Mitigating Service Account Credential Theft on Windows

Reducing the risk of automated authentication against untrusted endpoints

Version 1.0.0
Last Updated: 2014-09-16

Mitigating Service Account Credential Theft on Windows

Disclaimer

This document is for informational purposes only. The authors make no warranties, express, implied, or statutory as to the information in the document. This document is provided "as-is". Information and views expressed in this document, including URLs and other Internet website references, may change without notice. You bear the risk of using it.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Authors

HD Moore	**Joe Bialek**	**Ashwath Murthy**
Rapid7	Microsoft	Palo Alto Networks

Contents

Executive Summary

Over the last 15 years, the Microsoft Windows ecosystem has expanded with the meteoric rise of the internet, business technology, and computing in general. The number of vendors that provide management, assessment, and monitoring tools has exploded, along with the need for these products to handle ever-growing networks and respond to evolving security threats. The networks themselves are now becoming less trusted, as targeted attacks, advanced malware, cloud services, and bring-your-own-device (BYOD) policies erode the historic trust model of internal versus external networks. The time has come to assume breach when considering all aspects of network security.

Mindsets about the network perimeter may be changing, but most management, assessment, and monitoring products still rely on trust boundaries and unidirectional authentication to the assets they access. For example, an automated backup service running on a central server under the context of a privileged account may automatically authenticate to workstations in order to access their file systems. A compromised or otherwise untrusted workstation can take advantage of this to steal the credentials of the backup service during the authentication process. Similar problems affect everything from network monitoring systems to vulnerability assessment products.

This has led to an "elephant in the room" mentality among security practitioners, where there is a tacit understanding that the automated tools they use to maintain the security of the network could end up enabling an attack instead. Security product vendors often call out these risks in their documentation, but the greater IT ecosystem is less likely to be aware of these problems.

This document describes practical mitigation strategies that reduce the effectiveness of attacks against automated authentication processes in a Windows environment, with a focus on accounts used by privileged services. Specific attacks are documented, along with mitigation techniques that apply to all commonly-used versions of the Windows operating system.

HD Moore
Chief Research Officer
Rapid7, Inc.

Introduction

The Windows operating system runs background jobs in the context of Services, each of which is associated with either a local user account, or in an Active Directory environment, a domain account. Software vendors either generate their own user accounts during installation, or ask their customers to assign a username and password that has the appropriate rights for their service to run. This concept applies to any management, monitoring, or assessment product that depends on credentials to communicate with organizational assets.

The way that these accounts are managed, in combination with common configuration weaknesses, may lead to the inadvertent exposure of privileged credentials to untrusted endpoints. Attackers have a number of tools at their disposal that can allow trivial exploitation of these issues. This document attempts to describe the issue at depth and provide mitigation strategies that may prevent exploitation in the future.

Attacks

Windows supports two primary authentication protocols, NTLM and Kerberos. Windows networks depend on these protocols for authentication between clients and servers, servers and servers, and server-bound processes connecting to clients. A common problem is the management of credentials used by automated services. These credentials are rarely changed, often used to authenticate against untrusted endpoints, and generally overlooked by IT and security staff alike as attack vectors. The majority of management and security products rely on built-in authentication to query assets, deploy updates, and generally keep the network running.

NTLM and Kerberos both have well-known issues that allow an attacker who is able to view parts of the NTLM challenge-response or the Kerberos AS-REQ challenge-request to brute force the account's password. NTLM, in its default configuration, is also vulnerable to relay attacks that are described in further detail below. Note that the Kerberos issues are not unique to the Microsoft implementation of Kerberos, but are weaknesses in the protocol itself.

Kerberos

Attacking the Kerberos AS-REQ

Kerberos authentication depends on communication between the Kerberos client and a Kerberos Key Distribution Center (KDC) server. The initial request, known as an AS-REQ, is used to request a Ticket Granting Ticket (TGT) and session key. In the default configuration, pre-authentication is required for all accounts, and the initial AS-REQ will be rejected. The KDC will inform the Kerberos client that pre-authentication is required and supply the client with a list of allowable encryption types to use, including a salt to use where applicable.

