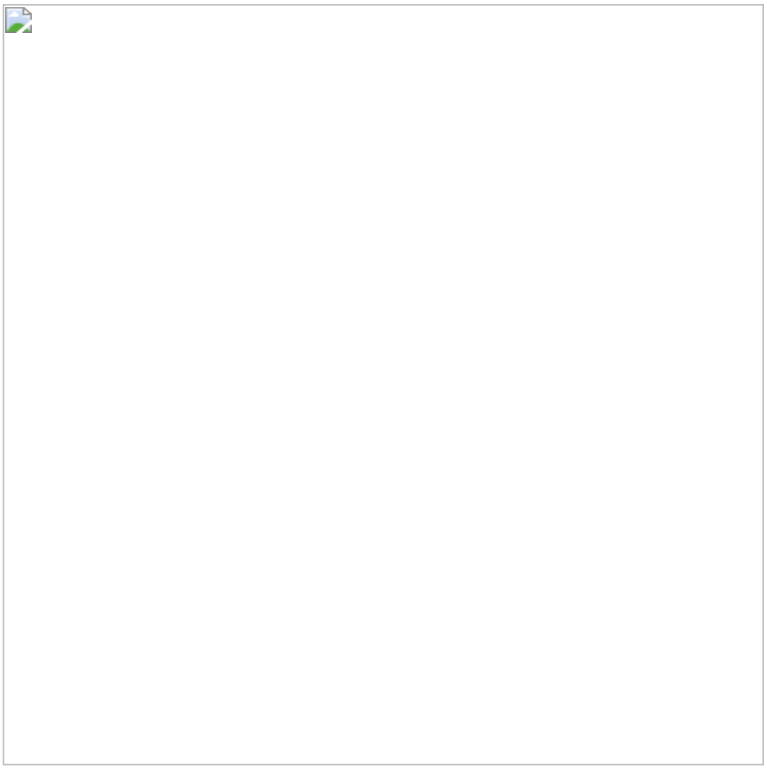
Extracting Password Hashes from the Ntds.dit File

blog.netwrix.com/2021/11/30/extracting-password-hashes-from-the-ntds-dit-file

Jeff Warren



With so much attention paid to credential-based attacks such as Pass-the-Hash (PtH) and Pass-the-Ticket (PtT), more serious and effective attacks are often overlooked. One such attack involves exfiltrating the Ntds.dit file from Active Directory domain controllers. Let's take a look at what this threat entails, how an attack can be performed and how you can protect your organization.

What is the Ntds.dit File?

The Ntds.dit file is a database that stores Active Directory data, including information about user objects, groups and group membership.

Importantly, the file also stores the password hashes for all users in the domain. Cybercriminals who extract these hashes can then perform PtH attacks using tools such as Mimikatz, or crack the passwords offline using tools like Hashcat. In fact, once an attacker has extracted the hashes, they are able to act as any user on the domain — including Domain Administrators.

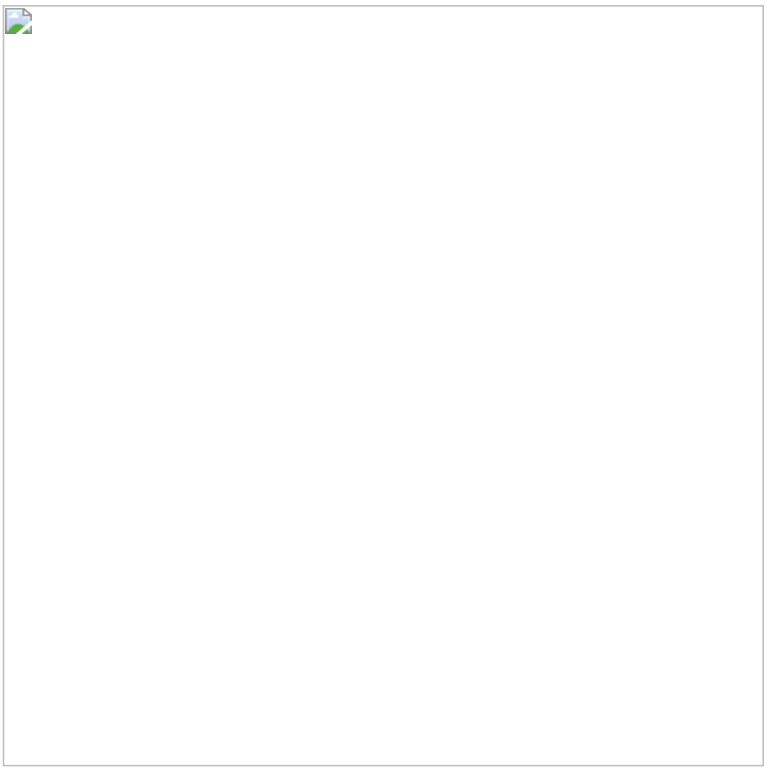
Handpicked related content:

