Delegate to the Top: Abusing Kerberos for arbitrary impersonations and RCE

Matan Hart

Abstract

In Windows networks, delegation allows a service to impersonate a user or computer in order to access resources throughout the network. While it being used in almost every enterprise, it is still one of the most confusing and least-understood mechanisms in Kerberos and Active Directory. In many cases, the accounts that are running applications and services which are trusted for delegation are misconfigured and, thus, ripe for exploitation.

This paper will review what delegation is, what types of delegation exist, and how they are used in enterprises. Risks associated with constrained delegation and SPNs will be introduced revealing how an attacker can impersonate another user and elevate privileges by exploiting built-in functionalities in Kerberos delegation, enabling remote execution with arbitrary user through unexpected services. Finally, some guidance will be shared that will allow tightening up of delegation rights to minimize risk.

Introduction

Delegation is the assignment of responsibility or authority to another identity to carry out specific activities. It is one of the core concepts of management, shifting the decision-making authority from one organizational level to a lower one and empowering a subordinate to take responsibility.

In Microsoft Windows Active Directory, delegation (aka impersonation or Kerberos double-hop) is the act of an application or service getting Kerberos tickets to gain access to resources on a remote machine on behalf of a different user. The identity that is trusted for that delegation is the service account that is running the application. In a typical scenario, the impersonating account would be a service account assigned to a web application or the computer account of a web server. The impersonated account would be a user account requiring access to resources (e.g. data in an SQL database) via a web application. In this scenario, SQL server would be accessed by the impersonating (service account) account, however access would be under the context of the impersonated (user) account.

Kerberos Delegation

There are three flavors of delegation in AD Kerberos implementation since Microsoft introduced Server 2003:
1. Unconstrained Delegation (TGT forwarding) – Enables a service to use a client's TGT to itself to request another ticket for delegation.
2. Constrained Delegation (S4U2Proxy) – Enables a service to use a client's service ticket (TGS) to itself to request another ticket for delegation.
3. Protocol Transition (S4U2Self) – Enables the service to acquire a service ticket (TGS) from an arbitrary principal to itself (the service is trusted to have authenticated the principal).

The reason why the Kerberos constrained delegation (KCD) extension was introduced in Windows Server 2003 can be best explained by describing the limitations in the Windows 2000 implementation of Kerberos delegation. In the Windows 2000 Kerberos delegation model, the Kerberos Key Distribution Center (KDC) does not limit the scope of services to which a Kerberos principal's identity can be delegated. In other words, after a service account is trusted for delegation, it can request service tickets on behalf of an authenticated user to any other service. This delegation method does not provide precise mechanisms for an

application to specify a subset of services that it determines to be trustworthy for delegation.

Essentially, applications are exposed to broader impersonation risks that may span across resource domains that have different levels of security policy requirements; some of the security policies may not be as strict as the applications' security requirements. From the domain administrator's point of view, it is too risky to enable unconstrained Kerberos delegation in the enterprise because there is no way to exclude untrusted servers from participating in delegation. With constrained delegation, domain administrators can configure service accounts so that they delegate only to specific sets of services.

Constrained delegation lets you limit the back-end services for which a front-end service can request tickets on behalf of another user. A common example of constrained delegation is the web-browser-to-IIS-to-SQL-Server scenario. In this scenario, a user navigates to a web-based reports server hosted on Microsoft IIS, which retrieves data using an authenticated connection to a Microsoft SQL Server system. Using constrained delegation, you can limit the IIS server (the front end) so that it can authenticate the user only to SQL Server (the back end) and no other service or application.

The protocol transition extension allows a service to obtain a Kerberos service ticket to the service on behalf of a Kerberos security principal. No user credential is required for the transition. Applications may transition into Kerberos even though the actual authentication is done via another authentication system such as NTLM, Radius, RSA SecureID, PKI/Certificates and other OTP systems.

If the service has the necessary impersonation privileges in Windows, when the service uses this token to impersonate the user and request a Kerberos service ticket to another service, the service ticket issued - which is to the requesting service - is mapped to the user token. The service may use the service ticket obtained through protocol transition to obtain service tickets to other services and thereby delegate the credentials if the account under which the service is running is configured correctly to use the Kerberos constrained delegation extension.

Protocol transition is commonly used to connect across a firewall or proxy software using one authentication method, such as NTLM, and transition that domain user to using Kerberos authentication for further actions that take place within the corporate network. Web applications (such as Outlook Web App and

SharePoint) and VPN vendors often require constrained delegation with protocol transition to enable single sign-on (SSO) and solve issues raised by federation of domains across organizational boundaries (both external and internal).

