with SYSTEM level access from illicitly extracting and decrypting them in the same manner that the operating system would for legitimate use.

Credential type	Description
NT hash	The NT hash of the password is calculated using an unsalted MD4 hash algorithm. MD4 is a cryptographic one-way function that produces a mathematical representation of a password. This hashing function is designed to always produce the same result from the same password input, and to minimize collisions where two different passwords can produce the same result. This hash is always the same length and cannot be directly decrypted to reveal the plaintext password. Because the NT hash only changes when the password changes, an NT hash is valid for authentication until a user's password is changed. This also means that if two accounts use an identical password, they will also have an identical NT password hash. To protect against brute force attacks on the NT hashes or the online systems, users who authenticate with passwords should set strong passwords or passphrases that include characters from multiple sets that are as long as your users can easily remember. For tips and guidance on helping your users set longer passwords, see Selecting Secure Passwords. **Note:** The use of unsalted MD4 may be seen as a hashing weakness, but it has very little impact on risk as the hash value is managed and protected equivalent to a plaintext password.
LM hash	LAN Manager (LM) hashes are derived from the user password. Legacy support for LM hashes and the LAN Manager authentication protocol remains in the Windows NTLM protocol suite, but default configurations and Microsoft security guidance have discouraged their use for more than a decade. LM hashes have a number of challenges that make them less secure and more valuable to attackers if stolen: • LM hashes required a password to be less than 15 characters long and contain only ASCII characters. • LM Hashes also do not differentiate between uppercase and lowercase letters. Techniques to obtain the plaintext value from a LM hash with relatively low effort have been available for a number of years, so the loss of a LM hash should be considered nearly equivalent to the loss of plaintext password.

Credential type	Description
Windows logon cached password verifiers	These verifiers are stored in the registry (HKLM\Security) on the local computer and provide validation of a domain user's credentials when the computer cannot connect to Active Directory during a user logon. These are not credentials, as they cannot be presented to another computer for authentication, and they can only be used to locally verify a credential. These password verifiers are resistant from brute force attack techniques through the use of a resource intensive validation process. They are also protected against rainbow table attacks through the use of *salt values* included during their calculation. These verifiers are not discussed further in this document as they cannot be used for credential theft attacks.

Table 5, "Credential Storage," lists the types of credential storage locations available on the Windows operating system.

Table 5. Credential Storage

Credential sources	Description
Security Accounts Manager (SAM) database	The SAM database is stored as a file on the local disk, and is the authoritative credential store for local accounts on each Windows computer. This database contains all the credentials that are local to that specific computer including the built-in local Administrator account and any other local accounts for that computer. The SAM database stores information on each account, including the username and the NT password hash. By default, the SAM database does not store LM hashes on current versions of Windows. It is important to note that no password is ever stored in a SAM database, only the password hashes.

Credential sources	Description
Local System Security Authority Subsystem (LSASS) process memory	The Local Security Authority (LSA) stores credentials in memory on behalf of users with active Windows sessions. This allows users to seamlessly access network resources, such as file shares, Exchange mailboxes, and SharePoint sites, without re-entering their credentials for each remote service. LSA may store credentials in multiple forms including: • Reversibly encrypted plaintext • Kerberos tickets (TGTs, service tickets) • NT hash • LM hash If the user logs on to Windows using a smartcard, LSA will not store a plaintext password, but it will store the corresponding NT hash value for the account and the plaintext PIN for the smartcard. If the account attribute for smartcard required for interactive logon is enabled, a random NT hash value is automatically generated for the account instead of the original password hash. This password hash that is automatically generated when the attribute is set does not change. If a user logs onto Windows with a password that is compatible with LM hashes, this authenticator will be present in memory. The storage of plaintext credentials in memory cannot be disabled in current versions of the Windows operating system, even if the credential providers that require them are disabled. The credentials stored are directly associated with the LSA logon sessions that have been started since the last reboot and have not been closed. For example, LSA sessions with stored LSA credentials are created when a user or service account does any of the following: • Logs on to a local session or RDP session on the computer. • Runs a process using the **RunAs** option. • Runs an active Windows service on the computer. • Runs a scheduled task or batch job. • Runs a process on the local computer using a remote admin tool, such as PSExec –u –p.

Credential sources	Description
LSA secrets on disk	A Local Security Authority (LSA) secret is a secret piece of data that is accessible only to SYSTEM account processes. Some of these secrets are credentials that must persist after reboot and are stored in encrypted form on disk. Credentials stored as LSA secrets on disk may include: • Account password for the computer's Active Directory account. • Account passwords for Windows services configured on the computer. • Account passwords for configured scheduled tasks. • Account passwords for IIS application pools and websites. • An attack tool running as an account with administrative privileges on the computer can exploit those privileges to extract these LSA secrets.

Credential sources	Description
Domain Active Directory Database (NTDS.DIT)	The Active Directory database is the authoritative store of credentials for all user and computer accounts in an Active Directory domain. Each writable domain controller in the domain contains a full copy of the domain's Active Directory database, including account credentials for all accounts in the domain. Read-only domain controllers (RODCs) house a partial local replica with credentials for a selected subset of the accounts in the domain. By default, RODCs do not have a copy of privileged domain accounts. The Active Directory database stores a number of attributes for each account, including both username types and the following: • NT hash for current password. • NT hashes for password history (if configured). NT hash values are also retained in Active Directory for previous passwords to enforce password history during password change operations. The number of password history NT hash values retained is equal to the number of passwords configured in the password history enforcement policy. LM hashes may also be stored in the Active Directory database depending on the domain controller operating system version, configuration settings, and password change frequency. For more information, see the section "Remove LM hashes from Active Directory" under "Additional recommendations."
Credential Manager (CredMan) store	Users may choose to save passwords in Windows using an application or through the Credential Manager Control Panel applet. These credentials are stored on disk and protected using the Data Protection Application Programming Interface (DPAPI), which encrypts them with a key derived from the user's password. Any program running as that user will be able to access credentials in this store. For more information about DPAPI, see Windows Data Protection.

Logon type definition

In Windows-based computers, all authentications are processed as one of several logon types, regardless of which authentication protocol or authenticator is used. The most common logon types and their attributes relative to credential theft are documented in Table 7, "Connection Methods and Where the Credentials Are Created and Cached."

Table 6. Logon types

Logon type	#	Authenticators accepted	Reusable credentials in LSA session	Examples

Interactive (a.k.a., Logon locally)	2	Password,
		Smartcard,
		other
			Yes	Console logon;
				RUNAS;
				Hardware remote control solutions (such as Network KVM or Remote Access / Lights-Out Card in server)
				IIS Basic Authn (before IIS 6.0)

Network	3	Password,
		NT Hash,
		Kerberos ticket
			No (except if delegation is enabled, then Kerberos tickets present)	NET USE;
				RPC calls;
				Remote registry;
				IIS integrated Windows authn;
				SQL Windows authn;

Batch	4	Password (usually stored as LSA secret)	Yes	Scheduled tasks

Logon type	#	Authenticators accepted	Reusable credentials in LSA session	Examples
Service	5	Password (usually stored as LSA secret)	Yes	Windows services
NetworkCleartext	8	Password	Yes	IIS Basic Authn (IIS 6.0 and newer); Windows PowerShell with CredSSP
NewCredentials	9	Password	Yes	RUNAS /NETWORK
RemoteInteractive	10	Password, Smartcard, other	Yes	Remote Desktop (formerly known as “Terminal Services”)

For more information about Logon Types, see SECURITY_LOGON_TYPE enumeration.

The following list provides definitions for the columns for logon types in Table 4, "Windows Credential Types":

- **Logon type** is the type of logon requested.
- **#** is the numeric identifier for the logon type that is reported in audit events in the Security event log.
- **Authenticators accepted** indicates which types of authenticators are able to initiate a logon of this type.
- **Reusable credentials in LSA session** indicates whether the logon type results in the LSA session holding credentials, such as plaintext passwords, NT hashes, or Kerberos tickets that could be used to authenticate to other network resources.
- **Examples** list common scenarios in which the logon type is used.

Common administrative tasks and remote credential exposure

Performing administration of remote computers with domain accounts can introduce credential theft risks that are difficult to mitigate with straightforward technical controls. Because of this, we have included Table 7, "Connection Methods and Where the Credentials Are Created and Cached," to describe the credential exposure risk from common administrative tools and methods.

This section does not address credential theft risks on the “source” computer, only on the “destination” computer that is being remotely administered.

> **Important:** As described in the section "Trust levels and credential theft," a workstation used to manage servers must have at least the same trust level as the managed servers.

The column headings in Table 7 are defined as follows:

- **Logon type** identifies the logon type initiated by the connection.
- **Reusable credentials on destination** indicates that the following credential types will be stored in LSASS process memory on the destination computer where the specified account is logged on locally:
  - NT hash.
  - Kerberos TGTs.
  - Plaintext password (if applicable).
  - LM hash (if applicable).

The symbols in Table 7 are defined as follows:

(**-**) denotes when credentials are not exposed.

(**√**) denotes when credentials are exposed.

