Constrained delegation and resource-based delegation

crowe.com/cybersecurity-watch/constrained-delegation-resource-based-delegation-outsmart-attacks

Constrained delegation and resource-based delegation improve network security, but attackers can still penetrate environments.

Constrained delegation and resource-based constrained delegation are features of Microsoft Active Directory™ that were created to address many of the security issues in their predecessor, unconstrained delegation. Constrained delegation was introduced as part of Microsoft Windows Server™ 2003, and resource-based constrained delegation was introduced in Windows Server 2012 to enable restricted delegation across domains. Organizations should take steps to understand why constrained delegation and resource-based constrained delegation are safer ways to allow services to act on behalf of other users, as well as how these more secure means of delegation can be hardened to further defend against attacks.

Constrained delegation and resource-based constrained delegation: What’s the difference?

Constrained delegation and resource-based constrained delegation are both forms of delegation, like unconstrained delegation. In contrast to unconstrained delegation, in which a service configured for delegation can access any target service as any user that authenticates to it, constrained delegation and resource-based delegation limit the access of services configured for delegation by restricting the target services that they can access on behalf of another user.

Constrained delegation and resource-based constrained delegation differ in where the restrictions on delegation are enforced. In constrained delegation, the list of target services that a service configured for delegation can access as another user is stored in Active Directory with the service configured for delegation in its ms-DS-Allowed-To-Delegate-To attribute. In resource-based constrained delegation, the list of services that can access a target service as another user is stored in Active Directory with the target service in its ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity attribute.

One use case for delegation is when a web server must access a database on behalf of other users to retrieve their data and serve it on a web page. In constrained and resource-based constrained delegation, system administrators can enforce restrictions that prevent the web server from accessing any target service other than the database as another user.

In this context, constrained delegation would be configured by adding only the database server to the list of target services that the web server can access as another user. Resource-based constrained delegation would be configured by adding the web server to the list of services that can access the database service as another user. The effects of both configurations are similar, but resource-based constrained delegation has the advantage of working even when the web server and database server belong to different domains.

Mechanics

Two extensions to the Kerberos protocol support the modern implementation of constrained delegation and resource-based constrained delegation: Service for User to Self (S4U2self) and Service for User to Proxy (S4U2proxy).

S4U2self allows a service to request a service ticket to itself as another user. If the service account has its TrustedToAuthForDelegation flag set, then the service ticket returned is marked as forwardable; otherwise, it is not. Both forwardable and nonforwardable service tickets can be used with the S4U2proxy extension to access a different resource on behalf of another user. The intended use case for S4U2self is to allow services not compatible with Kerberos authentication to access resources as other users.

In Exhibit 1, a service (service A) uses S4U2self to request a ticket-granting service (TGS) ticket that allows access to itself as a privileged user (user P), and the ticket returned is only forwardable when the user running service A (user A) is set as TrustedToAuthForDelegation.