Server 2012 introduces resource-based constrained delegation that addresses many of the shortcomings that exist with the previous constrained delegation model. This new implementation of constrained delegation removes the dependencies on SPNs for delegation configuration, enables the resource administrator to own the delegation experience, and increases the scope of delegation. Constrained delegation in Server 2012 introduces the concept of controlling delegation of service tickets using a security descriptor rather than an allow list of SPNs. This change simplifies delegation by enabling the resource to determine which security principals are allowed to request tickets on behalf of another user.

Service Principal Name

Kerberos uses SPNs to identify the security principal responsible for running an application or service. This enables the Key Distribution Center (KDC) to encrypt tickets with the correct key so that the security principal (running the service or application) can decrypt the service ticket upon receiving the AP_REQ. This design requires that SPNs registered on security principals be unique for the AD forest. An SPN registered on multiple security principals will cause authentication to fail.

Constrained delegation allow impersonation based on SPN which requires an additional configuration of adding an allow list of services (<service_class/host>) that are trusted for delegation. This list of SPNs represent the back-end services to which a front-end service is allowed to request tickets on behalf of the user. Those SPNs are stored in the AD under the msDS-AllowedToDelegateTo attribute of the principal which the application or service on the front-end server runs.

Constrained Delegation Risks

For an attacker looking for opportunities to elevate privileges inside the network, service accounts that are trusted for delegation can be extremely useful. Contrary to human accounts, those accounts are likely to be privileged, unmanaged and vulnerable, making them highly lucrative.

The fact that applications that are trusted for constrained delegation with protocol transition have the ability to make a claim to the KDC that it successfully authenticated a user regardless of whether it actually did so (or even the user actually ever attempted to authenticate to the application) allows an attacker to impersonate arbitrary users (if they are not forbidden for delegation) to the delegation servers. Even without protocol transition, an attacker can still reuse credentials cached in the impersonating server (front-end) to access delegated server (back-end) on behalf of users who were previously authenticated.

Moreover, different protocols and services may use the same SPN which means they share the same service ticket for authorization. In addition, the SPN is mapped to the machine and cannot be limited to an application (it's possible to specify a port but it's rare). For example, both IIS and WinRM (PowerShell remoting) services identify clients based on the HTTP service class (HTTP/WebServer). In other words - owning the web server's service account may allow RCE with arbitrary user to the allowed IIS delegated servers.

Many respected security and IT vendors recommend to configure Kerberos constrained delegation with protocol transition, sometimes permitting to delegate access to some sensitive services (LDAP) on the Domain Controller enabling complete domain-level access, which will stay under the radar.

Another important issue regarding the fact that SPNs are used as a security boundary in Kerberos constrained delegation. As mentioned earlier, in the Kerberos authentication protocol, a service validates in inbound service ticket by ensuring that the ticket is encrypted to that service’s symmetric key. The SPN used does not factor into this validation. In fact, the field containing the SPN is part of the unencrypted part of the ticket. This means that services do not use SPNs for authorization decisions since they won't provide any security protection against a malicious client anyway. If service account has registered two SPNs then their Kerberos service tickets are fully interchangeable. This extends the scope of

delegation from per service (SPN) to per principal or the secret key used to encrypt the ticket.