[Free Guide] Active Directory Security Best Practices

How does an attack on the Ntds.dit file work?

Step 1. Steal the Ntds.dit file.

The first step is to get a copy of the Ntds.dit. This isn't as straightforward as it sounds because this file is constantly in use by AD and therefore locked. If you try to simply copy the file, you will see an error message like this:



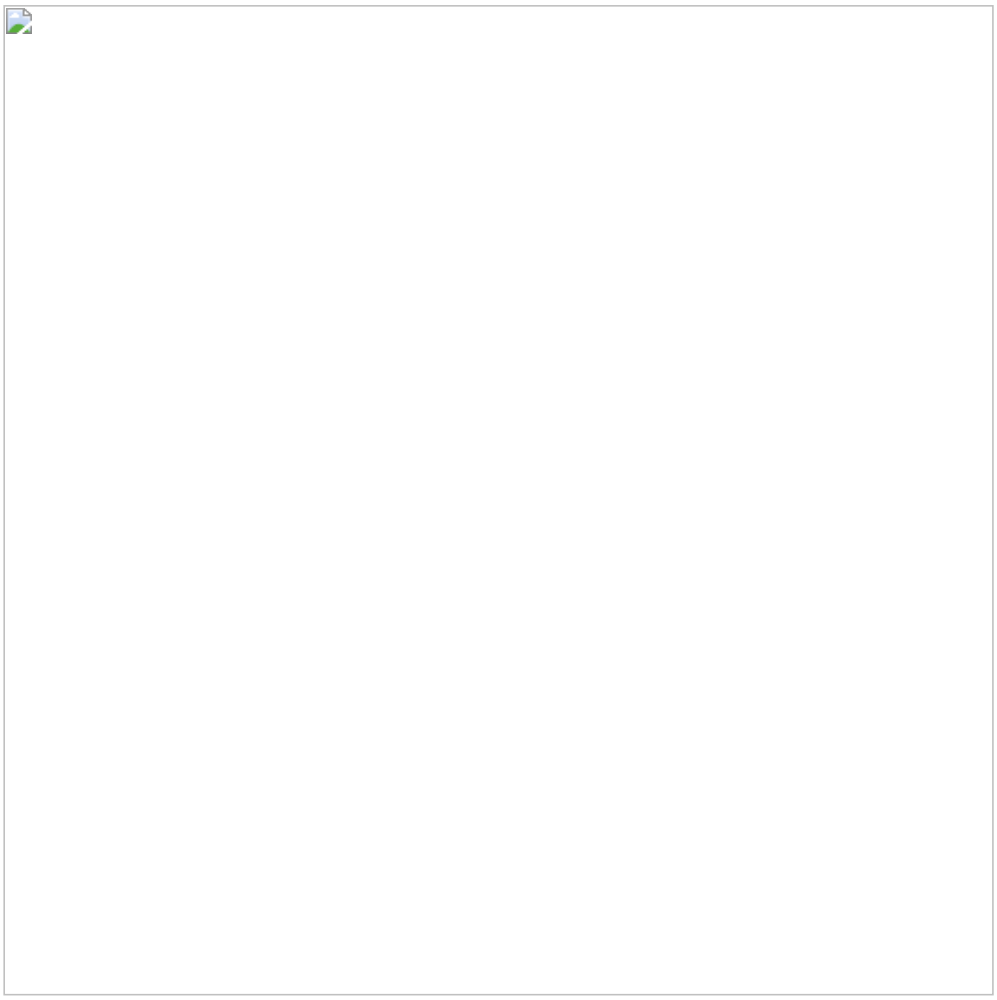
There are several ways around this roadblock using capabilities built into Windows or with PowerShell libraries. For example, an attacker can:

