Domain Trusts: An Exploitation Playbook

redfoxsec.com/blog/domain-trusts-exploitation-playbook

Shashi Kant Prasad June 12, 2024

- June 12, 2024
- Active Directory
- Shashi Kant Prasad

Understanding domain trusts in **Windows Active Directory (AD)** is essential for modern cybersecurity professionals. These trust relationships are the backbone of communication between domains, enabling seamless authentication and resource sharing across an organization. However, the same mechanisms that make trusts powerful can also turn them into dangerous gateways if misconfigured or abused.

When exploited, domain trusts can allow attackers to **traverse organizational boundaries**, moving laterally across networks, escalating privileges, and potentially achieving a **wide-reaching compromise** of critical systems. This makes them a prime target in advanced persistent threats (APTs) and red-team engagements.

In today's threat landscape, mastering the mechanics of domain trusts is no longer optional — it's a defensive necessity. By understanding how trusts are built, how they can be manipulated, and how to monitor them effectively, defenders can close one of the most overlooked yet critical gaps in enterprise security.

In this blog, we will explore the fundamentals of domain trusts, the most common exploitation techniques used by adversaries, and practical mitigation strategies every defender should know to protect Active Directory environments.

What Are Domain Trusts?

Domain trusts are relationships established between Active Directory (AD) domains that enable users, groups, and computers in one domain to seamlessly access resources in another. At their core, trusts are built on authentication and authorization mechanisms that reduce administrative overhead and improve collaboration across different parts of an organization.

When properly configured, domain trusts greatly enhance **operational efficiency**. For example, they allow centralized identity management, unified access control, and smoother integration after mergers or acquisitions. In large enterprises, trusts are indispensable for managing complex networks with multiple domains and forests.

However, these same conveniences can also introduce **serious security vulnerabilities** if left unchecked. Misconfigurations, over-permissive trust settings, or a lack of monitoring can effectively provide attackers with “bridges” to move laterally across domains. A compromise in one environment could cascade into others, leading to privilege escalation and even full enterprise compromise.

Types Of Domain Trusts

- **Bi-directional Trusts**: Users in both domains can authenticate and access each other’s resources.
- **Inbound Trusts**: The target domain trusts the originating domain, granting access to its resources.
- **Outbound Trusts**: The originating domain trusts the target, giving its users access to originating domain resources.
- **Transitive Trusts**: Extend trust transitively (e.g., if A trusts B, and B trusts C, then A trusts C).
- **Non-Transitive Trusts**: Limit trust to only the directly defined domains.

Directionality and transitivity define how attackers can navigate a trust network, making them critical in exploitation planning.

The Role Of Active Directory Forests In Domain Trusts

An **Active Directory forest** is the highest-level logical container within AD, comprising a collection of domains that share a common schema, configuration, and global catalog. Forests define the **security and administrative boundary** of an organization’s AD

infrastructure, and they rely on **implicit, bidirectional, transitive trusts** — known as intra-forest trusts — to connect the domains within them.

These trusts are established automatically when a new domain is created inside a forest, ensuring seamless authentication and resource sharing across the environment. While this design enhances operational efficiency, it also expands the potential attack surface.

Adversaries who compromise a single domain within a forest can often **leverage these trust relationships to escalate privileges**. A common tactic involves moving laterally from a compromised child domain toward the **forest root domain**. Since the forest root ultimately controls policies, schema, and trust configurations for the entire forest, gaining control here effectively means achieving **full enterprise compromise**.