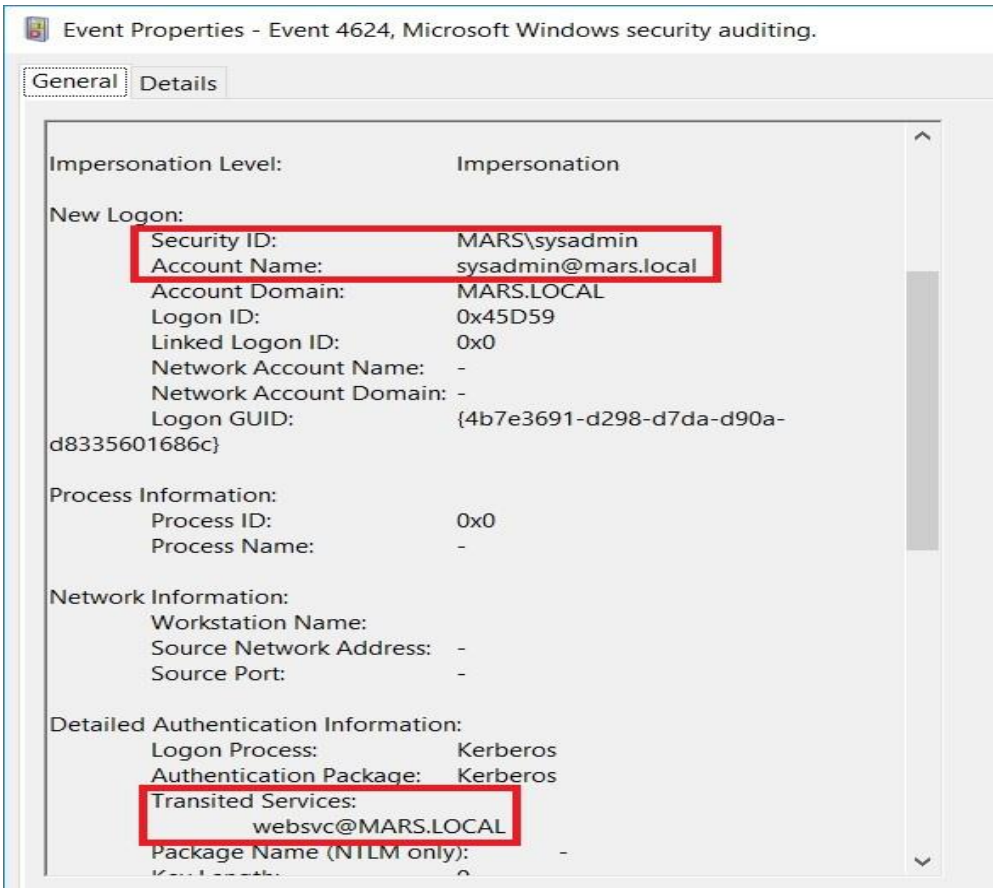
As a result, the attack surface is greatly increased in case the service account has registered several SPNs. An attacker could gain unlimited access to the back-end server if the allowed SPN is registered under the computer account, even if the delegating account (front-end) is restricted to specific service. In resource-based constrained delegation it does not even matter, since the back-end does not limit the scope of access when the identity (front-end) is trusted for delegation.

Detection

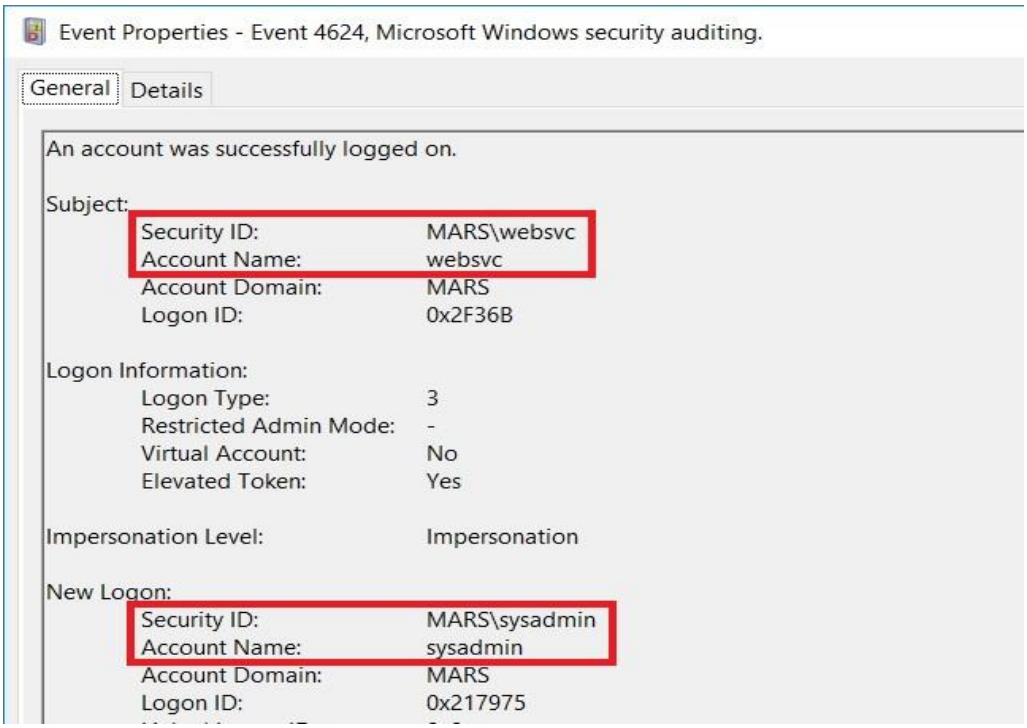
Detecting delegation activity requires identification of where delegation is occurring. From those in-scope systems where delegation occurs, it possible to identify which accounts where delegating credentials and which users' accounts were actually delegated. This can be important for both proactively assessing the types of accounts that are performing delegation, as well as forensically analyzing accounts that may have been exposed during a compromise. Mapping accounts trusted for delegation can be easily achieved by querying the Active Directory.