Exhibit 1: Self-access as another user with S4U2self

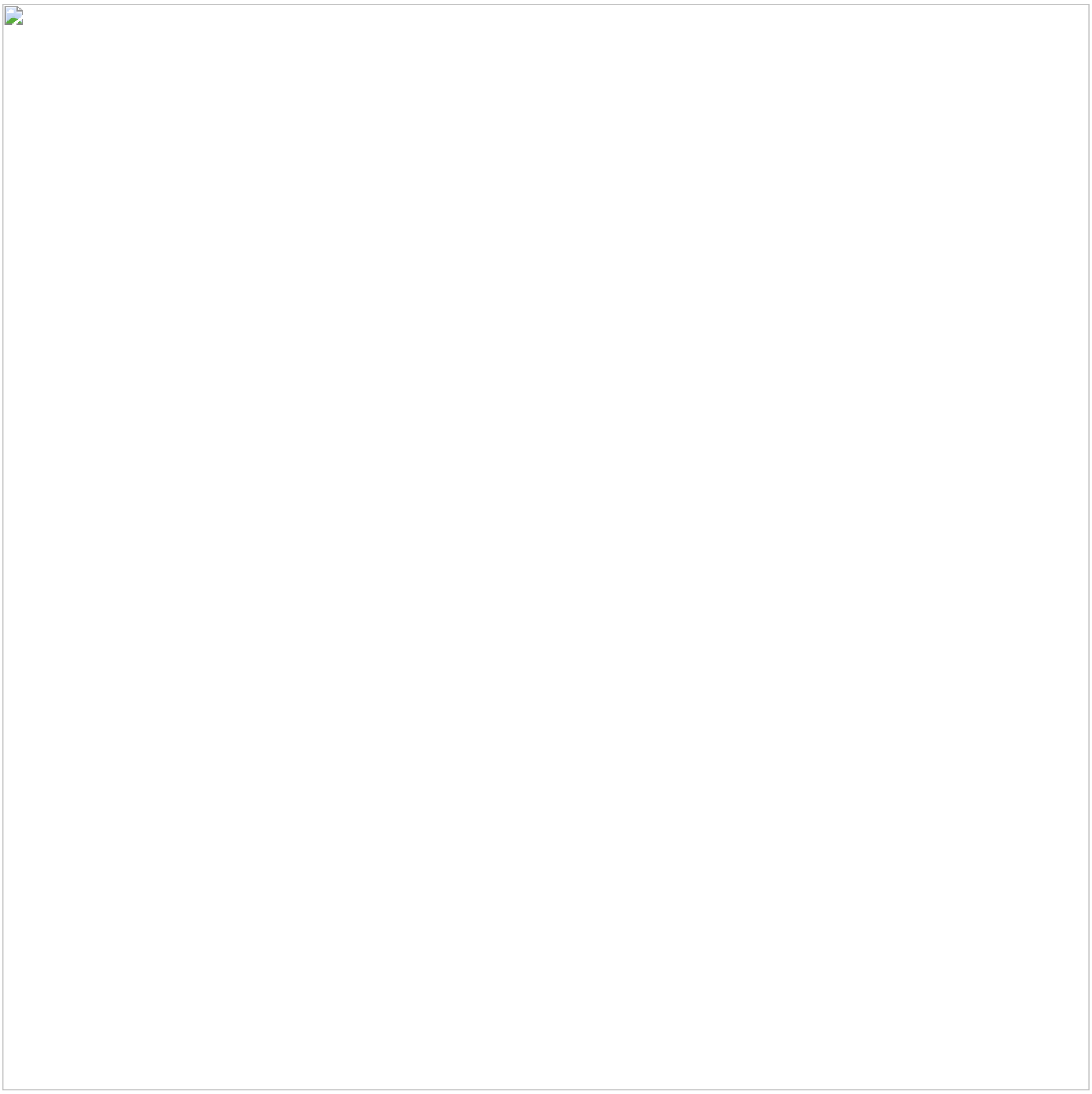
Source: Crowe analysis, March 2023

S4U2proxy enables a service (service A) to access a target service (service B) on behalf of another user through constrained delegation and resource-based constrained delegation, and its workings vary based on the type of delegation used. However, in all cases, it returns a forwardable service ticket to service A, which allows it to access service B on behalf of the specified user.

For both types of delegation, service A must have a service ticket that grants the user access to service A either sent to service A by the user or requested by service A using S4U2proxy. For constrained delegation, this service ticket must be marked as forwardable, and service A must be configured to access service B as another user. That is, user B must be included in user A's ms-DS-Allowed-To-Delegate-To attribute.

In Exhibit 2, service A uses S4U2proxy to access service B as user P. Service A uses a forwardable TGS ticket, granting user P access to service A as proof of authentication to request a TGS ticket for service B as user P.