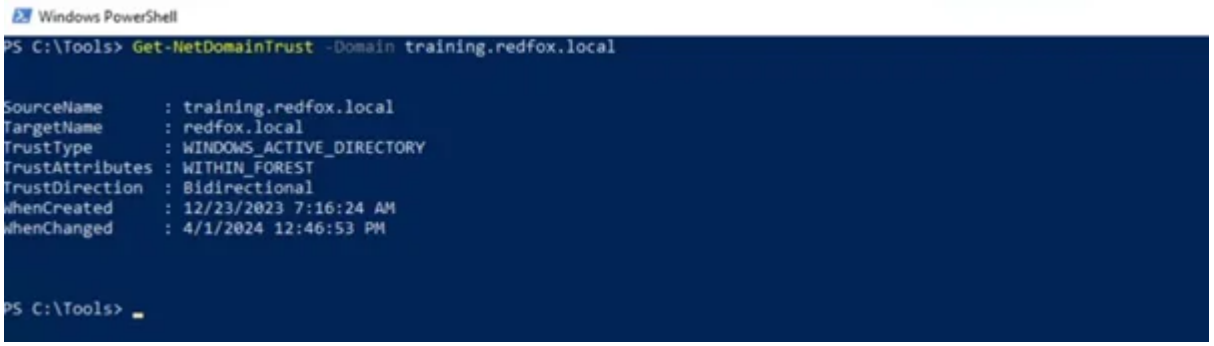
This makes forest-level security a critical focus area for defenders. Hardening domain controllers, monitoring authentication flows, and limiting unnecessary privileges across trusts are all vital steps in preventing attackers from abusing forest relationships.

Enumerating Domain Trusts

The initial step to exploiting domain trusts involves gathering intel on trust relationships within your target environment using various techniques and tools that enumerate and visualize trust architecture.

Trust Enumeration

Powerview, a popular PowerShell-based tool, offers an effective set of functions for enumerating domain trusts. The Get-DomainTrust cmdlet makes this easier; providing information about all trust relationships associated with your current domain as well as trust type attributes and directions. Running this cmdlet in our domain, we see that the domain training.redfox.local has a bidirectional trust relationship with the parent domain redfox.local.



A partial replica of the Active Directory Forest, the global catalogue can also be leveraged to map trust relationships within an environment. By querying it, you can gain a more holistic picture of trust mesh across multiple domains.

Exploiting Intra-Forest Trusts: "Trustpocalypse"

One of the most dangerous attacks inside a forest is “**Trustpocalypse**.” It abuses the **SIDHistory** attribute, originally intended to preserve group memberships during migrations.

The SIDHistory Vulnerability

At the core of Trustpocalypse lies its exploit of Windows 2000 Active Directory’s SIDHistory attribute for user migration between domains. This feature allows existing group memberships (SIDs) of users who have moved between domains to be maintained even after migration has occurred.

Importantly, the SIDHistory attribute is respected between forest domains, as SIDs are not stripped out during cross-domain referrals. This means an attacker who gains control of one of these child domains can alter a user account’s SIDHistory to include SIDs belonging to privileged groups like “**Enterprise Admins**,” thus giving them elevated access to the forest root domain.

This attack requires creating a “**Golden Ticket**“, or **Kerberos ticket-granting ticket (TGT)**, that has been signed off on by the target domain’s krbtgt account and signed by any user within it – even those with elevated privileges! A DCSync attack can quickly obtain this hash before creating this TGT with SIDHistory that impersonates anyone within its forest, including users with higher privileges.