Windows Event Logs

Determining exactly which accounts were delegated in the event logs was not really possible prior to Windows 8 and Server 2012. There was no simple way to determine if a delegation token was created for a user when they logged onto a machine. Fortunately, that has been resolved starting with Windows 8 and Server 2012. Event ID 4624 now specifies the impersonation type that was created when a user logs on. Below is an example of event ID 4624 after delegation occurred on a front-end server running Windows 2016:

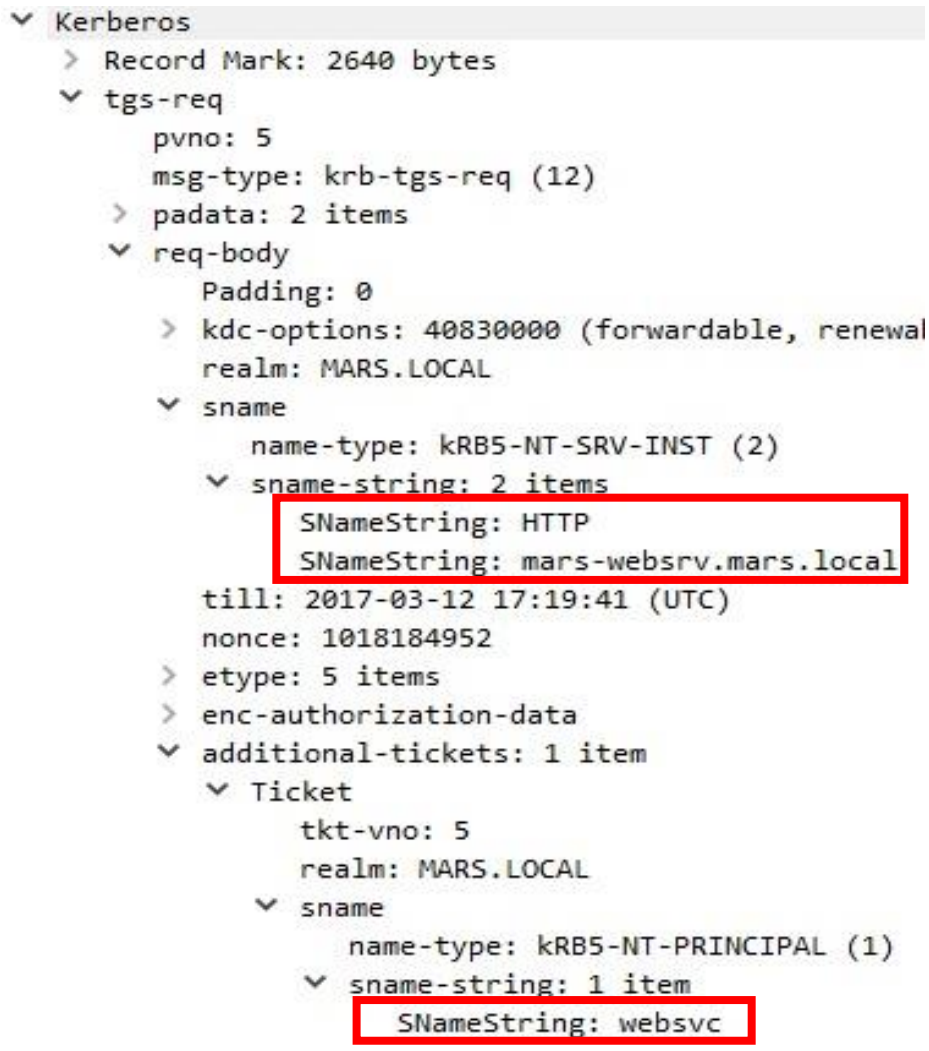


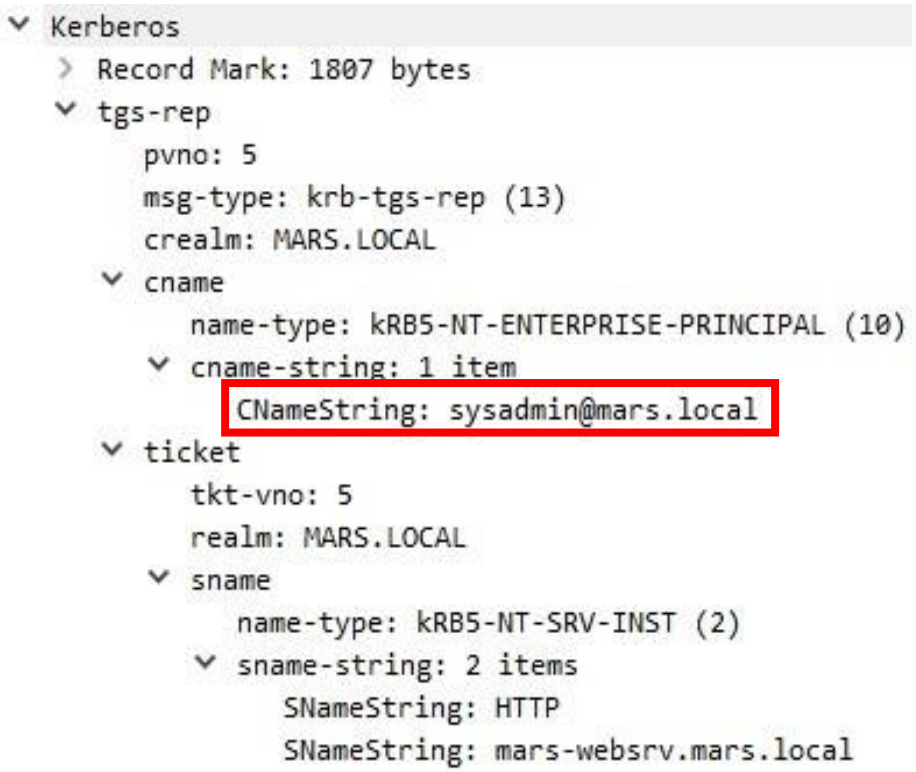
And Here's an example of Event ID 4624 triggered in the back-end server after the delegation occurred:



Network Traffic

It possible to identify delegation activity by capturing the S4U2Proxy Kerberos traffic which is performed in the TGS_REQ and TGS_REP message exchange. Correlating both messages can effectively pinpoint the delegating account, the service (SPN) and the delegated account. Below is an example of a network traffic capture of a TGS_REQ message followed by a TGS_REP message:





Mitigation

Hardening delegation rights to reduce risk can be sometimes tricky. First, unconstrained delegation should be avoided at all cost. With the resource-based constrained delegation introduced in Server 2012, which enables delegation across domains and forests, there is no real justification to configure unconstrained delegation (except for compatibility and laziness).