The client will then make a second AS-REQ to the KDC. The second AS-REQ will contain a pre-authenticator, which is the output of an encryption and hashing algorithm applied to a timestamp. The encryption key is derived from the user's password when password authentication is enabled. When smart cards are used, public key cryptography is used. This document does not address attacks which use Kerberos PKINIT for the AS-REQ.

An attacker who is able to view communication between a Kerberos client and the KDC can intercept this AS-REQ and use the pre-authenticator as an oracle to crack the user's plaintext password. The use of a weak password or a weak encryption type, such as DES-CBC-CRC or DES-CBC-MD5, can lead to the recovery of the user's plaintext password. If the user chooses a sufficiently strong password and a sufficiently strong encryption type is used, it is effectively impossible for an offline brute force attack of the pre-authenticator to succeed.

Downgrade Attacks on Kerberos Encryption

Kerberos supports multiple encryption algorithms for the pre-authenticator. Windows XP and Server 2003 support the DES-CBC-CRC, DES-CBC-MD5, and RC4-HMAC encryption types. Starting with Windows Vista and Server 2008, the AES128-CTS-HMAC-SHA1-96, and AES256-CTS-HMAC-SHA1-96 encryption types become available. Vista and Server 2008 also disable the use of DES-based encryption types by default. Note that both Windows XP and Server 2003 are no longer supported and any legacy systems should be migrated to Windows 8.1 and Server 2012 R2 respectively.

When an AS-REQ is made, the KDC will return a list of encryption types that it supports. The Kerberos client will choose the strongest encryption type that both it and the KDC support, and will then use the selected encryption type for the pre-authentication. An attacker that is able to man-in-the-middle (MITM) the connection between the Kerberos client and KDC can modify the encryption types the KDC claims to support in order to force a weaker algorithm.

If the DES algorithm is supported by the client, the attacker can conduct an offline brute force of the entire key space to obtain the user's Kerberos secret key. If the client does not support DES-based encryption types, the attacker can still try to downgrade the client to RC4-HMAC, which will speed up the rate of password cracking compared with the AES-based encryption types. Note that as of now, RC4-HMAC pre-authenticators cannot be completely enumerated through offline brute force attacks, so a strong password is still an effective defense.

Exploiting Kerberos

A number of open source and free security tools are available to demonstrate these issues.

- KerbCrack by Arne Vidstrom
- Cain & Abel by Massimiliano Montoro
- Password Recovery by ElcomSoft

NTLM

The NT LAN Manager (NTLM) protocol uses challenge-response authentication and is currently implemented in two versions, NTLMv1 and NTLMv2. Both versions of NTLM suffer from well-documented weaknesses. An attacker that is able to observe or man-in-the-middle (MITM) a NTLM session has a number of options available to recover the authentication credentials. An attacker that is able to convince a privileged account to authenticate to their system or one they can MITM can also force a downgrade attack unless prevented through security policies. Finally, an attacker that receives an authentication request from a privileged account or that can MITM a target, can redirect this authentication to another system through a relay attack. For all of these reasons, NTLM is considered to be less robust than Kerberos for Windows environments.

Attacking the NTLMv1 Challenge-Response

In NTLMv1, the server sends the client an 8-byte nonce during the challenge-response portion of the authentication process. The client computes two hashes of the user's clear-text password, one based on the MD4 algorithm (The NT One Way Function, or NTOWF) and a second hash based on the DES algorithm (the LanMan One Way Function, or LMOWF). The client splits the 16-byte NTOWF into 7-byte chunks, pads the last chunk with null bytes so that it is 7 bytes long, and uses each chunk as a DES key to encrypt the nonce with. The client then does the same thing with the LMOWF hash.

The client sends the resulting encrypted nonces to the server. Since the server knows the symmetric key, represented as the user's password hash in both LMOWF and NTOWF format, it can compute the same hashes as the client, check that the nonce is correct, and thus infer that the client knows the user's password. Critically, the server is only verifying that the client knows the hashes of the plaintext password (LMOWF and NTOWF), not the plaintext password itself.

Since the DES key space is relatively small, it is practical for an attacker that can observe the NTLMv1 challenge-response exchange to conduct an offline brute-force attack of the DES encryption key and recover the user’s NTOWF. The attacker can use the recovered NTOWF to authenticate to other NTLM-enabled servers as that user account without needing to know the actual plaintext password.