Table 7. Connection Methods and Where the Credentials Are Created and Cached

Connection method	Logon type	Reusable credentials on destination	Comments
Log on at console	Interactive	√	Includes hardware remote access / lights-out cards and network KVMs.
RUNAS	Interactive	√	
RUNAS /NETWORK	NewCredentials	√	Clones current LSA session for local access, but uses new credentials when connecting to network resources.
Remote Desktop (success)	RemoteInteractive	√	If the remote desktop client is configured to share local devices and resources, those may be compromised as well.
Remote Desktop (failure - logon type was denied)	RemoteInteractive	-	By default, if RDP logon fails credentials are only stored very briefly.
Net use * \\SERVER	Network	-	
Net use * \\SERVER /u:user	Network	-	
MMC snap-ins to remote computer	Network	-	Example: Computer Management, Event Viewer, Device Manager, Services

Connection method	Logon type	Reusable credentials on destination	Comments
PowerShell WinRM	Network	-	Example: Enter-PSSession server
PowerShell WinRM with CredSSP	NetworkClearText	√	New-PSSession server -Authentication Credssp -Credential *cred*
PsExec without explicit creds	Network	-	Example: PsExec \\server cmd
PsExec with explicit creds	Network + Interactive	√	PsExec \\server -u *user* -p *pwd* cmd Creates multiple logon sessions.
Remote Registry	Network	-	
Remote Desktop Gateway	Network	-	Authenticating to Remote Desktop Gateway.
Scheduled task	Batch	√	Password will also be saved as LSA secret on disk.
Run tools as a service	Service	√	Password will also be saved as LSA secret on disk.
Vulnerability scanners	Network	-	Most scanners default to using network logons, though some vendors may implement non-network logons and introduce more credential theft risk.

Connection method	Logon type	Reusable credentials on destination	Comments
Web Authentication			
IIS "Basic Authentication"	NetworkCleartext (IIS 6.0+) Interactive (prior to IIS 6.0)	√	
IIS "Integrated Windows Authentication"	Network	-	NTLM and Kerberos Providers.

For management applications that are not in this table, you can determine the logon type from the logon type field in the audit logon events. For more information, see Audit logon events.

Summary

The Pass-the-Hash (PtH) attack is one specific example of a credential theft and reuse attack, but other authentication credentials may also be stolen and reused in a similar manner. Any credential stored in memory or on disk can be harvested by an attacker with local administrator or SYSTEM access for authentication using techniques that are similar to those of a PtH attack.

Organizations should design mitigation plans and defenses to address the entire class of credential theft and reuse attacks rather than any single form. We have provided mitigations that are effective, practical, and robust if an organization implements them collectively in a strategic plan to prevent lateral movement and privilege escalation as described in this document.

This document also provided technical details as background information and guidance to help our customers address the risk of PtH attacks and other credential theft attacks. We are continuing to investigate the problem of credential theft and reuse to increase the security of Windows operating systems and to ensure that attackers find our platform significantly more difficult to compromise.

Microsoft is committed to improving security, privacy and reliability around the world through software innovation. Microsoft is committed to delivering the security, privacy and reliability that helps customers feel confident in their computing experience[2]

[2] http://www.microsoft.com/about/twc/en/us/security.aspx

Appendix A: Step-by-step instructions to mitigate PtH attacks

This appendix includes step-by-step instructions for the following mitigations that we recommend organizations use to help reduce the risk of Pass-the-Hash (PtH) attacks:

- Mitigation 1: Restrict and protect high privileged domain accounts.
- Mitigation 2: Restrict and protect local accounts with administrative privileges.
- Mitigation 3: Restrict inbound traffic using Windows Firewall.

Mitigation 1: Restrict and protect high privileged domain accounts

Domain administrator and other highly privileged accounts should be restricted so that they can only be used to log on to management systems and workstations that are secured at the same level as the managed systems.

While this multipart mitigation is robust and effective, it may be challenging to fully implement in all domain environments. Minimum, better, and ideal implementations are noted for each part of the mitigation where integration challenges are expected. As with all significant changes to a production environment, we recommend testing these changes thoroughly before implementing and deploying them, and then staging the deployment in a manner that allows for rollback of the changes in case of technical issues.

Implementing this mitigation is separated into the following tasks:

Task 1: Separate administrative accounts from user accounts for administrative personnel and create dedicated accounts for specific administrative tasks.

Task 2: Create dedicated administrative workstation hosts for administrators.

Task 3: Restrict Domain Administrator accounts and other sensitive accounts so that they cannot be used to log on to lower trust servers and workstations.

> **Note:** This task is especially important to implement for this mitigation.

Task 4: Disable the account delegation right for privileged accounts.

Task 1: Separate administrative accounts from user accounts for administrative personnel

Allocate separate accounts for personnel who require highly privileged accounts to perform administrative tasks and standard user tasks according to the following guidelines:

- ***Standard user account*** – Grant standard user privileges for standard user tasks, such as email, web browsing, and using line-of-business (LOB) applications. These accounts should not be granted administrative rights.
- ***Privileged account*** – Allocate these accounts for performing the following administrative duties:
    - **Minimum allocation** – Create separate accounts for domain administrators, enterprise administrators, or the equivalent with higher privileges on the domain or forest. Accounts granted these rights should not be used to administer anything except domain data and domain controllers.
    - **Better allocation** – We also recommend creating separate accounts with lesser administrative rights, such as accounts for workstation administrators, server administrators, and accounts with privileges over designated Active Directory organizational units (OUs).
    - **Ideal allocation** – Create separate accounts for personnel with multiple job responsibilities who are required to log on to systems with significantly different trust levels (workstations, servers, domain controllers) for each level of privilege (workstation administration, server administration, domain administration).

> **Important**: Ensure that privileged accounts cannot be used to access email or browse the Internet.

Task 2: Create specific administrative workstation hosts for administrators

Often, administrators don't have easy physical access to servers and require the ability to manage systems with high privileges from their normal workstation. A workstation connected to the Internet with email and web browsing access will be regularly exposed to compromise through phishing attacks, drive-by download attacks, and other Internet risks.

> **Note:** If the administrators in your environment can log on locally to managed servers and perform all tasks without elevated privileges or domain privileges from their workstations, you can skip this part of the mitigation.

Because of these threats, we recommend setting up new workstations for the administrators in Task 1 that are dedicated to administration duties and that do not have Internet or email access. The lateral movement mitigations (Mitigation 2 and Mitigation 3) should apply to these administrative workstations as well.

- **Minimum** – Build new workstations and configure Internet access restrictions.
- **Ideal 1** – Also, do not grant administrators membership in the local administrative group on the computer in order to limit the ability to bypass protections.
- **Ideal 2** – Restrict workstations from network connectivity to anything except domain controllers and servers that the administrative accounts can be used to manage.
- **Ideal 3** – Use AppLocker to restrict all applications from running except the operating system and approved administrative tools and applications. For more information about AppLocker, see the AppLocker Technical Overview.

Minimum – Build new workstations and deny Internet access

This section describes how to build dedicated administrative workstations and block Internet access on those workstations.

You can block Internet access in a number of ways that include the following:

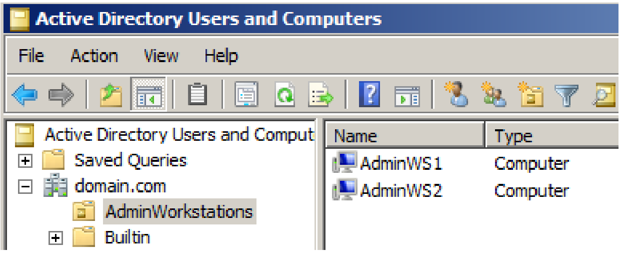
- Configure authenticating boundary proxy services (if deployed) to disallow privileged accounts from accessing the Internet.
- Configure boundary firewall or proxy services to disallow Internet access for the IP addresses assigned to these workstations.
- Block outbound access to the boundary proxy servers in Windows Firewall.

The following instructions describe how to block Internet access by creating a Group Policy Object (GPO) that configures an invalid proxy address on the administrative workstations. These instructions are only effective on computers running Internet Explorer and other windows components that use these proxy settings.

> **Note:** These instructions assume that the workstations will be dedicated to domain administrators. You can create additional OUs to manage less privileged administrators using these instructions by simply modifying which administrators you want to allow to log on locally (See Step 8 in the following procedure).

To install the administrative workstations in a domain and block Internet access with Group Policy proxy settings

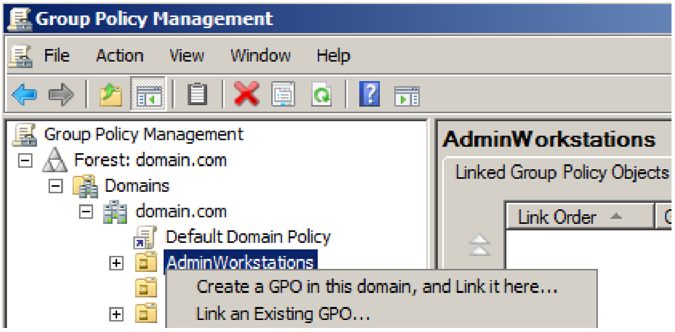
1. As a domain administrator on a domain controller, open Active Directory Users and Computers, and create a new OU for Administrative workstations.
2. Create computer accounts for the new workstations.


Active Directory Users and Computers
File
Action
View
Help
Active Directory Users and Comput
Saved Queries
domain.com
AdminWorkstations
Builtin

Name	Type
AdminWS1	Computer
AdminWS2	Computer

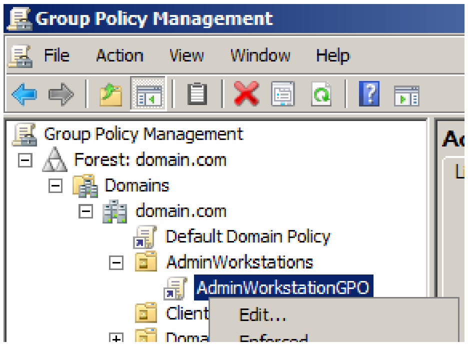
> **Note:** You may need to delegate permissions to join the domain using KB 932455 if the account joining the workstations to the domain does not already have permissions to join computers to the domain.