When using constrained delegation, the attack surface is determined by the allowed list of SPNs (msDS-AllowedToDelegateTo [A2D2]). It's possible to limit the exposure by configuring unique service accounts and SPNs instead of using the computer account or built-in SPNs. The best practice will be setting a dedicated account to run the back-end service and registering a custom SPN (including port) for the service under the account. This significantly reduces the likelihood of an attacker being able to extend the attack surface beyond the service itself.

Conclusion

Delegation is a very powerful and useful feature, though it comes with an equal amount of risk. Kerberos Constrained Delegation is designed to provide a mechanism to restrict what can be accessed by the impersonating account, yet SPNs and security descriptors are not precise enough to effectively isolate the attack surface. While limiting the potential risk exposed by impersonation to some extent, Kerberos constrained delegation without proper hardening, does not fully mitigate all escalation opportunities by a determined adversary that has already compromised the delegation server.

References

- [MS-KILE]: Kerberos Protocol Extensions
  http://msdn.microsoft.com/enus/library/cc233855.aspx
- [MS-SFU]: Kerberos Protocol Extension
  https://msdn.microsoft.com/en-us/library/cc246071.aspx
- Kerberos Constrained Delegation Overview
  https://technet.microsoft.com/en-us/library/jj553400(v=ws.11).aspx
- Service Principal Names
  https://msdn.microsoft.com/en-us/library/ms677949(v=vs.85).aspx
- Windows Event ID 4624
  https://technet.microsoft.com/en-us/itpro/windows/keep-secure/event-4624