In short, regardless of how complex a user's password may be, or whether LANMAN hashes (LMOWF) are disabled, credentials sent over NTLMv1 can be cracked through an offline brute-force attack of the three DES encryption keys that make up the challenge-response. Note that due to the 7-byte blocks of a 16-byte input, the third DES key is only 2 bytes long, and the offline brute-force would focus on the first two keys only, calculating the last key almost instantly.

Attacking the NTLMv2 Challenge-Response

NTLMv2 works similarly to NTLMv1 except that it is significantly more resistant to offline brute-force attacks. In the context of the attacks being described, the major difference between NTLMv2 and NTLMv1 is that NTLMv2 does not use DES encryption to encrypt the nonce; instead, it calculates an MD5-based HMAC on the nonce with the user's NTOWF as the key.

Currently it is not feasible to brute-force all possible input keys for MD5 HMAC, so a sufficiently strong password will be resilient to the types of brute-force attacks that can be performed against NTLMv1. An attacker can still use the NTLMv2 challenge as an oracle to perform an offline brute force attack of the user's password. A weak password will be easy to brute-force, while a strong password may be effectively impossible to recover.

Downgrade Attacks on NTLMv2

Windows security policies, specifically the LMCompatibilityLevel setting, determines the allowed NTLM versions and session types, both for inbound and outbound authentication. The LMCompatibilityLevel setting prior to Windows Vista and Server 2008 would allow NTLMv1 for both inbound and outbound authentication.

An attacker that can MITM a connection between a client and a server or force a privileged account to authenticate to a system they control can try to convince the client to authenticate using NTLMv1. The default settings of Windows Vista and Server 2008 reduce the impact of downgrade attacks, but the method is still effective for Windows XP and Server 2003 systems, and on networks where backwards compatibility with NTLMv1 has been explicitly enabled.

NTLM Relay Attacks

NTLMv1 and NTLMv2 are vulnerable to relay attacks. If a service attempts to connect to a system controlled by the attacker, the attacker can relay the connection to another server. This relay can occur across any combination of protocols that support NTLM authentication. At a high level, the way this works is as follows:

1. The attacker receives a connection from the initiating system and service
2. The attacker relays this connection to a server that it wants to access
3. The initiating system authenticates to the target server via the relay
4. The attacker disconnects the initiating system and stops the relay
5. The attacker has an authenticated session on the target server

If a privileged account is used to authenticate to an attacker-controlled system, the attacker can leverage the NTLM relay technique to compromise almost any system in the domain. Note that while mandatory SMB signing can prevent the attacker from injecting new commands into the session, it does not prevent the attacker from continuing to proxy the connection, resulting in visibility into any network resources or files accessed by the server.

Exploiting NTLM

A number of open source and free security tools are available to demonstrate these issues.

- Responder from Trustwave SpiderLabs
- Squirtle by Kurt Grutzmacher
- Cain & Abel by Massimiliano Montoro
- SMBRelay3 by Andres and Miguel Tarasco
- The Metasploit Framework by Rapid7

Offline brute-force capabilities are available via a mix of open source, free, and commercial tools and services.

- John the Ripper by Solar Designer
- Cain & Abel by Massimiliano Montoro
- HashCat & oclHashCat by Atom
- L0phtcrack by L0pht Holdings
- Password Recovery by ElcomSoft
- CloudCracker.com by Thoughtcrime Labs

Defenses

Over the last 8 years, Microsoft has made significant improvements to the authentication protocols of the Windows operating system, but compatibility concerns have resulted in relatively low levels of awareness and adoption. Starting with Windows Vista and Server 2008, version 2 of the Server Message Block Protocol (SMB) became available, and featured SHA256-based message signing, among many other improvements. With the introduction of Windows 8 and Server 2012, Microsoft introduced SMB3, which provides AES-based message signing and end-to-end encryption, a major improvement that takes into account the erosion of trust within internal networks.

These improvements aside, many organizations still use SMB1 and continue to use NTLM authentication across the enterprise. Organizations that use older operating systems can still apply mitigations that reduce the viability of an attack or at least improve the detection of an intrusion. Microsoft and all organizations dependent on Windows are challenged by legacy systems that are not compatible with stronger security controls. This forces many organizations to focus their efforts on detection instead of prevention.