Exhibit 2: Service access with constrained delegation and S4U2proxy

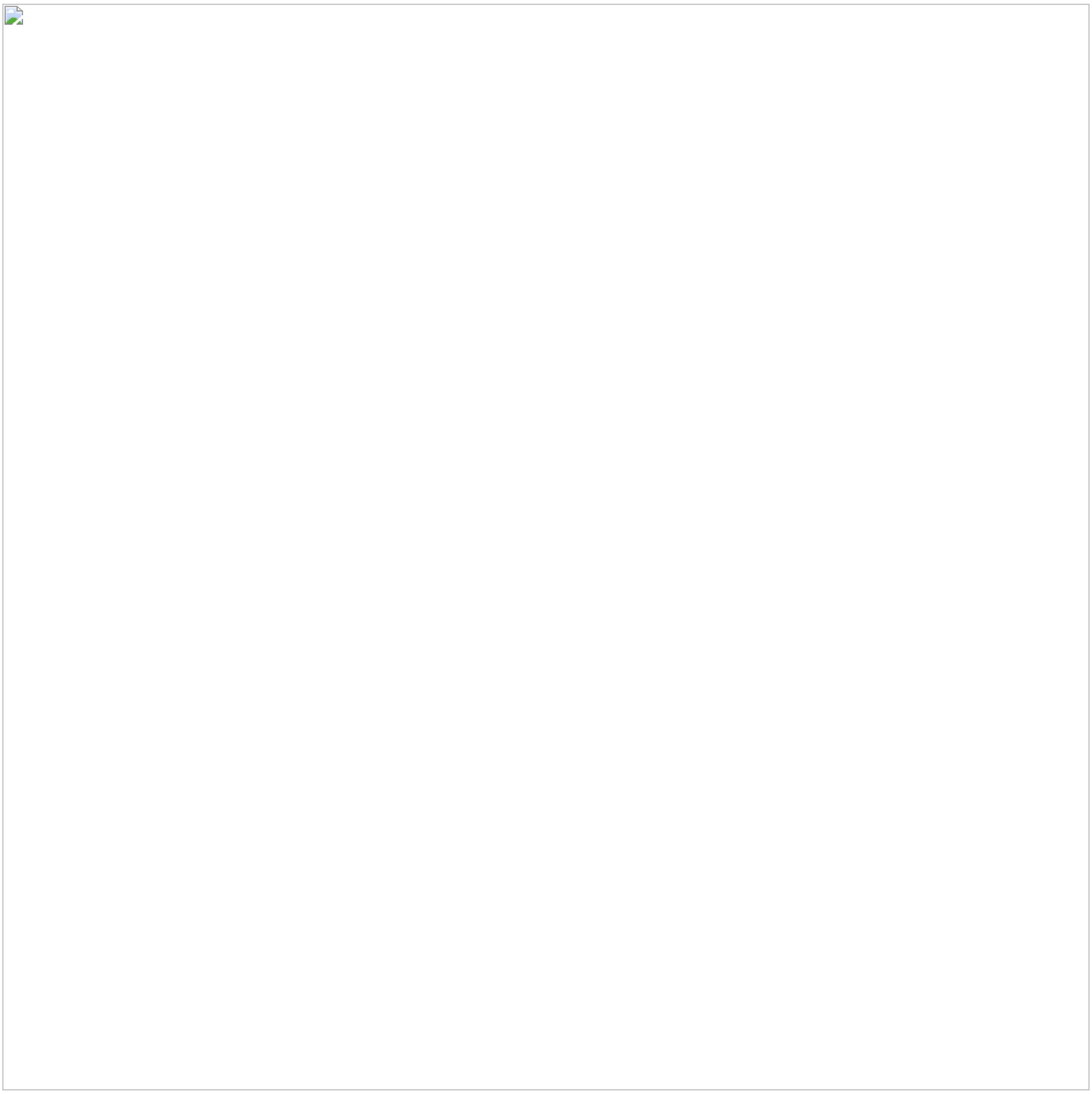


Source: Crowe analysis, March 2023

For resource-based constrained delegation, the service ticket does not need to be marked as forwardable, but service B must be configured to accept delegated access from service A. That is, service A must be included in service B’s ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity attribute.

