Compromising Plaintext Passwords in Active Directory

blog.netwrix.com/2022/10/06/compromising-plain-text-passwords-active-directory

Jeff Warren

A lot of attention gets paid to preventing pass-the-hash and pass-the-ticket attacks, but these tactics limit adversaries to what they can perform from the command line. Compromising a plaintext password gives an attacker unlimited access to an account — which can include access to web applications, VPN, email and more.

One way to extract plaintext passwords is through Kerberoasting, but this brute-force technique takes a lot of time and patience. There are quicker and easier ways to extract plaintext passwords, which we'll explore in this post.

Handpicked related content:

[Free Guide] Active Directory Security Best Practices

Group Policy Preferences

In Windows Server 2008, Microsoft introduced Group Policy Preferences (GPPs). One of the common use cases for GPPs is to create and manage local accounts (such as the Administrator account) on servers and workstations. As part of this, an administrator can push out a password for these accounts.

The password is stored in the Group Policy XML file in SYSVOL and is encrypted using an AES key. However, Microsoft published the AES key, which can be used to decrypt these passwords — making them effectively plaintext passwords.

Because the SYSVOL share is open to Authenticated Users, any user in the organization can read the files stored there. Therefore, any user account can find and decrypt the Group Policy file and thereby gain access to the plaintext passwords for Administrator accounts. The PowerSploit command Get-GPPPassword will find and decrypt these passwords for you.

For a more detailed write-up on this, check out Sean Metcalf's post and Microsoft's post. Also, Microsoft provides a useful script for scanning for GPPs that contain passwords as part of their security bulletin.

Mimikatz and LSASS Minidumps

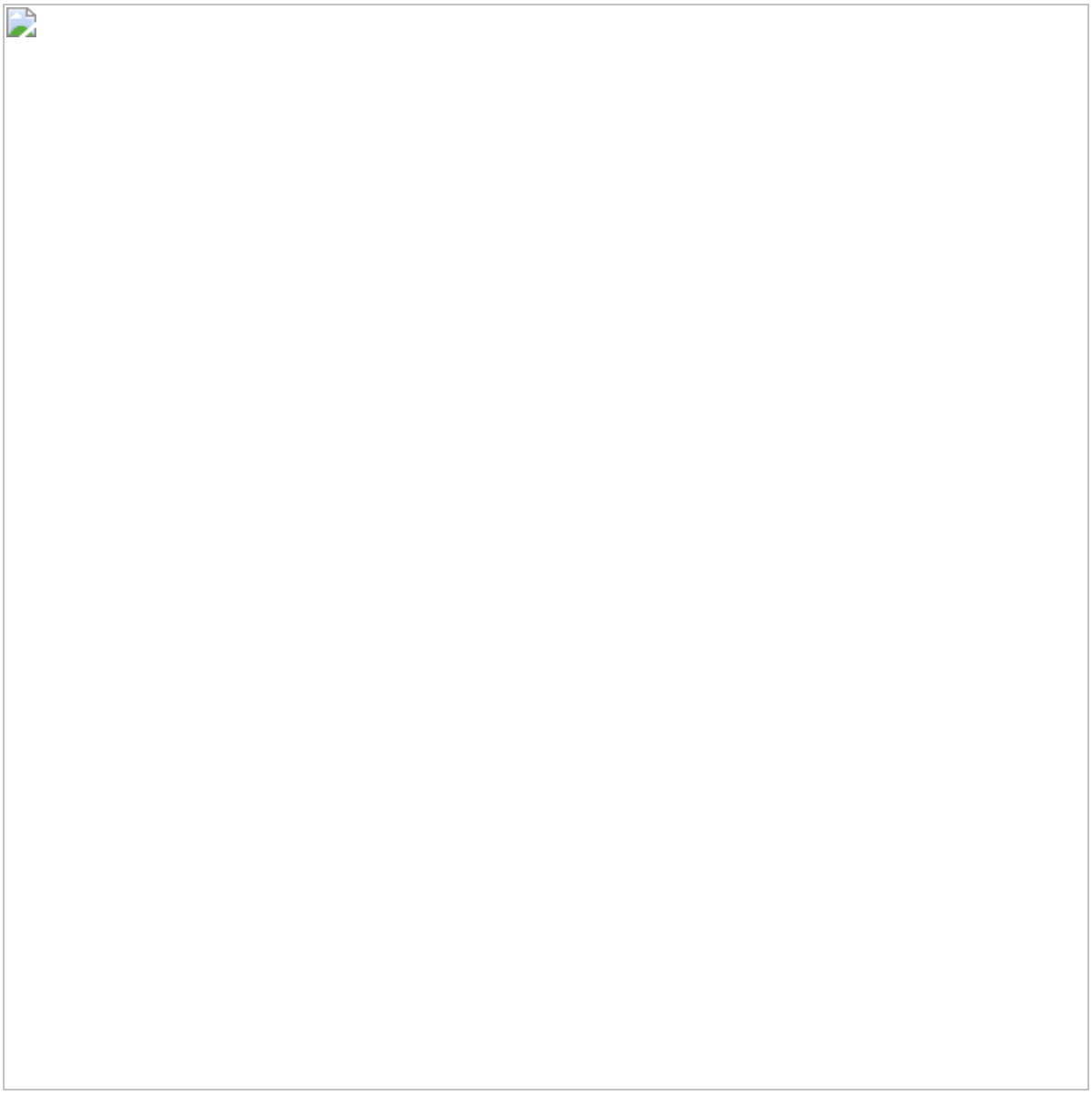
Typically, Mimikatz is used to extract NTLM password hashes or Kerberos tickets from memory. However, one of its lesser-known capabilities is the ability to extract plaintext passwords from dumps created for the LSASS process. This means that an attacker can compromise plaintext passwords without running any nefarious code on domain controllers. Dump files can be created interactively or using ProcDump, and in either