Expectations

There is no perfect solution to the problem of service account credential management and untrusted endpoints. Most proposed solutions will be met with criticism, and rightly so, as even when the capabilities are available to stop attacks, these changes may not be compatible with legacy systems or business processes. The guidelines below are designed to help organizations improve the overall security of their Windows environment, but may not be applicable to every network, and are certainly not a comprehensive defense against a determined attacker with administrative access to managed endpoints.

Recommendations

Kerberos should be mandated and NTLM authentication disabled if at all possible. This is relatively easy to accomplish when building a network from scratch, but can be challenging for existing environments that need to support a wide range of operating systems and authentication methods. In an effort to provide useful mitigation steps to all organizations, the following recommendations have been grouped by protocol and specific use cases.

Hardening Kerberos

1. Disable weaker HMAC algorithms. Ideally only the AES-256 encryption type should be enabled. Note that Server 2008 and Vista are the first Microsoft operating systems to support AES; the strongest encryption algorithm supported by Windows XP and Server 2003 is RC4-based HMAC. Interoperating with older Kerberos implementations may require the use of RC4-HMAC or even DES-based encryption types, but most organizations can at least depend on RC4-HMAC. Modern versions of Linux support AES-256 for Kerberos authentication to a Server 2012 KDC.

2. Enable Kerberos Armoring to protect against attackers using captured AS-REQ's to brute force passwords. Kerberos Armoring encrypts the users AS-REQ with the computer accounts key, which is guaranteed to be complex. This feature is only supported on Windows 8, Server 2012, and newer operating systems.

Hardening NTLM

1. Disable NTLM authentication within the domain and only use Kerberos if at all possible. Note that attempts to authenticate to a server by IP address will still attempt to use NTLMv2 even when Kerberos is mandated.

2. Use Group Policies to disable NTLM authentication on clients and servers within the domain that are compatible with Kerberos. At the least, this will limit the attack surface for NTLM-related exposures.

3. Disable NTLMv1 and mandate NTLMv2 on all systems within the domain. This is especially important for servers and any services that make frequent connections to untrusted workstations. NTLMv1 will always expose the account credential, regardless of complexity, due to the weakness of the DES algorithm.

4. Mandate the use of SMB signing on all systems within the domain to protect against NTLM relay attacks. SMB signing is not a perfect fix, as an attacker can simply proxy the connection to another target and view data exposed over the connection, even if they can't issue their own requests. SMB signing does not protect against NTLM relay attacks against non-SMB resources, such as DCERPC endpoints, Microsoft SQL Server, and Microsoft IIS web servers and services.

5. Enable extended protection to prevent NTLM relay against IIS web servers. Note that extended protection only works if the web server has been configured to use TLS. Without TLS an attacker can modify the Service Principal Name (SPN), defeating this protection.

6. Disable SMB1. This is not an option for organizations that need their servers to be accessible to Windows XP clients or to support older non-Microsoft SMB implementations. Disabling SMB1 can dramatically improve the security of the network by enabling stronger signing algorithms and end-to-end encryption when SMB3 is available. SMB2 is supported by Mac OS X as of 10.9 (Mavericks) and the open source Samba software starting with version 3.6. Linux and BSD-based network attached storage (NAS) systems typically use Samba, but may not be running a version recent enough for full compatibility.

Hardening Service Accounts

1. Service accounts should be configured with long passwords, consisting of at least 32 characters, randomly generated using a vetted password generation tool, and containing as wide of a range of characters as possible.

2. Only allow logon types needed, such as Network Logon and Logon as a Service. Explicitly prevent these accounts from logging on interactively, unless required by the service. Note that this does not prevent a service account from accessing the file system of a target (ADMIN$) or even executing certain RPC calls, but it does prevent interactive logins via normal mechanisms.

3. Service accounts should be explicitly denied access to external resources such as VPN gateways, Outlook Web Access servers, and IIS web servers that support NTLM authentication.

4. Service accounts should have the least privileges needed for the application to function. Unfortunately this often equates to administrative access, but it is possible to reduce the privileges of some services and still retain functionality.