In Exhibit 3, service A uses S4U2proxy to access service B as user P. Service A uses a nonforwardable TGS ticket, granting user P access to service A as proof of authentication to request a TGS ticket for service B as user P.

Exhibit 3: Service access with resource-based constrained delegation and S4U2proxy



Source: Crowe analysis, March 2023

Modification of the TrustedToAuthForDelegation and ms-DS-Allowed-To-Delegate-To properties used to configure constrained delegation requires the SeEnableDelegationPrivilege. This privilege is granted only to domain administrators by default, but it could be manually granted to other users. However, services can configure their own ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity attribute to accept resource-based constrained delegation from other services.

Weaknesses

While there are many avenues for escalating access using constrained delegation and resource-based constrained delegation, three weaknesses in particular are exposed across multiple attack types.

First, in constrained delegation, the service ticket received from S4U2proxy by service A can be modified to access any target service run by the same user as service B because there is no integrity check for the field specifying the target service. Second, in resource-based constrained delegation, any account can request a service ticket on behalf of another user with S4U2self that can be used to access a different target service as the user with S4U2proxy. Third, services can begin accepting resource-based constrained delegation by altering their own ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity attribute.

Following are descriptions of two different attacks and how resource-based constrained delegation and constrained delegation can be compromised. These attacks can be accomplished using Impacket, mitm6, and an Abstract Syntax Notation One (ASN.1) editor. While variations of these attacks can be performed using other tools, such as Rubeus, the attacks presented here have an advantage because they can be run from an attacker's machine within a target network. They do not require executing tools locally on legitimate machines within the target network, which might increase the likelihood of detection by endpoint detection and response (EDR) or other security technologies.

Attack 1: Gain a foothold with resource-based constrained delegation

Unauthenticated attackers can use resource-based constrained delegation to establish an initial foothold within a target environment where Windows New Technology LAN Manager (NTLM) is used for network authentication. Internet Protocol version 6 (IPv6) and Web Proxy Auto-Discovery (WPAD) are used in the following attack, but any means of coercing NTLM authentication over the Hypertext Transfer Protocol (HTTP) or Lightweight Directory Access Protocol (LDAP) is sufficient.

Prerequisites

- IPv6 and WPAD are allowed on the target network but not used. In general, this is the default configuration.
- NTLM authentication is allowed on the target network.
- The MS-DS-Machine-Account-Quota attribute, which allows accounts to join additional computers to the domain, is set to a non-zero value, the default.
- There exists at least one LDAP endpoint where LDAP signing is not required or LDAP over SSL (LDAPS) endpoint where Extended Protection for Authentication (EPA) is not required. LDAP signing and LDAPS EPA are not required by default.

Steps

1. Leverage IPv6 to become the domain name system (DNS) server of any target machine (machine T) and coerce NTLM authentication from the machine's computer account by serving it a malicious WPAD configuration.
2. Relay the coerced authentication to a vulnerable LDAP or LDAPS endpoint.
3. Using the relayed authentication, create an attacker-controlled computer account (account A) and configure machine T to accept delegated access from account A.
4. Use account A to request a service ticket for itself as a domain administrator with S4U2self. The ticket returned will not be forwardable, but it can still be used with S4U2proxy for resource-based constrained delegation.
5. Use the first service ticket to request a second service ticket to access machine T as a domain administrator using S4U2proxy.

Results

The service ticket grants the attacker access to machine T as a privileged user. Additionally, the attacker has established persistent, privileged access to machine T by repeating steps 4 and 5 at any time.

Attack 2: Compromise a target host with constrained delegation

After performing the previous resource-based constrained delegation attack or another attack that grants access to an authenticated user, an attacker can use constrained delegation to gain access to a high-value target machine. The following attack targets a service run by the computer account for a target machine; however, the attack can be performed by targeting any service run by a user with elevated access to the target machine that also meets the following prerequisites.