3. Close Active Directory Users and Computers.
4. Open the Group Policy Management Console (GPMC).
5. Right-click the new OU and select **Create a GPO in this domain, and Link it here…** as indicated in the following figure.



6. Name the GPO and click **OK**.

7. Expand the GPO, right-click the new GPO, and then click **Edit** as indicated in the following figure.

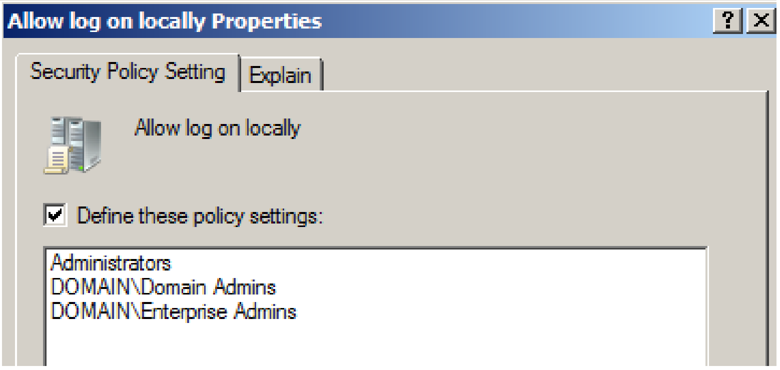


8. Configure which accounts may log on locally to these administrator workstations by doing the following:

   a. Navigate to Computer Configuration\Policies\Windows Settings\Local Policies and then click **User Rights Assignment**.
   b. Double-click **Allow log on locally** and select **Define these policy settings**.
   c. Click **Add User or Group...**, click **Browse**, type **Enterprise Admins**, and then click **OK**.
   d. Click **Add User or Group...**, click **Browse**, type **Domain Admins**, and then click **OK**.

   > **Note:** These instructions assume that the workstations will be dedicated to domain administrators.

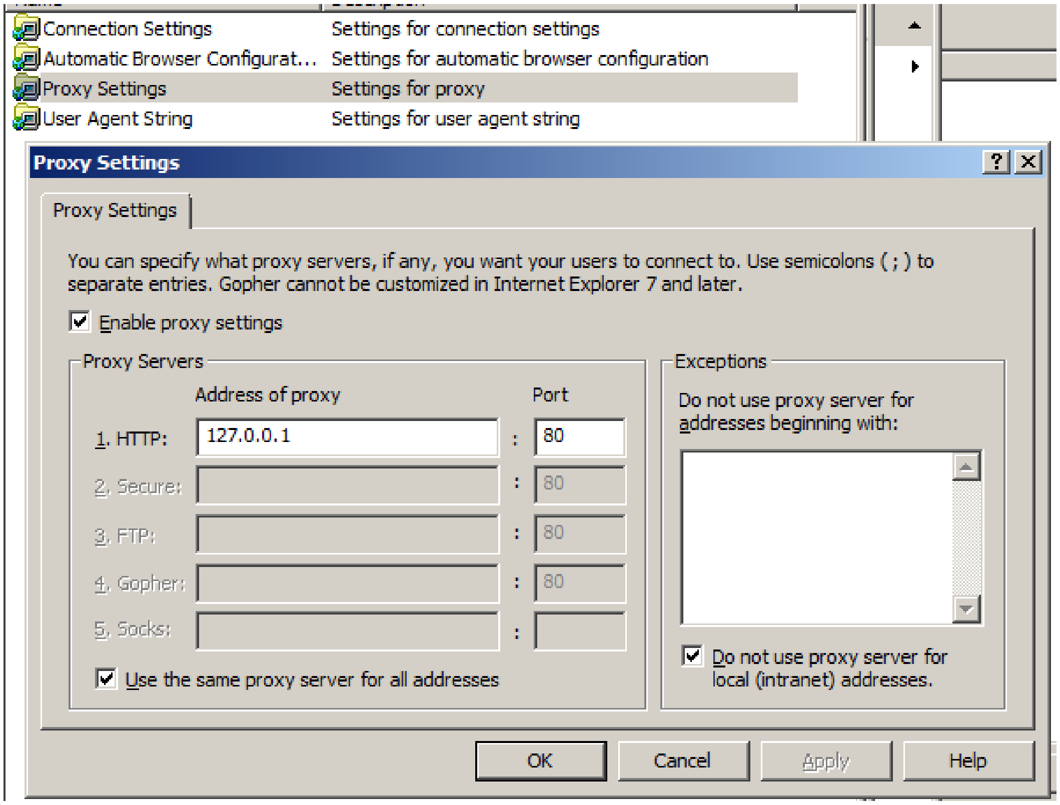
   e. Click **Add User or Group...**, type **Administrators**, and then click **OK**.



☑ Define these policy settings:

Administrators
DOMAIN\Domain Admins
DOMAIN\Enterprise Admins

9. Configure the proxy configuration
   a. Navigate to User Configuration\Policies\Windows Settings\Internet Explorer and then click **Connection**.
   b. Double-click **Proxy Settings**, select **Enable proxy settings**, type **127.0.0.1** (the network Loopback IP address) as the proxy address, and then click **OK**.



Connection Settings	Settings for connection settings
Automatic Browser Configurat…	Settings for automatic browser configuration
Proxy Settings	Settings for proxy
User Agent String	Settings for user agent string

Proxy Settings

You can specify what proxy servers, if any, you want your users to connect to. Use semicolons (;) to separate entries. Gopher cannot be customized in Internet Explorer 7 and later.

☑ Enable proxy settings

Proxy Servers

	Address of proxy	Port
1. HTTP:	127.0.0.1	80
2. Secure:		80
3. FTP:		80
4. Gopher:		80
5. Socks:		

☑ Use the same proxy server for all addresses

Exceptions

Do not use proxy server for addresses beginning with:

☑ Do not use proxy server for local (intranet) addresses.

10. Configure the loopback processing mode so that the user Group Policy proxy setting will apply to all users on the computer by doing the following:
    a. Navigate to Computer Configuration\Policies\Administrative Templates\System and then click **Group Policy**.
    b. Double-click **User Group Policy loopback policy processing mode** and select **Enabled**.
    c. Select **Merge Mode** and click **OK**.

11. Configure software updates by doing the following:
    a. Navigate to Computer Configuration\Administrative Templates\Windows Components, and then click **Windows Update**.
    b. Configure the Windows Update settings in the following table.

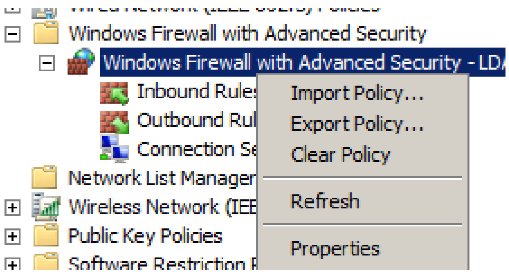
Table 8. Windows Update Configuration Settings

Windows Update setting	Configuration
Allow Automatic Updates immediate installation	Enabled
Configure Automatic Updates	Enabled • 4 – Auto download and schedule the install • 0 – Every Day 03:00
Enabling Windows Update Power Management to automatically wake up the system to install scheduled updates	Enabled
Specify intranet Microsoft update service location	Enabled • http://<WSUSServername> • http://<WSUSServername> Where <WSUSServername> is the DNS name or IP address of the WSUS server in the environment.
Automatic Updates detection frequency	6 Hours
Re-prompt for restart with scheduled installations	1 minute
Delay Restart for scheduled installations	5 minutes

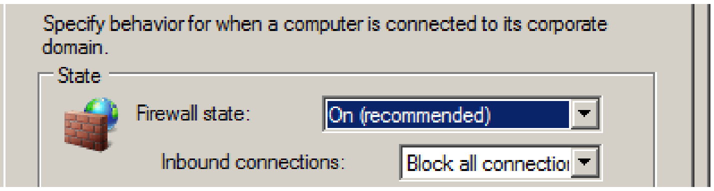
> **Note:** This step assumes that Windows Server Update Services (WSUS) is installed and configured in the environment. You can skip this step if you use another tool to deploy software updates. Also, if only the public Microsoft Windows Update service on the Internet is used, then these administrative workstations will no longer receive updates.

12. Configure the inbound firewall to block all connections by doing the following:

    a. Right-click **Windows Firewall with Advanced Security – LDAP://path** and select **Properties** as indicated in the following figure.



    b. On each profile, ensure that the firewall is enabled and that inbound connections are set to **Block all connections** as indicated in the following figure.


Specify behavior for when a computer is connected to its corporate domain.

State

Firewall state: On (recommended)

Inbound connections: Block all connectio

    c. Click **OK** to complete the configuration.

13. Close the GPMC.
14. Install the Windows operating systems on the workstations, give them the same names as the computer accounts for them, and then join them to the domain.

Task 3: Restrict server and workstation logon access

This section describes how to restrict administrators from using high privileged administrator accounts to log on to lower trust workstations. This restriction prevents administrators from inadvertently increasing the risk to credential theft by logging on to a lower trust computer.

> **Important:** Before starting this procedure, ensure that you either have local access to the domain controller or that you have completed building at least one administrative workstation.

