Exploiting Weak Active Directory Permissions with PowerSploit

blog.netwrix.com/2023/04/14/exploiting-weak-active-directory-permissions

Jeff Warren

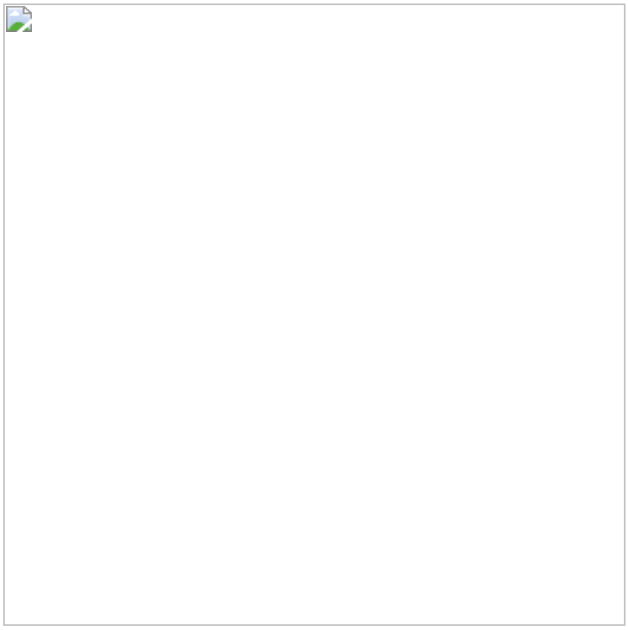
Adversaries use multiple techniques to identify and exploit weaknesses in Active Directory (AD) to gain access to critical systems and data. This blog post explores 3 ways they use PowerShell PowerSploit to elevate or abuse permissions, and offers effective strategies for protecting against them.

Handpicked related content:

[Free Guide] Active Directory Security Best Practices

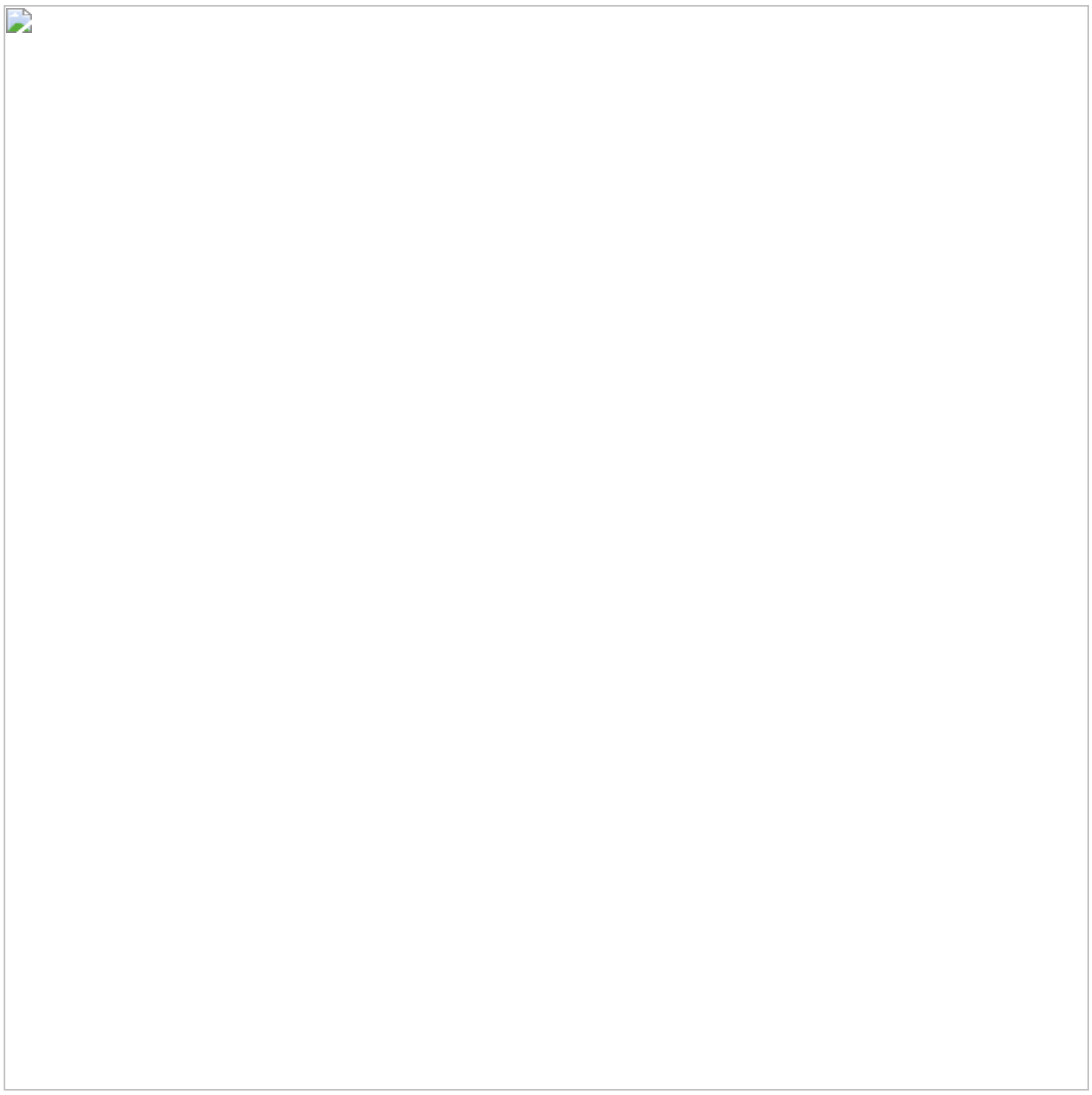
Finding Vulnerable AD Security Groups

By adding accounts they control to Active Directory security groups, adversaries can gain the access they need to carry out an attack. Often, they remove their membership afterward to hide their tracks.



Attackers first need to find AD groups that allow the group manager to update the group membership list. This ability is controlled by the “Manager can update membership list” setting shown below; the specified value is stored in the **CanManageWrite** attribute.

Attackers can use the PowerSploit command **Find?ManagedSecurityGroups** to get a list of groups and their respective managers, so they know which groups can be exploited and what accounts to target to gain modification rights:



Finding Weak Permissions

Another valuable tool in PowerSploit is the **Invoke-ACLScanner** command. As the name suggests, this command scans all access control lists (ACLs) and returns the associated permissions.

However, AD permissions can be complex and confusing, with many built-in permissions that are not easily exploitable and not worth investigating. Accordingly, Invoke-ACLScanner finds on the easiest permissions to exploit by filtering on two criteria:

- The security identifiers (SIDs) of the users or groups associated with the permission have resource IDs (RIDs) above 1000.
- The rights granted provide “modify” access to the target object.

As you can see below, with a single command, a hacker can see all exploitable permissions, whether they secure users, groups, Group Policy objects (GPOs), organizational units (OUs) or other AD objects:

Finding Rights for the Current User

If scanning all exploitable rights is too much work, it's simple to find the exploitable rights for an account the attacker already controls. The following command filters the list generated by Invoke?ACLScanner to show the rights for the logged-in user only:

```
Invoke-ACLScanner | Where-Object {$_.IdentityReference –eq
[System.Security.Principal.WindowsIdentity]::GetCurrent().Name}
```

This command returns a list of permissions that can be immediately used by the logged-in account:

Protecting Your Active Directory Permissions

The Netwrix Active Directory Security Solution can help you defend against attacks on AD permissions by making it easy to:

- Scan Active Directory permissions and report on weaknesses.
- Remove permissions granted to inactive or disabled accounts.
- Check the accuracy of the Managed-by group attribute.
- Enforce the least-privilege principle for all users, especially administrators.
- Enforce strong password policies.
- Promptly spot and respond to threats, including escalation of permissions.

Jeff Warren