5. Service accounts are often only used to authenticate from certain servers to the rest of the network, or from client workstations to certain servers. It may be possible to restrict the account so that it can only be used to access those specific systems through the use of Group Policies.

6. Service account passwords should be changed on a frequent basis. The frequency will depend on the application, but monthly should be considered the minimum, with weekly being an option for some services. This can be automated using Powershell scripts or through third-party products.

7. If at least one Server 2012 domain controller is available and have the services in question are running on Server 2012 systems, Group Managed Service Accounts can handle password rolling automatically. The accounts are managed by Active Directory and will be auto-rolled by the domain controller.

8. Authentication Policies and Silos can also be used to restrict service accounts and user accounts to specific sets of servers, but these features are not backwards compatible with older operating system versions.

9. It may be useful to move all accounts out of the default Domain Users group and to create specific groups for each department or role. This allows precise configuration of service accounts privileges and simplifies auditing.

Hardening LDAP

In addition to NTLM and Kerberos, many organizations use the Lightweight Directory Access Protocol (LDAP) services provided by Active Directory to centralize authentication across non-Windows services, including Linux server authentication and web applications.

1. Consider the use of SASL or Sicily for services that need to bind to the directory server, instead of relying on STARTTLS to secure user credentials. When using SASL, select the strongest mechanism supported by the operating systems in use.

2. Mandate signing of all LDAP requests to prevent replay attacks. A simple registry change can be used to identify clients that do not support signing.

3. Monitor and inspect all bind attempts from service accounts and force the use of STARTTLS on all incoming connections to the directory server.

4. Monitor and discourage the use of simple authentication.

Hardening the Perimeter

This document is focused on the mitigation of credential theft due to automated authentication attempts against internal assets, but these issues can also apply to external assets and affect internal client systems, unless the perimeter firewalls and external systems are properly configured.

1. Block outbound access to TCP ports 88, 135, 139, and 445 and UDP ports 137, 138, and 88 at the network perimeter. Block outbound access on these ports for all external servers using the built-in Windows Firewall. These changes prevent easy credential harvesting through client and server side attacks that make use of UNC paths.

2. Block inbound access to all TCP and UDP services on external servers except for those explicitly used to provide necessary functionality and enable remote administration. Restrict access to administrative services, such as Remote Desktop, to trusted IP ranges.

Improving Monitoring

1. Events should be collected from servers, workstations, and perimeter devices and stored in a system that allows for full-text searches. This system should support notifications when specific signatures or patterns are identified. Security Information and Event Management (SIEM) products are commonly used for this purpose, but open source and free tools are also available that can be effective with a modest time investment.

2. Starting with Vista and Server 2008 R2, Windows supports event log forwarding and collection (WEF). Source Initiated Subscriptions are preferable to pull mode, especially in the context of protecting service accounts. The Centralizing Windows Events with Event Forwarding guide by Avecto provides an excellent walk-though of this process. Older operating systems can be handled with the open source Eventlog to Syslog Service and a compatible syslog service, but keep in mind that this service does not encrypt the event data as it travels over the network. Commercial SIEM products typically use agents or collector services for automated event log aggregation.

3. The Sysmon service, recently released by Microsoft, is an excellent source of event data that goes far beyond what standard auditing policies provide. Sysmon generates event log entries that existing SIEM solutions can make use of with little additional effort.

4. Failed logon events can be an early indicator of a compromised service account, especially when interactive logins have been disabled for that account or group. For example, a service account that attempts to login interactively via Remote Desktop is a clear indicator of a stolen credential. Any failed logon by a service account can indicate that the target of the authentication was an unmanaged asset, has been misconfigured, or is controlled by an attacker that was waiting for the service account to authenticate in order to crack or relay its credentials.

5. Account creation and group membership change events initiated by a service account should be treated as red flags and investigated as soon as possible. Attackers often use stolen credentials to create additional accounts for long-term use.

6. Events generated by MSIInstaller, Service Control Manager, and Volume Shadow Copy services associated with a service account can provide an early indicator of compromise. The MSIInstaller service generates events when new software is installed, the Service Control Manager is a frequent target of attackers using the PSExec method of remote code execution, and the Volume Shadow Copy service can be used to access the Active Directory database (ntds.dit) and local user account (SAM) database on a running server.