As with the other tasks of this mitigation, there are minimum, better, and ideal levels of implementation as described in the following subsections:

- **Minimum** – Restrict domain administrators from workstations.
- **Better** – Also restrict domain administrators from non-domain controller servers.
- **Ideal** – Also restrict server administrators from logging on to workstations.

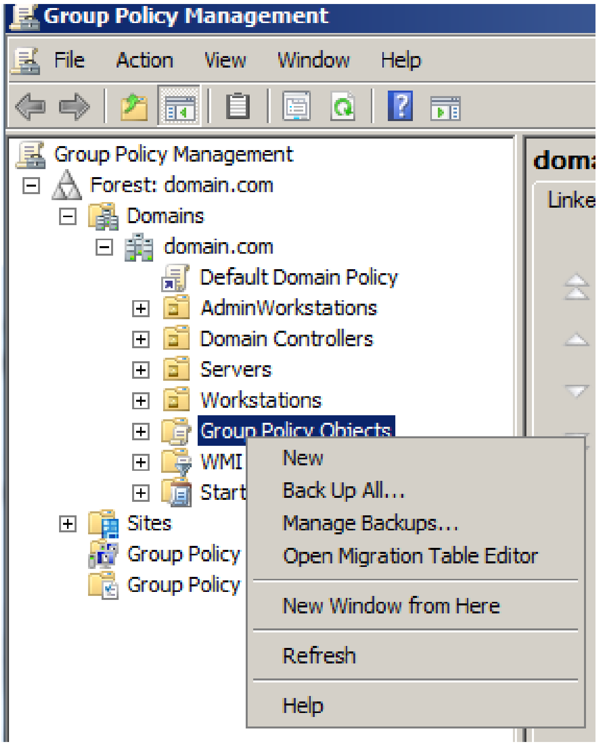
Minimum – Restrict domain admins from workstations

Before starting this process, ensure to identify all OUs in the domain that contain workstations and servers. Any computers in OUs that are missed will not restrict administrators with highly privileged accounts from logging on to them.

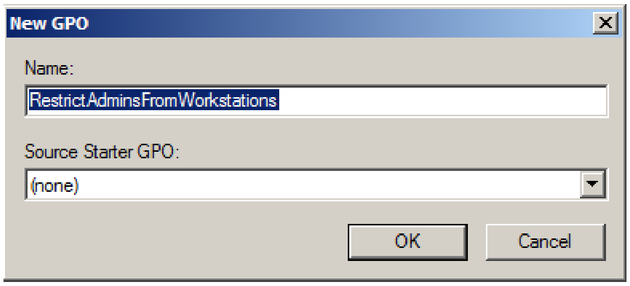
> **Note:** Don't link this to the OU containing the administrative workstations that you created using the mitigation instructions for Task 2: Create specific administrative workstation hosts for administrators.

To restrict domain admins from workstations

1. As a domain administrator, open the **Group Policy Management Console (GPMC)**.
2. Open **Group Policy Management** and expand ***Forest*\Domains\*Domain***, and then expand **Group Policy Objects**.
3. Right-click **Group Policy Objects** and click **New** as indicated in the following figure.



4. In the **New GPO** dialog box, name the GPO that will restrict administrators from logging on to workstations, and then click **OK**.



New GPO

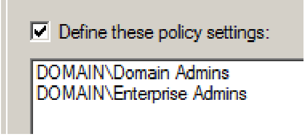
Name: RestrictAdminsFromWorkstations

Source Starter GPO: (none)

5. Right-click the new GPO and select **Edit…**
6. Configure user rights to deny log on locally for domain administrators.

   Navigate to Computer Configuration\Policies\Windows Settings\Local Policies, click **User Rights Assignment**, and then do the following:

   a. Double-click **Deny log on locally** and select **Define these policy settings**.
   b. Click **Add User or Group**…, click **Browse**, type **Enterprise Admins**, and click **OK**.
   c. Click **Add User or Group**…, click **Browse**, type **Domain Admins**, and click **OK**.



☑ Define these policy settings:

DOMAIN\Domain Admins
DOMAIN\Enterprise Admins

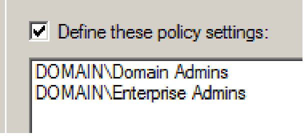
   d. **Optional**: Add any groups that contain server administrators who shouldn't log on to workstations.
   e. Click **OK** to complete the configuration.

7. Configure the user rights to deny batch and service logon rights for domain administrators by completing the following substeps:

   > **Note:** Completing this step may cause issues with administrative tasks that run as scheduled tasks or services with accounts in the domain admins group. The practice of using domain administrator accounts to run services and tasks on workstations creates a significant risk of credential theft attacks and therefore should be replaced with alternative means to run scheduled tasks or services.

   a. Double-click **Deny log on as a batch job** and select **Define these policy settings**.
   b. Click **Add User or Group…**, click **Browse**, type **Enterprise Admins**, and then click **OK**.

c. Click **Add User or Group**..., click **Browse**, type **Domain Admins**, and then click **OK**.



☑ Define these policy settings:

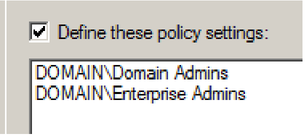
DOMAIN\Domain Admins
DOMAIN\Enterprise Admins

Optional: Add any groups that contain server administrators who shouldn't log onto workstations.

d. Double-click **Deny log on as a service and select Define these policy settings**.

e. Click **Add User or Group...**, click **Browse**, type **Enterprise Admins**, and then click **OK**.

f. Click **Add User or Group...**, click **Browse**, type **Domain Admins**, and then click **OK**.



☑ Define these policy settings:

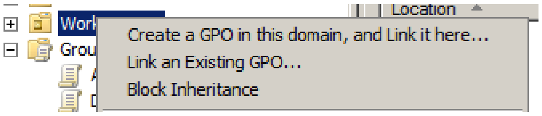
DOMAIN\Domain Admins
DOMAIN\Enterprise Admins

Optional: Add any groups that contain server administrators who shouldn't log onto workstations.

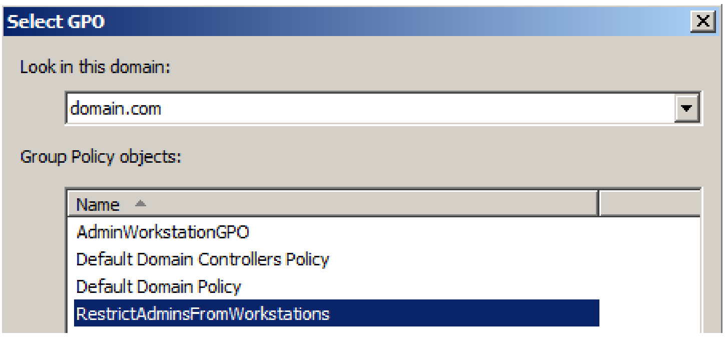
8. Link the GPO to the first Workstations OU.

Navigate to the <*Forest*>\Domains\<*Domain*>\OU Path, and then:

a. Right-click the workstation OU and select **Link an Existing GPO**...



b. Select the GPO that you just created and click **OK**.



Select GPO

Look in this domain:

domain.com

Group Policy objects:

Name
AdminWorkstationGPO
Default Domain Controllers Policy
Default Domain Policy
RestrictAdminsFromWorkstations

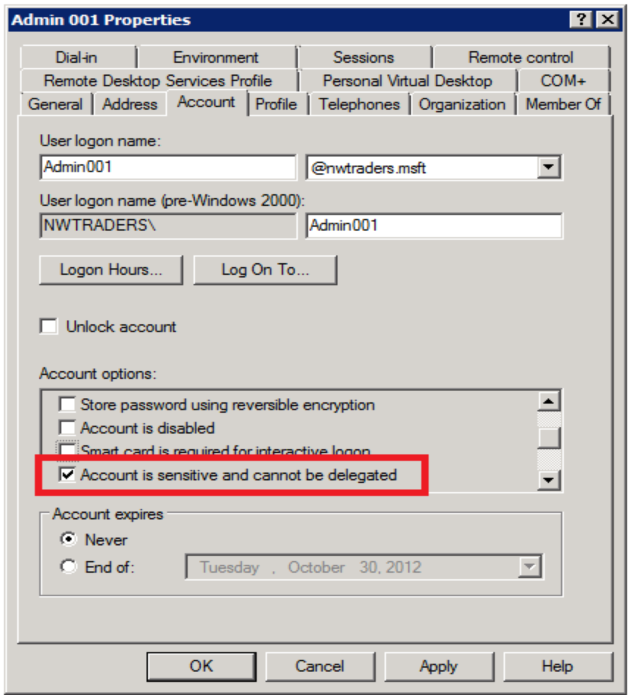
9. Test the functionality of enterprise applications on workstations in the first OU and resolve any issues caused by the new policy.
10. Link all other OUs that contain workstations. However, do not create a link to the Administrative Workstation OU that you created in Task 2.

> **Important:** If you later extend this solution, do not deny logon rights for the **Domain Users** group. The **Domain Users** group includes all user accounts in the domain, including users, domain administrators, and enterprise administrators.

Task 4: Disable the account delegation right for privileged accounts

Although no user accounts are marked for delegation by default, accounts in an Active Directory domain can be trusted for delegation. This means that a service or a computer trusted for delegation can impersonate an account that authenticates to them to access other resources across the network.