Exploiting Inter-Forest Trusts: Forged Inter-Realm Tickets

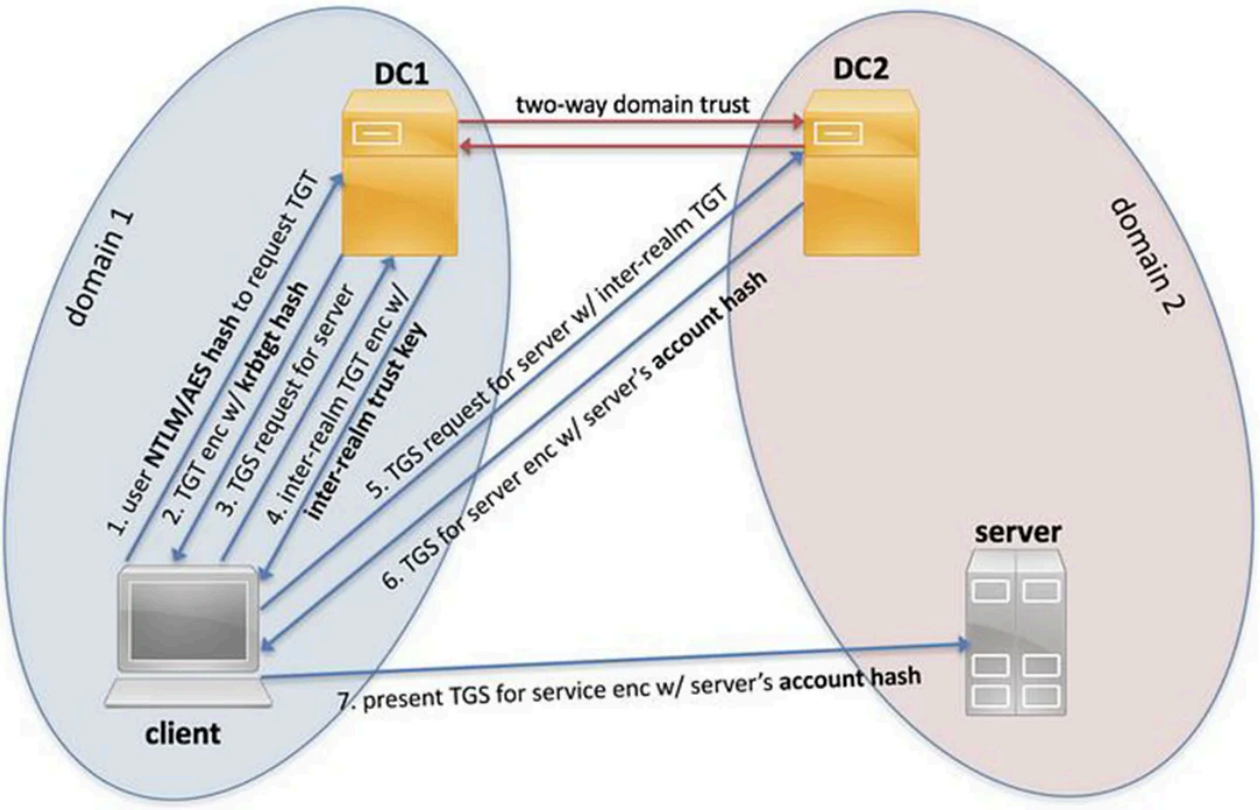
Although Trustpocalypse attacks can be used to compromise intra-forest trust relationships, exploiting cross-forest trust relationships requires taking a different approach due to extra security controls like SID filtering that may obfuscate their attack surface and complicate exploit attempts. Even though this is the preferred attack for cross-forest domains, it can also be performed to compromise the parent domain in our scenario: redfox.local

Understanding Inter-Realm Trust Tickets

When users attempt to access resources in another domain, their domain controller issues an inter-realm ticket-granting-ticket (TGT), signed with an encryption key shared between domains.

Foreign domains then use their trust key agreement to validate and decrypt referral tickets sent from different domains, authenticating user identities and granting access when appropriate.

Let us now see how this works:

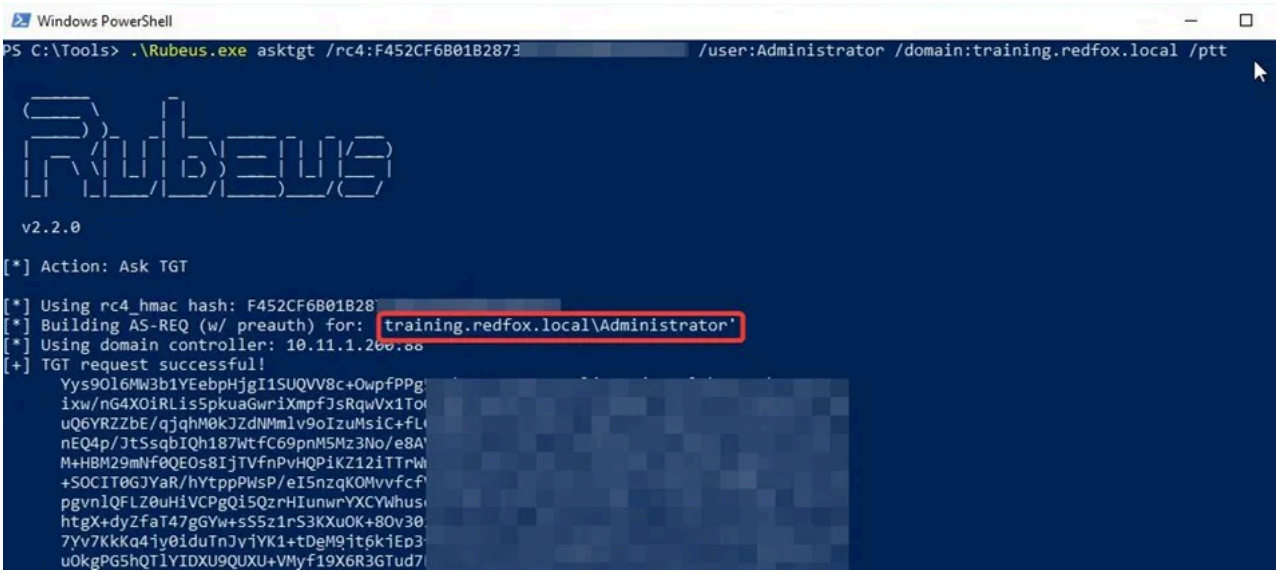


1. The user in Domain 1 sends a plaintext credentials AS request to their domain controller (DC1) to obtain a TGT.
2. DC1 verifies the user's credentials, creates a TGT for the user, and encrypts it with a key derived from the user's NT hash (the hash of their password). It sends this TGT back to the user.
3. When the user needs to access a service (e.g. SQL Server) in the trusted Domain 2, they send a Ticket Granting Service (TGS) request to DC1 using their TGT, requesting a service ticket for the service in Domain 2.
4. DC1 recognizes that the requested service is in a trusted domain. It cannot issue a service ticket directly, so instead, it constructs an inter-realm TGT. This is a referral ticket encrypted with the inter-realm trust key that Domain 1 shares with the trusted Domain 2.
5. The user receives this inter-realm TGT from DC1 and sends it in a TGS request to a DC in Domain 2 (DC2), requesting a service ticket for the desired service.
6. DC2 uses the shared inter-realm trust key to decrypt and validate the inter-realm TGT from Domain 1. If valid, DC2 generates and sends back a service ticket for the requested service, encrypted with the service account's NT hash in Domain 2.
7. Finally, the user can present this service ticket to the service (e.g. SQL Server) in Domain 2 to authenticate and gain access.