case, the activity is unlikely to be flagged by anti-virus software. Once the dumps are created, they can be copied off the domain controller and the plaintext credentials can be harvested using Mimikatz offline.

Here you can see the creation of the process dump on a domain controller using ProcDump:

This command essentially creates a snapshot of the LSASS process, which contains plaintext password information:

Once created, the file can be copied to another host for offline password extraction using Mimikatz. By using the sekurlsa::minidump command, you can switch the context of Mimikatz to the extracted dump file and issue the sekurlsa::logonpasswords command:



Using Mimikatz Against the Digest Authentication Protocol

The Digest Authentication protocol (WDigest.dll), introduced in Windows XP, is used for HTTP and SASL. Most importantly, enabling WDigest will result in the storing of plaintext credentials for locally authenticated accounts. In 2014, Microsoft released a patch that allows you to disable WDigest using the UseLogonCredential registry value. However, many organizations still run many servers and workstations with WDigest enabled.

If WDigest is enabled, an adversary can extract plaintext credentials easily with the sekurlsa::logonpasswords command:

Exploiting Reversible Encryption

Active Directory enables the storing of user passwords with reversible encryption, which is essentially the same as storing them in plain text. This policy was introduced in Windows Server 2000 and still exists in even the most recent versions. According to Microsoft, it was introduced to provide “support for applications that use protocols that require the user’s password for authentication.”

By default, this policy is off; but if it is enabled, an adversary can easily extract cleartext passwords using techniques such as DCSync:

The command above will return the plaintext password:

The reversible encryption policy can be enabled through the Group Policy User Account Control settings and through fine-grained password policies. An attacker may be able to maliciously create a fine-grained password policy that links to Domain Admins to enable their passwords to be stored with reversible encryption, giving them access to the plaintext password for privileged accounts.

Conclusion

As you can see, there is no shortage of ways for an interested attacker to obtain plaintext passwords for Active Directory accounts. For more information, read our related posts on extracting plaintext passwords using PowerSploit, finding weak passwords, attacking weak passwords and attacking local account passwords.

Netwrix StealthDEFEND allows you to effectively detect this and even more sophisticated attacks across your infrastructure and respond to them in real time.

Jeff Warren