For privileged accounts, such as those belonging to members of the Administrators, Domain Admins, or Enterprise Admins groups in Active Directory, delegation can present a substantial risk of privilege escalation. For example, if an account in the Domain Admins group is used to log on to a compromised member server that is trusted for delegation, that server can request access to resources in the context of the Domain Administrator account, and escalate the compromise of that member server into a domain compromise. We recommend configuring the user objects for all highly-privileged accounts in Active Directory by enabling the **Account is sensitive and cannot be delegated** account option so that they cannot be delegated. As with any configuration change, ensure to test this enabled setting prior to implementation.



Admin 001 Properties

User logon name: Admin001 @nwtraders.msft

User logon name (pre-Windows 2000): NWTRADERS\ Admin001

□ Unlock account

Account options:

□ Store password using reversible encryption
□ Account is disabled
□ Smart card is required for interactive logon
☑ Account is sensitive and cannot be delegated

Account expires

● Never
○ End of: Tuesday , October 30, 2012

Figure 3. Account is sensitive and cannot be delegated account option

Mitigation 2: Restrict and protect local accounts with administrative privileges

The mitigation approaches in this section have a similar effect on preventing lateral movement using stolen credentials from local accounts. Task 1: Enforce local account restrictions for remote access, and Task 2: Deny network logon to all local accounts, focus on logon restrictions for the local accounts. Task 3: Create unique passwords for local privileged accounts, uses a password randomization approach. Each approach should prevent an attacker from using a password or password hash stolen from one local computer to authenticate on another computer with administrative rights.

We recommend implementing Task 1 first because it is simple to deploy, and then following up with planning to implement Task 2 and Task 3 as soon as possible. Randomizing passwords (Task 3) also addresses other credential theft and reuse attacks, as it prevents local Administrator accounts from using identical passwords.

> **Note:** This mitigation does not apply if all administrative local accounts are disabled.

The following table summarizes the settings in this mitigation.

Table 9. Summary of Mitigation 2 Settings

Task 1: Enforce local account restrictions for remote access (Windows Vista and later Windows operating systems)		
1	Policy location	**Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options**
	Policy name	**User Account Control: Run all administrators in Admin Approval Mode**
	Policy setting	**Enabled**
2	Policy location	**Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options**
	Policy name	**User Account Control: Admin Approval Mode for the Built-in Administrator account**
	Policy setting	**Enabled**
3	Registry key	**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System**
	Registry value name	**LocalAccountTokenFilterPolicy**
	Registry value type	**DWORD**
	Registry value data	**0**

Task 2: Deny network logon to all local accounts		
1	Policy location	**Computer Configuration\Windows Settings\ Security Settings\Local Policies\User Rights Assignment**
	Policy name	**Deny access to this computer from the network**
	Policy setting	**Username of the built-in Administrator account** (May be renamed through policy.)
2	Policy location	**Computer Configuration\Windows Settings\ Security Settings\Local Policies\User Rights Assignment**
	Policy name	**Deny log on through Remote Desktop Services** (Windows Server 2008 R2 and later.) **Deny log on through Terminal Services** (Windows Server 2008 and earlier.)
	Policy setting	**Username of the built-in Administrator account** (May be renamed through policy.)

Task 1: Enforce local account restrictions for remote access (Windows Vista and later Windows operating systems)

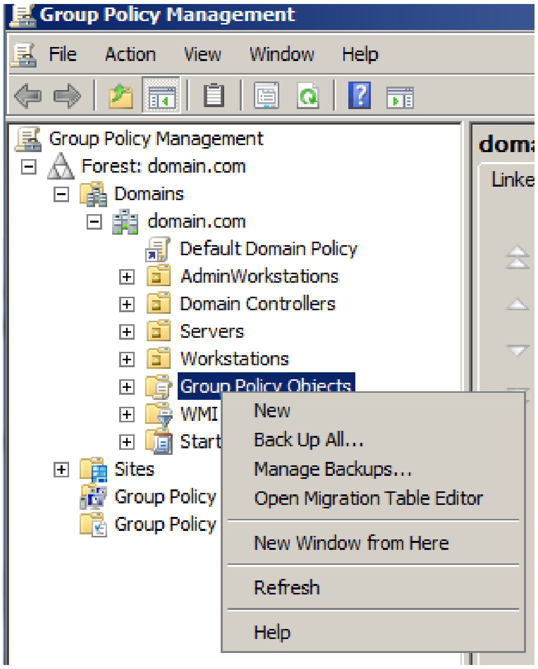
User Account Control (UAC) in all Windows operating systems starting with the release of Windows Vista makes it possible for a privileged account to be treated as a standard user "non-admin" account until full rights ("elevation") is requested and approved. A default feature of UAC is that when a local account logs on from a remote computer using Network logon (for example, by using "NET.EXE USE"), it is issued a standard user token with no administrative rights, and no ability to request or receive elevation. Consequently, local accounts that log on using Network logon cannot access administrative shares such as C$ or ADMIN$ or perform any other remote administration. To ensure that these restrictions are applied, use the following procedure to enforce them through Group Policy.

> **Note:** Perform the following steps using an account that is a member of the Domain Admins group.

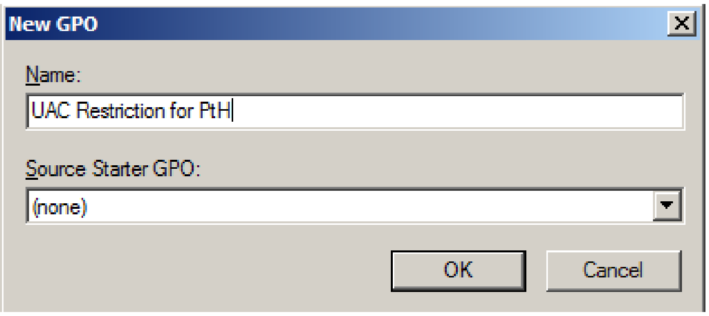
To enforce local account restrictions for remote access

1. Start the **Group Policy Management** Console (GPMC).
2. In the console tree, expand <*Forest*>\Domains\<*Domain*>, and then **Group Policy Objects** (where *forest* is the name of the forest, and *domain* is the name of the domain where you want to set the Group Policy).

3. In the console tree, right-click **Group Policy Objects**, and select **New** as indicated in the following figure.



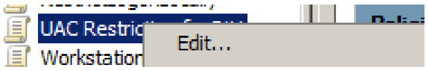
4. In the **New GPO** dialog box, type <***gpo_name***>, and then click **OK** (where *gpo_name* is the name of the new GPO that should indicate it is being used to restrict the local administrator privileges from being carried over to another computer.


New GPO

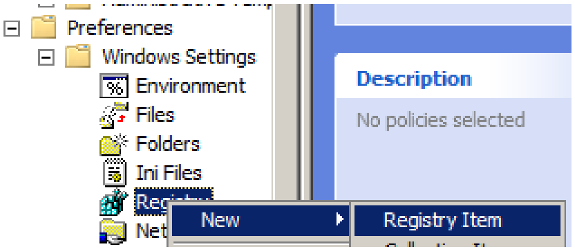
Name:
UAC Restriction for PtH

Source Starter GPO:
(none)

5. In the details pane, right-click <***gpo_name***>, and click **Edit** as indicated in the following figure.



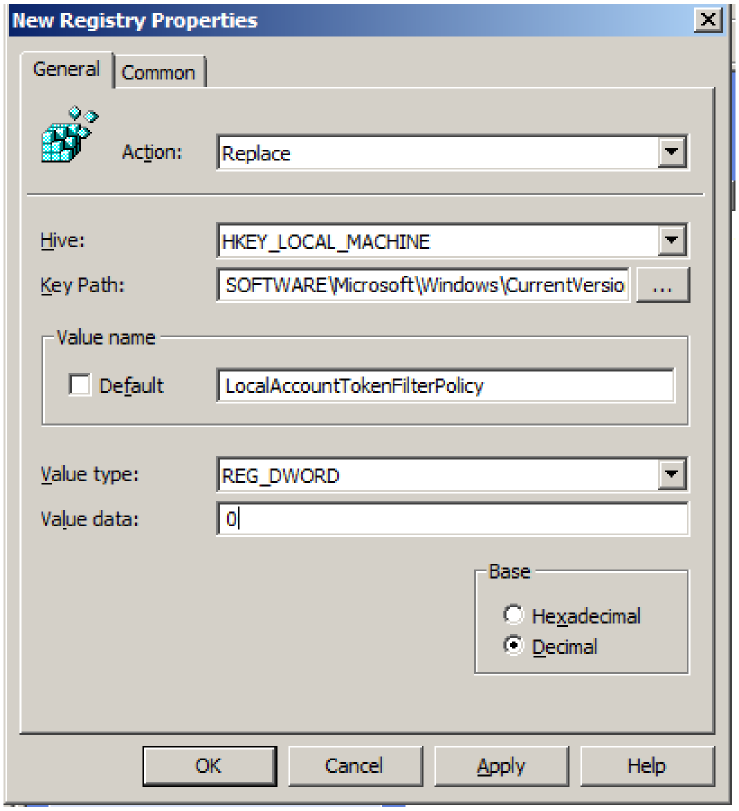
6. Ensure that UAC is enabled and that UAC restrictions apply to the built-in Administrator account by doing the following:
   a. Navigate to Computer Configuration\Policies\Windows Settings and Local Policies, and then click **Security Options**.
   b. Double-click **User Account Control: Run all administrators in Admin Approval Mode**, select **Enabled**, and then click **OK**.
   c. Double-click **User Account Control: Admin Approval Mode for the Built-in Administrator account**, select **Enabled**, and then click **OK**.
7. Ensure that the local account restrictions are applied to network interfaces by doing the following:
   a. Navigate to Computer Configuration\Preferences and Windows Settings, and then click **Registry.**
   b. Right-click **Registry** and then click **New > Registry Item** as indicated in the following figure.



   c. In the **New Registry Properties dialog box**, on the **General tab**, change the setting in the **Action:** box to **Replace** as indicated in the following figure.
   d. Ensure the **Hive:** box is set to **HKEY_LOCAL_MACHINE**.
   e. Click (**…**), browse to the following location for **Key Path:**, and click **Select** for:

      SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
   f. In the **Value name** area, type **LocalAccountTokenFilterPolicy**.
   g. In the **Value type:** box, expand the drop-down box to change the value to **REG_DWORD**.
   h. In the **Value data:** box, ensure that the value is set to **0**.

i. Verify this configuration and click **OK**.