1. Use Volume Shadow Copies via the VSSAdmin command
2. Use the PowerSploit penetration testing PowerShell modules
3. Leverage the NTDSUtil diagnostic tool available as part of Active Directory
4. Leverage snapshots if the domain controllers are running as virtual machines

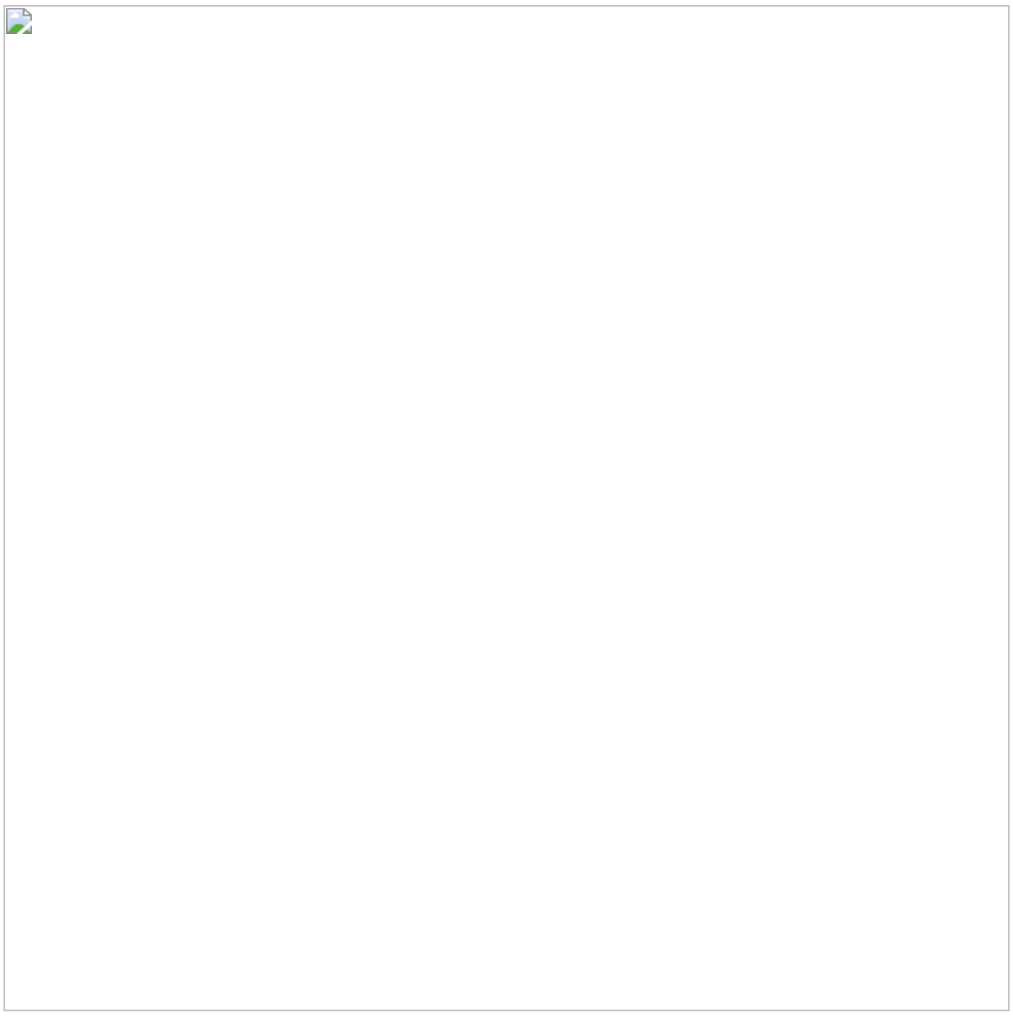
Let's walk through the first two of these approaches.