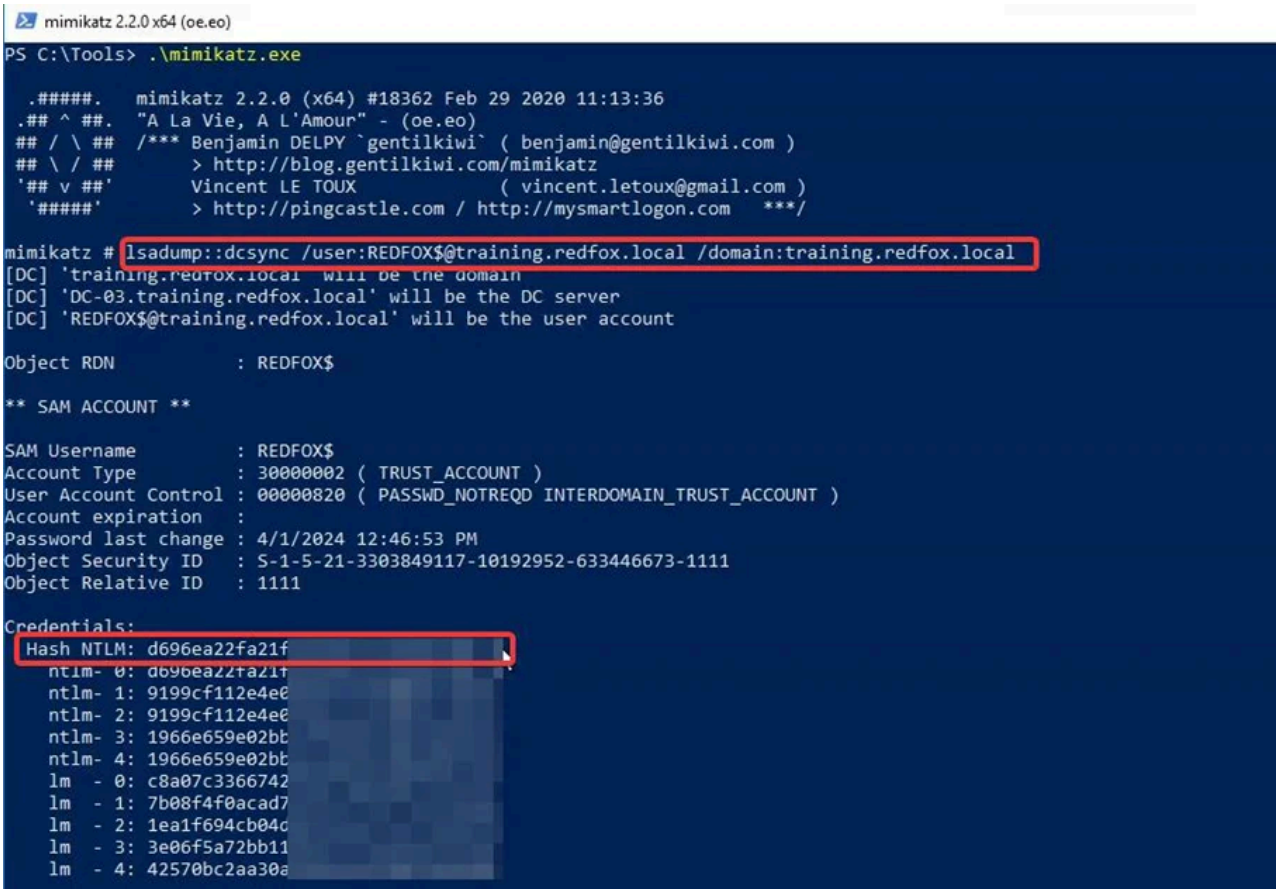
However, if an attacker compromises the inter-realm trust key, they can forge the inter-realm referral tickets themselves, bypassing the need to first authenticate to the original domain.

Let us now extract the inter-realm trust key from our domain training.redfox.local.

With the hash of the compromised domain admin ADMINISTRATOR, let us create a session for that user using Rubeus by the overpass-the-hash method.



Once we have the domain Admin ADMINISTRATOR's session in the training.redfox.local domain, we extract the trust key. We can use mimikatz's dcsync to do this. The trust key is the trust name with a dollar sign at the end. So, in our case, it will be REDFOX$. We would also need to specify the domain.

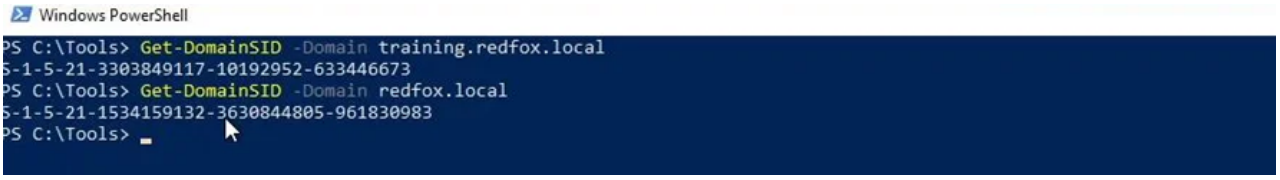


Here, we've successfully extracted the trust key, essentially the NT hash of the account.