Action: Replace

Hive: HKEY_LOCAL_MACHINE

Key Path: SOFTWARE\Microsoft\Windows\CurrentVersio

Value name

☐ Default LocalAccountTokenFilterPolicy

Value type: REG_DWORD

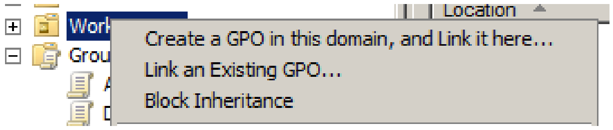
Value data: 0

Base

○ Hexadecimal

● Decimal

8. Link the GPO to the first **Workstations** OU by doing the following:
   a. Navigate to the <*Forest*>\Domains\<*Domain*>\OU path.
   b. Right-click the **Workstation** OU and select **Link an Existing GPO…**



   c. Select the GPO that you just created and click **OK**.
9. Test the functionality of enterprise applications on the workstations in this first OU and resolve any issues caused by the new policy.
10. Create links to all other OUs that contain workstations.
11. Create links to all other OUs that contain servers.

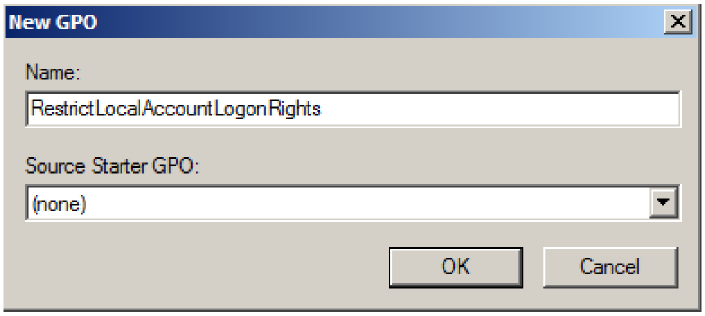
Task 2: Deny network logon to all local accounts

Denying local accounts the ability to perform network logons can help prevent a local account password hash from being reused in a PtH attack. This mitigation helps prevent lateral movement, and helps ensure that credentials stolen from a compromised operating system cannot be used to compromise additional computers with the same local account passwords.

> **Note:** Before performing this task, you will need to identify the name of the local, built-in Administrator account (if not the default "Administrator") and any other accounts that are members of the local Administrators group.
>
> Perform the following steps using an account that is a member of the Domain Admins group.

To deny network logon to all local administrative accounts

1. Start the **Group Policy Management** Console (GPMC).
2. In the console tree, expand <*Forest*>\Domains\<*Domain*>, and then **Group Policy Objects** (where *forest* is the name of the forest and *domain* is the name of the domain where you want to set the Group Policy).
3. In the console tree, right-click **Group Policy Objects**, and select **New**.
4. In the **New GPO** dialog box, type <***gpo_name***>, and then click **OK** (where *gpo_name* is the name of the new GPO that should indicate it is being used to restrict administrative local accounts from interactively logging on to the computers).



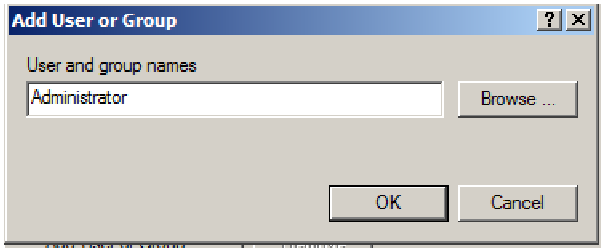
RestrictLocalAccountLogonRights

(none)

5. In the details pane, right-click <***gpo_name***>, and click **Edit.**

6. Configure the user rights to deny network logons for administrative local accounts by doing the following:
   a. Navigate to Computer Configuration\Policies\Windows Settings and Local Policies, and then click **User Rights Assignment**.
   b. Double-click **Deny access to this computer from the network** and select **Define these policy settings**.

c. Click **Add User or Group…**, type the username of the built-in administrator account, and then click **OK**. (The default name is **Administrator** on US English installations but may be renamed by policy or manually.)



Add User or Group

User and group names: Administrator

Important: In the **User and group names** box, type only the user name of the account that you identified prior to starting this process. Do not click **Browse** and do not type the domain name or the local computer name in this dialog box. For example, type only **Administrator**. If the text that you type in this dialog box resolves to a name that is underlined, includes a computer name, or includes the domain, it will restrict the wrong account and cause this mitigation to work incorrectly. Also, be careful not to enter the group name "Administrators", as this will block domain accounts in that group as well.

d. For any additional local accounts in the Administrators group on all of the workstations that you are configuring, click **Add User or Group…** type the usernames of these accounts in the dialog box in the same manner as in the previous step, and then click **OK**.

7. Configure the user rights to deny Remote Desktop (RemoteInteractive) logons for administrative local accounts by doing the following:

   a. Navigate to Computer Configuration\Policies\Windows Settings and Local Policies, and then click **User Rights Assignment**.

   Note: Depending on the Windows operating system, you can choose the name of the RemoteInteractive logon right.

   b. On computers running Windows Server 2008 and earlier, double-click **Deny log on through Terminal Services** and select **Define these policy settings**.

   c. On computers running Windows Server 2008 R2 and later, double-click **Deny log on through Remote Desktop Services** and select **Define these policy settings**.

d. Click **Add User or Group…**, type the username of the built-in administrator account and click **OK**. (The default name is Administrator on US English installations but may be renamed by policy or manually.)

> **Important:** In the **User and group names** box, type only the user name of the account that you identified prior to starting this process. Do not click **Browse** and do not type the domain name or the local computer name in this dialog box. For example, type only **Administrator**. If the text is underlined, includes a computer name, or includes the domain name, this it will restrict the wrong account and cause this mitigation not to work correctly. Also, be careful not to enter the group name "Administrators", as this will block domain accounts in that group as well.

e. For any additional local accounts in the Administrators group on all of the workstations that you are configuring, click **Add User or Group…** type the usernames of these accounts in the dialog box in the same manner as in the previous step, and then click **OK**.

8. Link the GPO to the first **Workstations** OU by doing the following:
   a. Navigate to the <*Forest*>\Domains\<*Domain*>\OU path.
   b. Right-click the **Workstation** OU and select **Link an existing GPO…**
   c. Select the GPO that you just created and click **OK**.
9. Test the functionality of enterprise applications on the workstations in that first OU and resolve any issues caused by the new policy.
10. Create links to all other OUs that contain workstations.
11. Create links to all other OUs that contain servers.

> **Note:** You may need to create a separate GPO if the usernames of the built-in administrator accounts are different on the workstations and servers.

Task 3: Create unique passwords for privileged local accounts

Passwords should be unique per individual account. While this is generally true for individual user accounts, many enterprises have identical passwords for common local accounts, such as the built-in Administrator account. This is usually the case with deployed operating system images or other scenarios where the organization chooses to use the same passwords for local accounts during operating system deployments.

Passwords that are left unchanged or are changed synchronously to keep them identical add a significant risk for organizations. Randomizing the passwords mitigates PtH attacks using local accounts by hampering the ability of attackers to use password hashes of those accounts to compromise other computers.

Randomizing passwords can be done by:

- Purchasing and implementing an enterprise tool to accomplish this task. These tools are commonly referred to as Privileged Password Management tools.
- Configure, customize and implement a free tool to accomplish this task. A sample tool with source code is available at Solution for management of built-in Administrator account's password via GPO.

> **Note:** This tool is free and is not supported by Microsoft. There are some important considerations to make prior to deploying this tool because it requires client side extensions and schema extensions to support password generation and storage.

- Create and implement a custom script or solution to randomize local account passwords

Mitigation 3: Restrict inbound traffic using the Windows Firewall

Workstations can use Windows Firewall to restrict inbound traffic to specific services, servers, and trusted workstations used for desktop management. Applications that do not directly accept authentication credentials may also be allowed through the Windows firewall without incurring the risks of credential theft and reuse.An organization can do this by denying all inbound access unless explicitly specified by a rule. However, because servers are typically designed to accept inbound connections to provide services, this mitigation is not typically feasible on server operating systems.