Using VSSAdmin to steal the Ntds.dit file

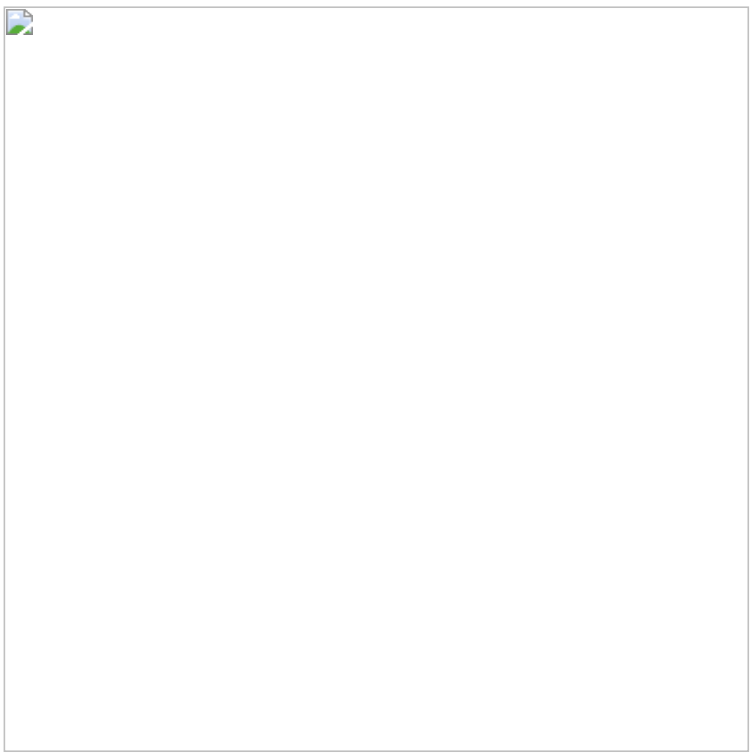
Step 1. Create a volume shadow copy:

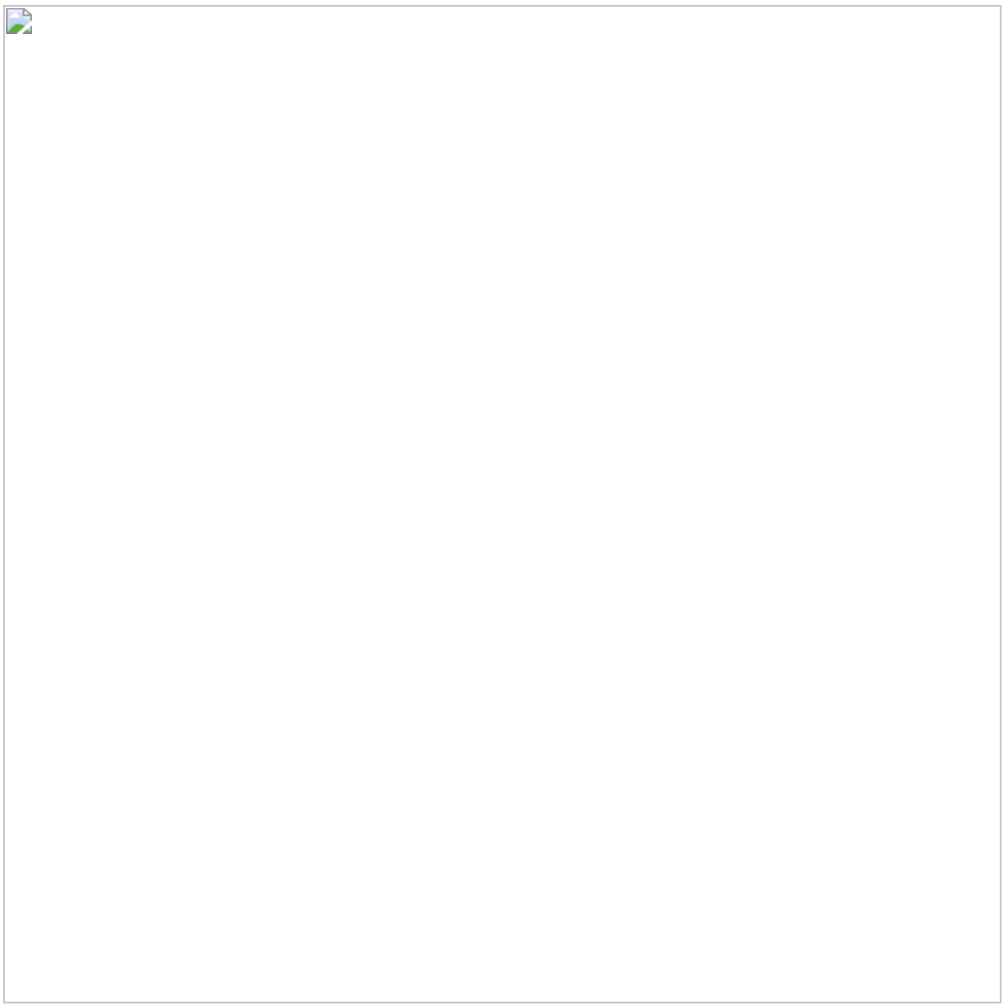


Step 2. Retrieve the Ntds.dit file from volume shadow copy:

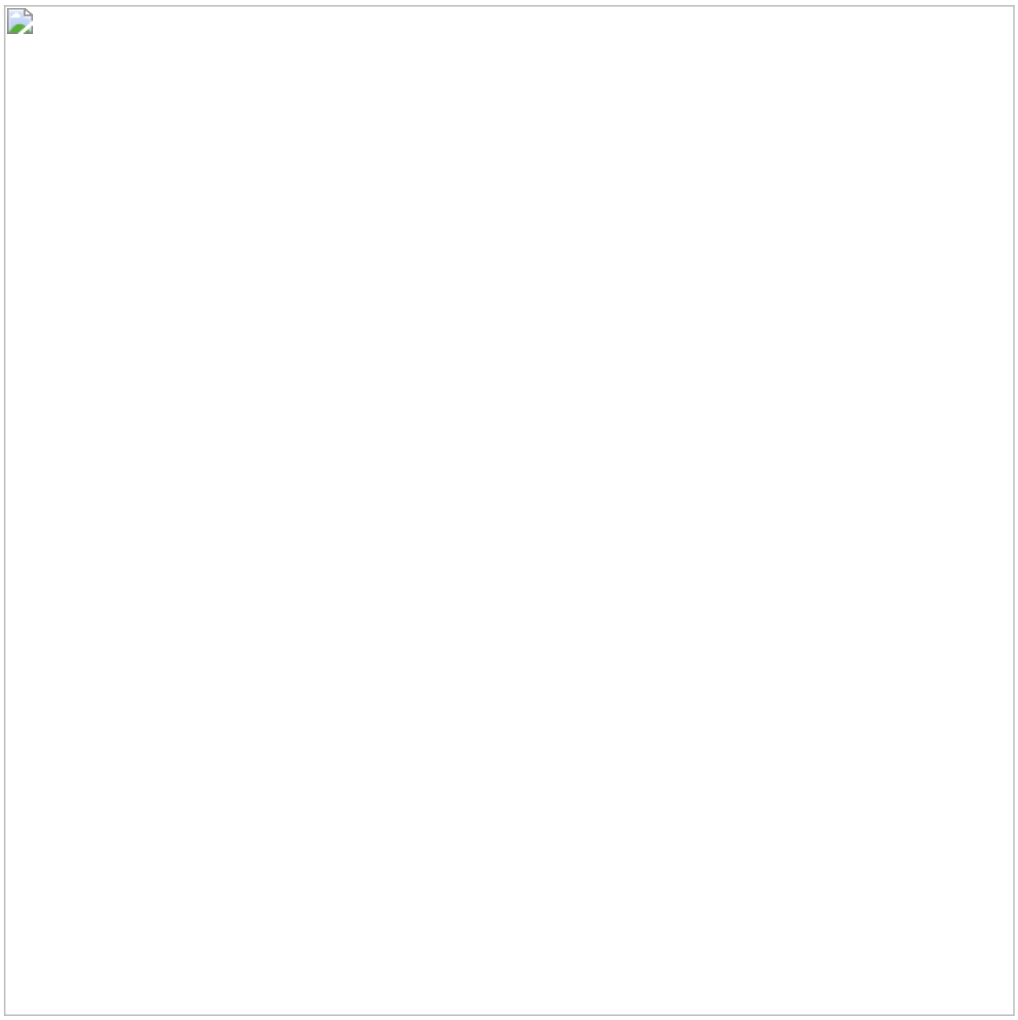


Step 3. Copy the SYSTEM file from the registry or volume shadow copy, since it contains the Boot Key that will be needed to decrypt the Ntds.dit file later:





Step 4. Cover your tracks:

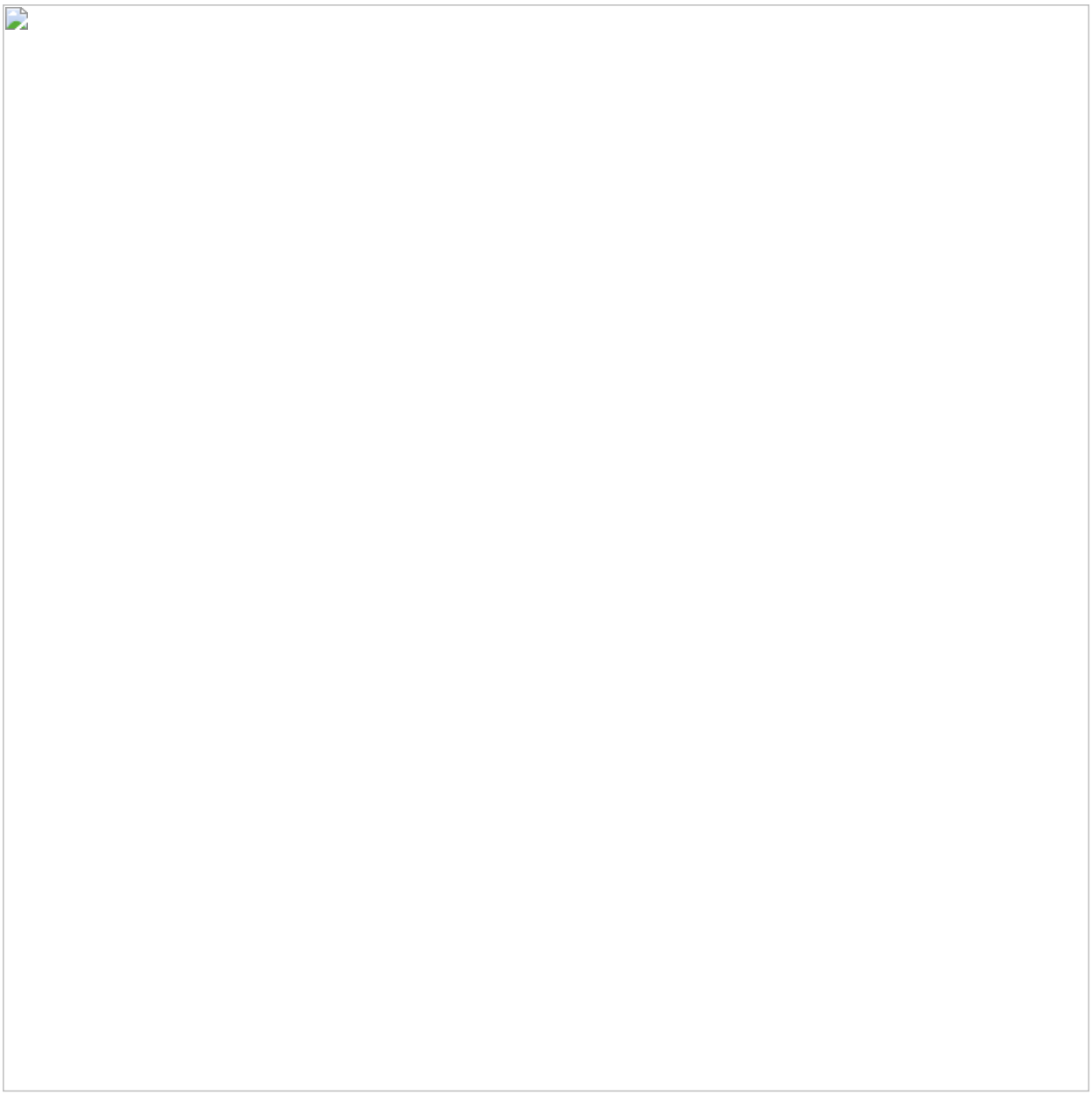


Using PowerSploit NinjaCopy to steal the Ntds.dit file

PowerSploit is a PowerShell penetration testing framework that contains various capabilities that can be used for the exploitation of Active Directory. The Invoke-NinjaCopy module copies a file from an NTFS-partitioned volume by reading the raw volume, which enables an attacker to access files that are locked by Active Directory without alerting any monitoring systems.

Step 2. Extract the password hashes

Once the attacker has a copy of the Ntds.dit file, the next step is to extract the password hashes from it. DSInternals provides a PowerShell module that can be used to interact with the Ntds.dit file; here's how to use it to extract password hashes:



Step 3. Use the password hashes to complete the attack.

Once an attacker has extracted the password hashes from the Ntds.dit file, they can use tools like Mimikatz to perform pass-the-hash (PtH) attacks. Furthermore, they can use tools like Hashcat to crack the passwords and obtain their clear text values. Once an attacker has those credentials, there are no limitations on what they can do with them.

How can organizations protect against attacks on the Ntds.dit file?

The best way defend your organization against this attack is to limit the number of users who can log on to domain controllers, which includes not just members of highly privileged groups such as Domain Admins and Enterprise Admins, but also members of less privileged groups like Print Operators, Server Operators and Account Operators. The membership of all these groups should be strictly limited, constantly monitored for changes and frequently recertified.

In addition, consider using monitoring software that can alert on — and even prevent — users from retrieving files off volume shadow copies.

How Netwrix solutions can help

Netwrix provides a multi-layered approach to defending against Ntds.dit password extraction attacks.

Detect attempts to steal the Ntds.dit file

Netwrix StealthDEFEND is an effective tool for detecting Ntds.dit password extraction attacks. It continually looks for unexpected access events on the Ntds.dit file, including:

- **Direct access to the file on the file system** — Since the Ntds.dit file is locked by Active Directory while in use, an attacker typically cannot obtain the file without stopping the Active Directory service. Monitoring for both successful and denied access events by user accounts can provide meaningful insight into unwanted access attempts, because the AD service runs as Local System.
- **Access to the Ntds.dit file through volume shadow copies** — Even if the Ntds.dit file is locked, attackers are able to create a shadow copy of the entire drive and extract the Ntds.dit file from the shadow copy.

Block access to the Ntds.dit file

Netwrix StealthINTERCEPT can block both direct access to the Ntds.dit file on the file system and access to it through volume shadow copies. As a result, even if an attacker stops Active Directory to unlock the file and has full admin rights, they will not be able to gain access to the file directly.

Proactively reduce your attack surface area

Netwrix StealthAUDIT helps you review and control the administrative groups with access to your domain controllers, such as Domain Admins and Server Operators. By closely limiting the membership of these groups, you reduce the risk of attackers gaining access to the Ntds.dit file.

Jeff Warren

Jeff Warren is SVP of Products at Netwrix. Before joining Netwrix, Jeff has held multiple roles within Stealthbits - now part of Netwrix, Technical Product Management group since joining the organization in 2010, initially building Stealthbits' SharePoint management offerings before shifting focus to the organization's Data Access Governance solution portfolio as a whole. Before joining Stealthbits - now part of Netwrix, Jeff was a Software Engineer at Wall Street Network, a solutions provider specializing in GIS software and custom SharePoint development. With deep knowledge and experience in technology, product and project management, Jeff and his teams are responsible for designing and delivering Stealthbits' high quality, innovative solutions. Jeff holds a Bachelor of Science degree in Information Systems from the University of Delaware.