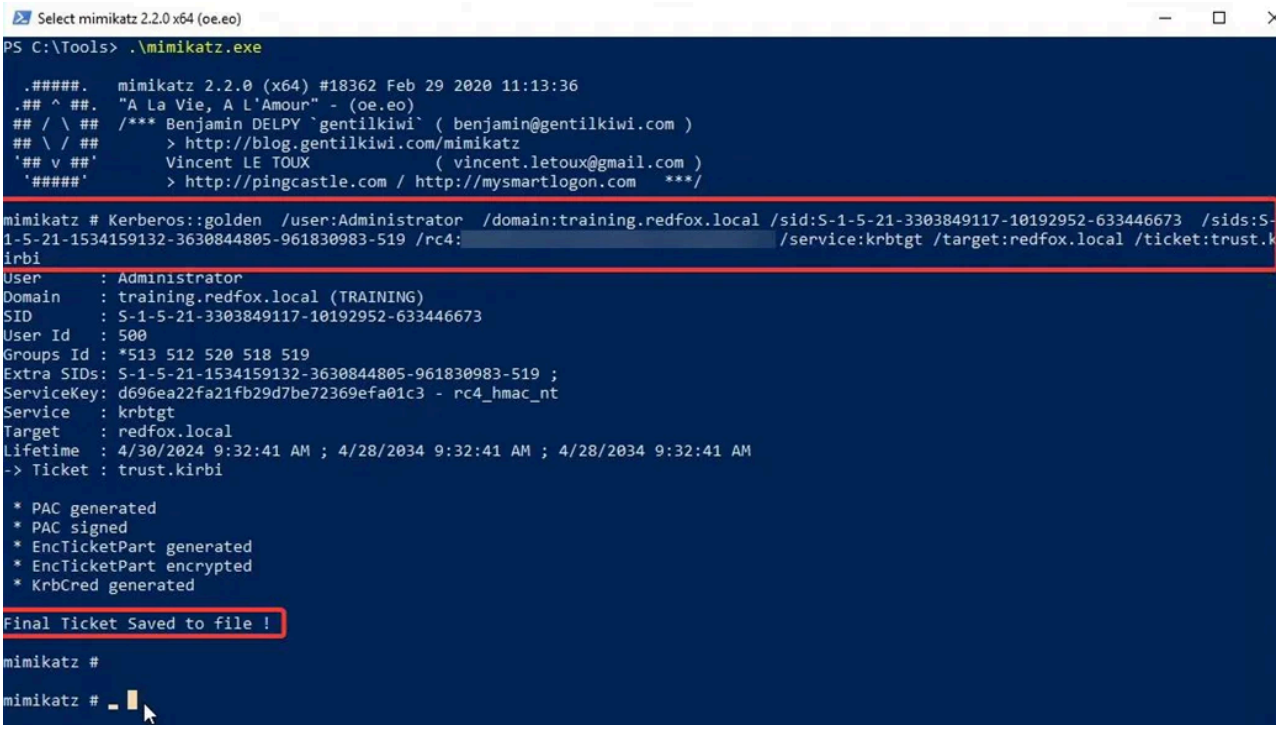
Forging Inter-Realm Trust Tickets

An attacker with the inter-realm trust key can forge trust tickets to impersonate any user from their domain in foreign domains. This is useful when they've compromised one domain but lack privileges to access another directly.