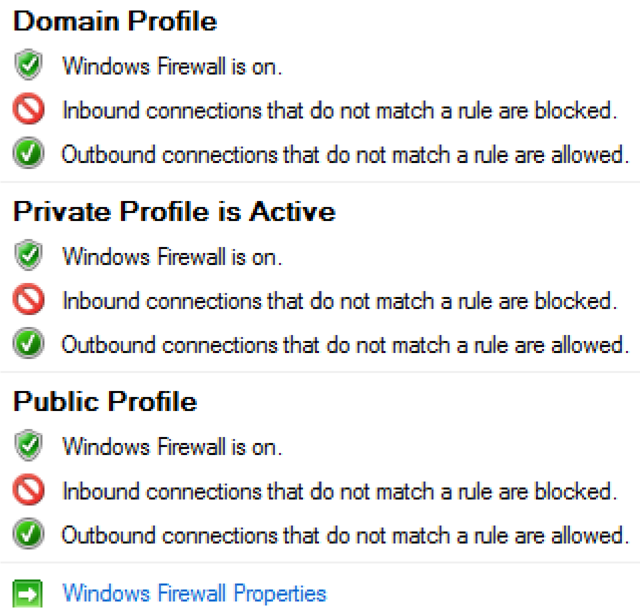
Nonetheless, using Windows Firewall to restrict inbound traffic is a very simple and robust mitigation that you can use to prevent captured hashes from being used for lateral movement or privilege escalation. This mitigation significantly reduces the attack surface of the organization's network resources to a PtH attack and other credential theft attacks by disabling an attacker’s ability to authenticate from any given host on a network using any type of stolen credentials.

Because your organization may have configured firewall rules that are different than the default rules, this mitigation is not universal and fully prescriptive.

> **Note:** We advise caution when updating or rolling out new firewall rules and testing is strongly encouraged to prevent outages or connectivity issues with applications that depend on inbound connections to client computers. *Do not follow these step-by-step instructions if your organization is using another host firewall instead of Windows Firewall*. However, the concepts described here can be implemented using other host firewalls.

The recommended strategy to follow for this mitigation is to:

1. Block all inbound traffic, and then use rules to only allow inbound traffic by exception. In Windows Firewall, you can use the **Block (Default)** setting to configure all profiles that appears in the snap-in as indicated in the following figure.



Domain Profile

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Private Profile is Active

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Public Profile

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Windows Firewall Properties

Figure 4. The Block (Default) setting in Windows Firewall

2. Enable inbound exceptions only for authorized hosts that your organization uses to manage workstations and for specific applications that do not directly accept Windows credentials for authentication. Applications on workstations that include services which accept inbound connections from multiple hosts should be reviewed for the risk of credential theft and reuse attacks prior to approving exceptions.

   To enable authorized inbound rules, administrators are required to define specific exceptions that are allowed through the firewall defined by specific programs, IP addresses, subnets, ports, and protocols.

   > **Note:** Most management software today, including Microsoft System Center Configuration Manager, use agents running locally on the client computers in the organization that connect with the management server to receive policy and software updates over the network. These pull operations by the client computers do not require an inbound firewall exception.

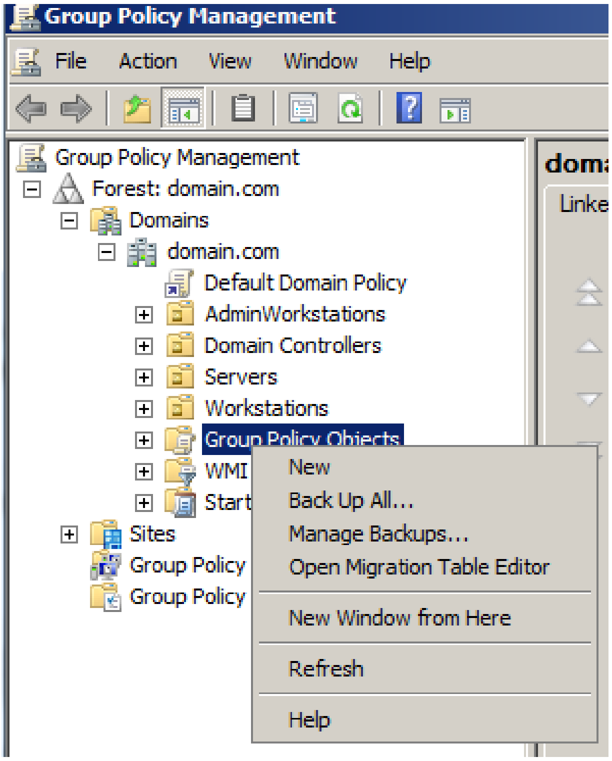
3. Review your organization's firewall configuration to ensure that no previously configured inbound rules bypass the specific connections defined above in item 2, and introduce unnecessary risks.

Using a GPO to set up Windows Firewall rules

Configure a Group Policy Object to block inbound connections that do not match a rule, create an allow rule for management servers, and identify whether any other inbound rules can allow inbound authenticated connections. The following subsections provide instructions on how to accomplish these tasks.

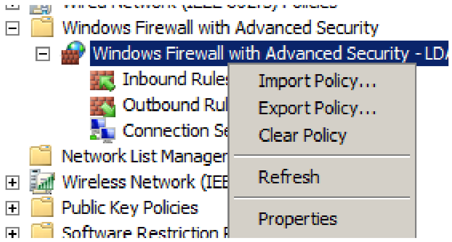
Part A – Enable and configure Windows Firewall inbound policy

1. As a domain administrator, open the Group Policy Management Console.
2. Expand the **Group Policy Management** node, expand <***Forest***>, **Domains**, <***Domain***>, and then **Group Policy Objects**.
3. Right-click **Group Policy Objects** and click **New as indicated in the following figure**.

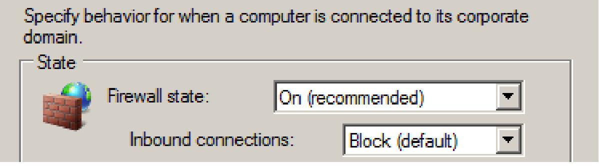


4. Name the new Group Policy Object (GPO) that you will use to configure Workstation Firewall settings.
5. Right-click the new GPO and select **Edit…**
6. Navigate to **Computer configuration\Windows Settings\Security Settings**, and then expand **Windows Firewall with Advanced Security**.

7. Right-click **Windows Firewall with Advanced Security – LDAP://<*path*>** and select **Properties** as indicated in the following figure.



8. On each profile, ensure that the firewall is enabled and that inbound connections are set to **Block (default)** as indicated in the following figure.



Specify behavior for when a computer is connected to its corporate domain.

State

Firewall state: On (recommended)

Inbound connections: Block (default)

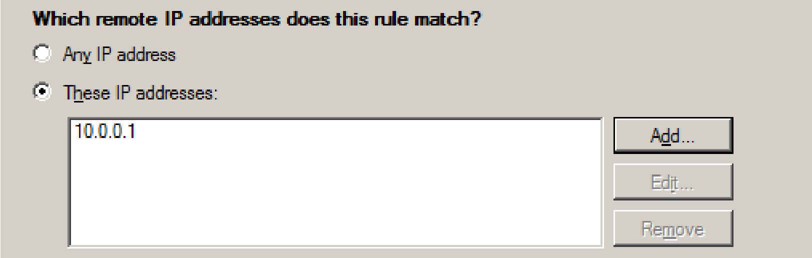
9. Select **Settings**, and then under **Rule Merging**, select **No** to prevent local administrators from creating rules that can bypass incoming connection restrictions (allowing all incoming connections).

   > **Important**: Allowing firewall rules to merge will negate the effect of this mitigation.

10. Click **OK** to complete the configuration.

Part B – Configure an inbound exception for remote management hosts

1. As a domain administrator, open the Group Policy Management Console (GPMC), navigate to expand **Windows Firewall with Advanced Security**, and then expand **Windows Firewall with Advanced Security – LDAP://<*path*>**.
2. Right-click **Inbound Rules** and click **New**.
3. Select the Rule type **Custom** and click **Next**.
4. Select **All Programs** and click **Next**.
5. If you know the inbound ports for the management application, configure them at this location. Otherwise, click **Next** to allow all traffic from the management hosts.
6. On the **Scope** page, **Click Add** to enter the remote hosts that will be initiating network connections to these hosts, as indicated in the following example figure.



Which remote IP addresses does this rule match?

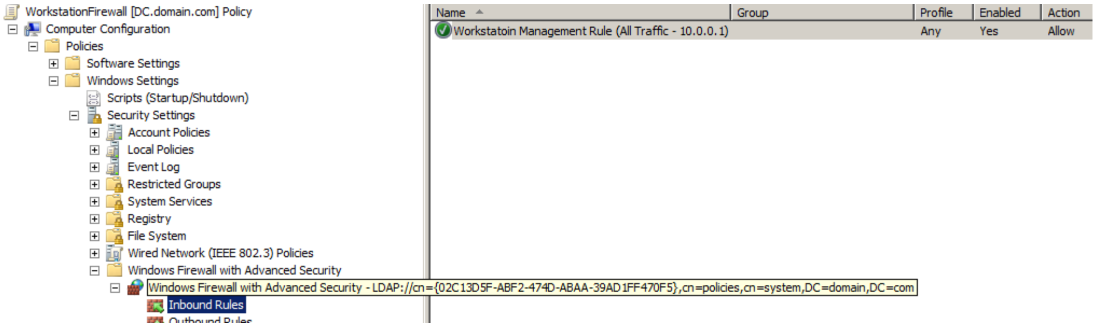
○ Any IP address

● These IP addresses:

10.0.0.1

> **Important:** Ensure to not use the **Any IP address** option. This option leaves this dialog box blank to allow any remote IP address, which will allow traffic from any host and defeat the purpose of this mitigation. Also, do not select any of the predefined sets of computers or specify a large IP address range or subnet that includes non-management computers.

7. Select **Allow the connection** and click **Next**.
8. Ensure that all profiles are selected (**Domain**, **Private**, and **Public**) and click **Next**.
9. Name the rule and create a description that includes which remote applications are allowed to connect through the firewall based on this rule, as indicated in the following example figure.