Prerequisites

- Access is available to any low-privilege account (account A).
- There exists an account (account B) with a service principal name (SPN) that can delegate access using constrained delegation to a target service run by the computer account on a machine that is a high-value target (machine T).

Steps

1. Use account A to perform an LDAP query to identify an account B that can delegate access to any service run by the computer account for machine T.
2. Gain access to account B using the previous resource-based constrained delegation attack or another attack providing account compromise.
3. Use account B to request a service ticket for B as a domain administrator with S4U2self. The returned ticket must be forwardable for constrained delegation.
4. Use the first ticket to request a second service ticket to access the service on machine T that is configured for constrained delegation as a domain administrator with S4U2proxy.
5. If necessary, alter the service field of the second ticket with an ASN.1 editor to gain access to any service on machine T.

Results

As in the first attack, the service ticket grants the attacker access to any service on machine T as a privileged user. Additionally, the attacker has established persistent, privileged access to machine T by repeating steps 4 and 5 at any time.

Remediation

Attackers can exploit constrained delegation and resource-based constrained delegation to establish initial, elevated, and persistent access within an Active Directory environment. However, resource-based constrained delegation is the most secure variety of delegation available in Active Directory, and delegation is a useful feature in many environments.

The following remediation steps provide a reasonable, but not impenetrable, defense against attacks on constrained delegation and resource-based constrained delegation:

- Identify and audit existing delegation configurations using Impacket or the Microsoft PowerShell™ Active Directory module.
- Assign SeEnableDelegationPrivilege only to users that require the ability to configure delegation for services, such as domain administrators.
- Configure discretionary access control lists (DACLs) according to the principle of least privilege. Any user that can modify the ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity attribute of a computer object can allow themselves to impersonate users on that computer object with resource-based constrained delegation.
- Use resource-based constrained delegation in place of constrained delegation where possible. This creates a one-to-one mapping of a service allowed to impersonate users to another target service. Constrained delegation creates a less restrictive one-to-many relationship.
- Mark accounts with elevated access as sensitive for delegation, including computer accounts for domain controllers, to prevent services from requesting service tickets for the relevant accounts using S4U2proxy, but not S4U2self.
- Add sensitive accounts to the Protected Users security group to prevent services from requesting service tickets for the relevant accounts using S4U2proxy, but not S4U2self. Computer and service accounts should not be added to this group due to adverse effects on their functionality.
- If resource-based constrained delegation is not used, add an access control entry (ACE) to the DACL to explicitly permit only necessary accounts to write to the ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity and TrustedToAuthForDelegation attributes.
- If constrained delegation is not used, add an ACE to explicitly permit only necessary accounts to write to the ms-DS-Allowed-To-Delegate-To attribute.
- Apply NTLM relay mitigations to prevent attackers from using NTLM relay attacks to set up vulnerable resource-based constrained delegation configurations.
- Monitor S4U2self and S4U2proxy calls as part of requests for service tickets (Event ID 4769) from devices that are not configured by IT for delegation, because unexpected delegation from such devices might indicate compromise.
- Add an ACE to the system access control list to generate an event (ID 5136) when changes are made to directory service objects, including changes to the ms-DS-Allowed-To-Act-On-Behalf-Of-Other-Identity or ms-DS-Allowed-To-Delegate-To attributes.

Hardening networks

Constrained delegation and resource-based constrained delegation are more secure alternatives to unconstrained delegation. Resource-based constrained delegation offers the most security and granular control over permissions for delegation, and it should be used for new configurations of delegation. However, any implementation of delegation increases the attack surface of a network and might allow attackers to elevate access and establish persistence. Organizations should implement delegation only when necessary for business functions, prefer resource-based constrained delegation over other varieties, and harden delegation configurations to mitigate the risk of attacks.