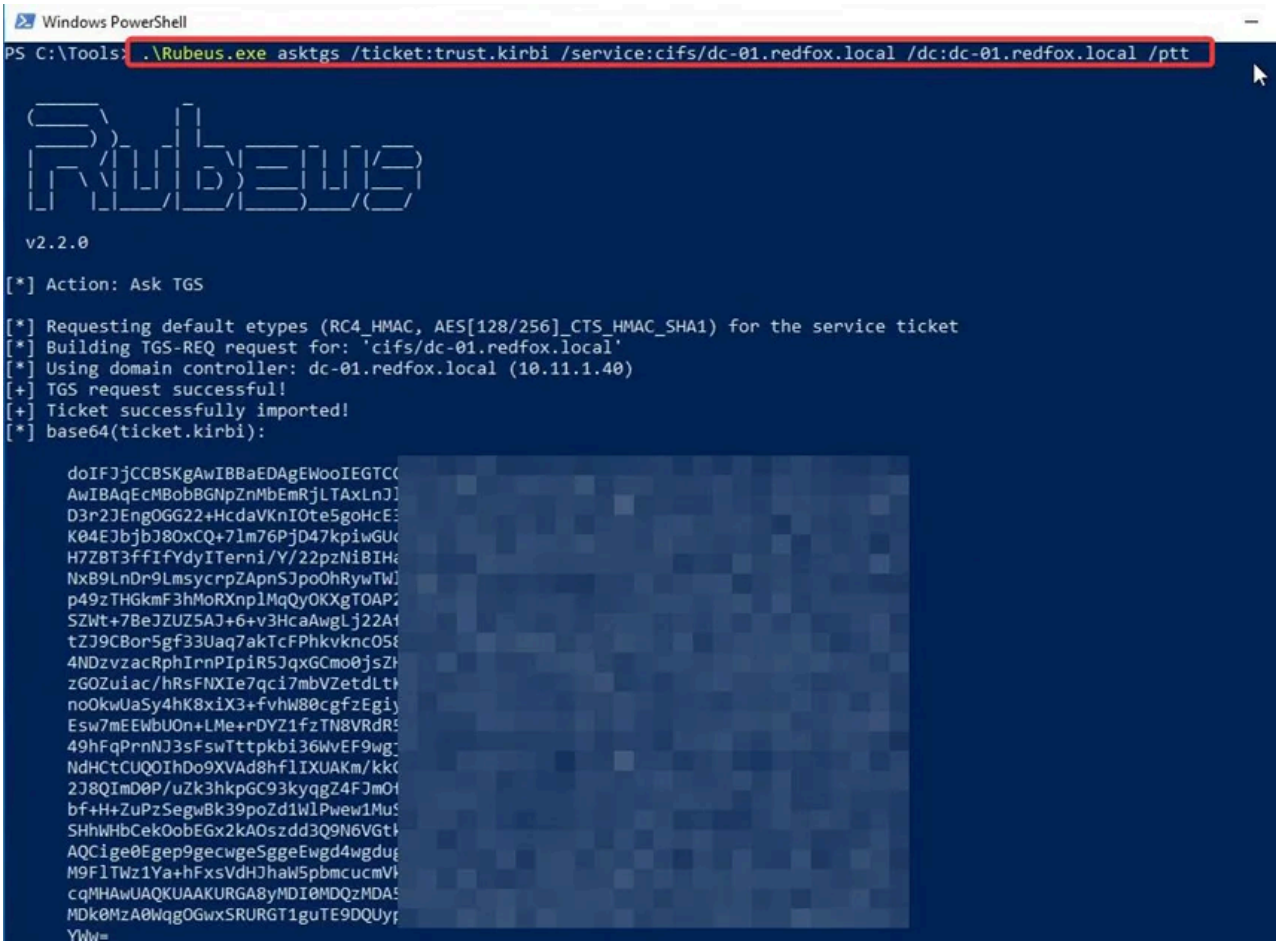
In order to forge the Inter-realm TGT, we would first need the domain SIDs of both domains. We can use the PowerView cmdlet Get-NetDomainSID to get this. We need to specify the domains using the -Domain flag.