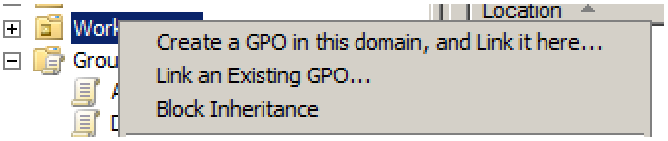


```
WorkstationFirewall [DC.domain.com] Policy
⊟ Computer Configuration
  ⊟ Policies
    ⊞ Software Settings
    ⊟ Windows Settings
        Scripts (Startup/Shutdown)
      ⊟ Security Settings
        ⊞ Account Policies
        ⊞ Local Policies
        ⊞ Event Log
        ⊞ Restricted Groups
        ⊞ System Services
        ⊞ Registry
        ⊞ File System
        ⊞ Wired Network (IEEE 802.3) Policies
        ⊟ Windows Firewall with Advanced Security
          ⊟ Windows Firewall with Advanced Security - LDAP://cn={02C13D5F-ABF2-474D-ABAA-39AD1FF470F5},cn=policies,cn=system,DC=domain,DC=com
              Inbound Rules
              Outbound Rules
```

Name	Group	Profile	Enabled	Action
✓ Workstatoin Management Rule (All Traffic - 10.0.0.1)		Any	Yes	Allow

Part C – Deploy the GPO

1. Link the GPO to the first Workstations OU by doing the following:
   a. Navigate to the <*Forest*>\Domains\<*Domain*>\OU Path.
   b. Right-click the **Workstation** OU and select **Link an existing GPO…** as indicated in the following figure.



   c. Select the GPO that you just created and click **OK**.
2. Test the functionality of management and other applications on the workstations in the first OU and resolve any issues caused by the new policy.
3. Create links to all other existing OUs that contain workstations.

For more information on configuring firewall rules, see Creating Rules that Allow Required Inbound Network Traffic.

Appendix B: Pass-the-Hash (PtH) attack FAQs

The following are frequently asked questions about Pass-the-Hash (PtH) attacks:

- **Does this problem only affect Windows operating systems?**

  No. This issue affects other platforms as well, not just Windows. Computers need to perform actions on behalf of the user, so some form of authentication credentials must be available to the operating system to avoid requiring the user to have to re-enter credentials every time a network task is performed.

- **Can Microsoft modify the code, or release an update to address the problem?**

  Not in the short term. While we are continually looking for enhancements to increase the security of the Windows operating system, this issue requires employing best practices and proper management of privileged accounts. Even if Microsoft releases a quick fix to prevent current tools from exploiting this issue, attackers will update their tools accordingly and customers would remain vulnerable. The employment of the mitigations proposed in this document provides a much more robust approach to protecting against these attacks.

- **Why isn't there a single solution to prevent these attacks?**

  There are two problems when it comes to credential theft and reuse as stated in the document: lateral movement and privilege escalation. Both require a combination of controls and best practices to contain an attacker that successfully exploits a single host. Multiple mitigations increase the organization's security posture, and provide more barriers that attackers must overcome during an attack.

- **Why is the Kerberos protocol not a proposed mitigation?**

  While we encourage customers to use the Kerberos protocol, certain services still rely on NTLM and addressing application compatibility can be challenging. In mixed environments, password hashes are still available to an attacker. The Kerberos protocol alone is also susceptible to similar attacks, such as Pass the Ticket (TGTs) attacks, and exploit tools to perform such attacks are already available to attackers. *Mitigation 1 Restrict and protect high privileged domain accounts* and Mitigation 3 *Restrict inbound traffic using the Windows Firewall* proposed in this document, also protect against Pass the Ticket attacks.

- **Will using smartcard logons mitigate the risk of the problem?**

  No, not significantly. While smartcard logons can enhance security to mitigate credential theft by removing the need for a user to know their account password, underlying password hashes and Kerberos tickets can still be stolen and re-used for network connections.

- **What is SSO and why is it supported?**

  Single sign-on (SSO) authentication is available on Windows and other platforms to allow the operating system to perform network tasks on behalf of the user. It is supported on most platforms to avoid prompting the user authentication information every time the computer needs to perform a task (for example checking email).

- **Will NTLM or LSASS be enhanced in future versions of Windows to protect the operating system from these attacks?**

  While we will continue enhancing the security of the Windows operating system, including the Local Security Authority Subsystem (LSASS), NTLM will not be enhanced, and we are encouraging our customers to deploy the Kerberos protocol.

- **How do I detect a PtH attack and credential theft in my domain?**

  Detecting these types of attacks is very difficult because the attacker activity cannot be easily differentiated from legitimate authentication. Stolen credentials allow attackers to use standard authentication mechanisms with valid credentials, creating audit logs that appear to be legitimate user activity.

  By using the recommended mitigations in this document to limit administrative activities, it may be easier to identify suspicious account usage or failed logon attempts. Suspicious account use patterns may be used in monitoring for malicious activity or investigating incidents.

- **Will my antimalware or HIDS solution be able to detect or stop a PtH attack?**
  Current known tools that enable PtH attacks are detected and blocked by most antivirus products, but these detections can be disabled by the attacker after a computer is compromised. Additionally, attackers can use techniques like binary packing to evade signature detection in some circumstances. Host Intrusion Detection Systems (HIDS) will not detect normal network authentication as an intrusion. PtH attacks look like normal network traffic with legitimate authentication credentials, so HIDS will not likely detect this seemingly normal activity as an intrusion without additional indicators of anomalous behavior. After authenticating with a privileged account, it is very likely that attackers will also disable HIDS.

- **Are Privileged Password Management tools and password vaults effective mitigations against this attack?**
  Yes, provided these tools are configured properly and that they provide the following:

  a. Password randomization that ensures unique passwords for all privileged local or domain accounts.

b. Timely rotation or check-in/check-out rotation of passwords.

These measures greatly increase password uniqueness in the environment and ensure that passwords are changed after a certain period of time. They also prevent credentials from remaining valid as a result of the password rotation.

As these solutions and their administrators have effective control over other administrative accounts in the domain, the configuration and operation of these solutions should follow security rigor similar to that applied to domain controllers and domain administrators.

Appendix C: Definitions

This white paper includes the following terms and definitions

- **authentication**: The process of creation, submission, and validation of credentials.
- **authentication credential**: The combination of an identity and an authenticator.
- **authenticator**: A data structure used by one party to prove to another party that it knows a secret. In the Kerberos authentication protocol, authenticators also include timestamps to prevent replay attacks, and are encrypted with the session key issued by the Key Distribution Center (KDC).
- **identity**: A person or entity that must be verified by means of authentication, based on criteria such as password or a certificate.
- **Pass-the-Hash (PtH) attack**: A technique in which an attacker captures account logon credentials (username and NT Hash) on one computer and then uses those captured credentials to authenticate to other computers over the network using the NTLM Protocol.
- **Pass the Ticket attack**: A credential theft and reuse attack that resembles a PtH attack in its execution steps, but involves the theft and re-use of a Ticket Granting Ticket (TGT) with the Kerberos protocol rather than an NT Hash value and the NTLM protocol.
- **password hash**: A direct one-way mathematical derivation of the password. The password hash for an account changes only when the user’s password changes.
- **privileged account**: A user account that has been granted administrative privileges to operating systems such as Domain Admin or Enterprise Admin that have full access to objects in an Active Directory domain, or an application (for example, Microsoft Exchange Server or SQL Server).
- **salt value:** Random or variable data that is sometimes included as part of a cryptographic operation. Salt values are added to increase the work required to mount a brute-force (dictionary) attack against encrypted or hashed data.
- **unsalted**: A cryptographic operation that does not include any salt values. Unsalted operations are subject to greater risk of brute-force attacks because the same input always results in the same output. If a variable input (even a piece of public data like a computer name or username) is included as a salt, dictionaries used for brute force attacks would need to include much more data to mount an effective attack.

Appendix D: References

This white paper includes the following references and resources:

- AppLocker Technical Overview.
- Audit logon events.
- Auditing and restricting NTLM usage guide.
- Description of User Account Control and remote restrictions in Windows Vista
- Determined Adversaries and Targeted Attacks.
- Digest Authentication Technical Reference.
- Extended Protection for Authentication.
- How to Configure the Server to be Trusted for Delegation.
- Kerberos Authentication Technical Reference.
- Microsoft Security Compliance Manager.
- [MS-PAC]: Privilege Attribute Certificate Data Structure.
- Network security: Do not store LAN Manager hash value on next password change.
- SECURITY_LOGON_TYPE enumeration.
- Selecting Secure Passwords.
- Solution for management of built-in Administrator account's password via GPO.
- Store passwords using reversible encryption
- The Most Misunderstood Windows Security Setting of All Time.
- User Account Control Technical Reference.
- What's New in Kerberos Authentication.
- Windows Data Protection.

Appendix E: Document Update

This section of the document describes updates done in this minor revision.

The following updates were made to the document:

- Clarified and addressed consistancy when referring to an “administrative local account” to reflect local accounts with administrative rights on workstations and servers throughout the document.
- Corrected various minor grammatical and technical terminology used throughout the document.
- Added clarification that Mitigation 1 does not prevent, but reduces risk of privileged accounts being exposed to attackers (page 14).
- Added that credential theft is not a problem that can be addressed with a simple software update under (page 15)
- Added clarification to Mitigation 3 when adding firewall exception rules for applications (page 66).