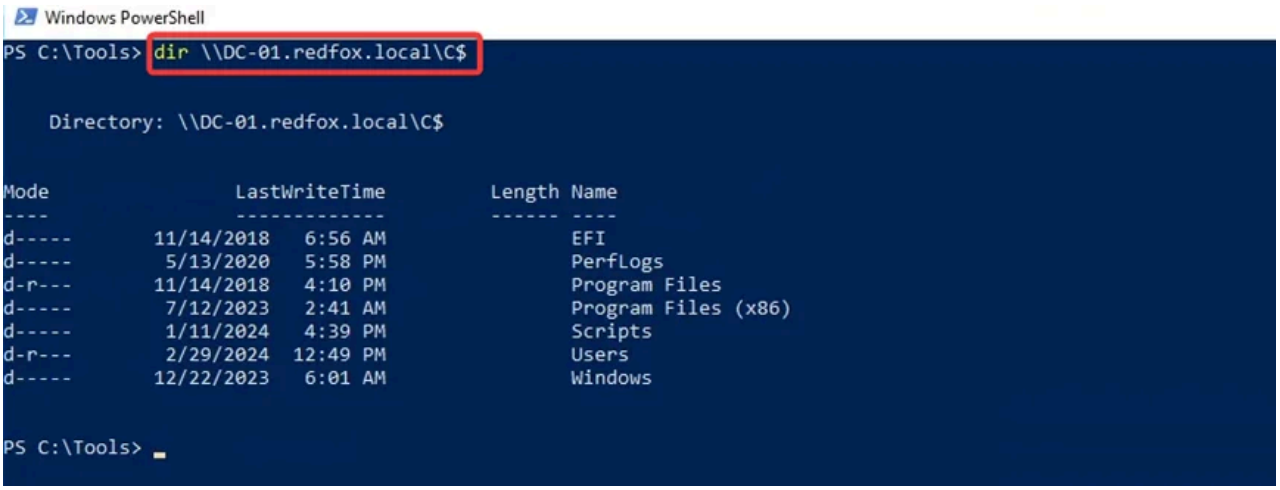
Once we get both the SIDs, we can now craft an inter-realm TGT using mimikatz's "golden" command, to forge a TGT for the domain admin ADMINISTRATOR for the service krbtgt on the domain redfox.local using the trust key. This ticket is then saved to a file trust.kirbi



Now using this, we can try to access to Domain controller DC-01 of the parent domain redfox.local. First, we need to forge a service ticket for the CIFS service on DC-01 using Rubeus with the inter-realm TGT.



Now, we will try to access the C: drive of the domain controller DC-01. We can list its contents, confirming that we can compromise the parent domain redfox.local through this DC.



Mitigation Strategies

Mitigating risks related to domain trust exploitation requires a multifaceted approach, including installing comprehensive security controls and conducting regular audits, in addition to adopting a “Red Forest” architecture.

1. Implementing SID Filtering and Quarantined Domains

Configuring SID filtering and quarantined domains prevents lateral movement of privileged accounts across trust boundaries, reducing the attack surface. By carefully managing which SIDs traverse each trust boundary, organizations can substantially lower their attack surface.

2. Adopting the Enhanced Security Administrative Environment (ESAE)

Microsoft recommends adopting a Security Administrative Environment, known as Red Forest, for Active Directory. This minimizes domain trust vulnerabilities and enhances overall security. By adopting and implementing such an administrative environment, organizations can bolster their overall security posture.

3. Continuous Monitoring and Auditing

Continuous monitoring and auditing of domain trust relationships, group memberships, and ACLs are vital. This helps quickly identify misconfigurations or vulnerabilities and take appropriate action. Automated tools like BloodHound can assist in this effort to assess trust architecture and identify possible attack paths.

Conclusion

Domain trusts in Active Directory, while essential for business operations, expose a complex and often overlooked attack surface. Techniques like **Trustpocalypse** and **inter-realm ticket forging** allow attackers to pivot across domains and forests, elevating privileges dramatically.

Security teams must master trust enumeration, vulnerability detection, and mitigation strategies to safeguard enterprise environments effectively.

Redfox Security is a global network of cybersecurity experts dedicated to helping organizations identify vulnerabilities, strengthen defenses, and foster security awareness. **Contact us** for expert consulting or enroll in our **comprehensive cybersecurity courses**.

PreviousCar Hacking- The New Frontier In Cybersecurity
NextSecurity Advisory – Multiple Vulnerabilities in LB-link BL-W1210M Router

Recent Blog

September 09, 2025
Is APK Decompilation Legal? What You Need To Know
September 06, 2025
When Hackers Hit the Road: The Jaguar Land Rover Cyberattack
September 05, 2025
This Is the Hacker’s Swiss Army Knife. Have You Heard About It?