Exploiting Service Accounts: Silver Ticket Attack

blog.netwrix.com/2022/08/31/impersonating-service-accounts-with-silver-tickets

Jeff Warren

In the first post of these series we showed how an adversary can discover Active Directory service accounts with PowerShell, and the second post demonstrated how to crack their passwords using the Kerberoasting technique. Now let's see how an attacker can exploit a compromised service account using Kerberos Silver Tickets to forge TGS tickets.

Silver Tickets overview

Silver Tickets are forged Ticket Granting Service (TGS) tickets that an attacker created using the password hash of a compromised service account. When, as in our example, an attacker has cracked a service account's password, it might seem that they wouldn't need to forge tickets to act on its behalf.

However, Silver Tickets do offer several benefits, including the following:

- The attacker is not required to authenticate the account to the domain controller to obtain the forged TGS, so they can proceed without creating network traffic and event logs to avoid detection.
- A Silver Ticket can be created for any user account, even fictitious accounts. This allows the attacker to exploit the service account without risking detection, which could result in a password reset and loss of access.
- The Privileged Attribute Certificate (PAC) in the TGS ticket can also be manipulated to elevate the account's access to Domain In most cases, the PAC is not validated against the domain controller when the TGS is provided.

For example, in our previous post, we showed how an attacker could compromise a SQL service account. But that account does not have any access to the databases and cannot log in interactively to the computer, so there isn't a whole lot the attacker can easily do with it. Silver Tickets enable them to elevate its privileges so they can use it to gain full control over all the databases hosted on the compromised SQL Server. Even better for them, they can do this while remaining very difficult to detect.

Silver Tickets: How they work

Here are the steps involved in a Silver Ticket attack.

Step 1. Obtain the password hash of a service or computer account.

In order to forge TGS tickets, an adversary must have the NTLM password hash for either a service account running a service on a computer or the computer account itself. In the previous post, we focused on compromising a service account running the SQL service on a particular host. The benefit of using a service account is that these accounts can be compromised without any elevated domain privileges. In order to compromise a computer account's NTLM hash, an adversary must obtain Administrator rights to that computer. If they are able to obtain the computer NTLM hash, they can forge tickets for any service that runs as the local system account. This could include the Common Internet File System service (CIFS service), which would provide access to all files stored on the computer. Microsoft describes other services that run on Windows operating systems here.

For this post, we will continue the example of the compromised SQL Server service account.

Step 2. Create forged service tickets using Mimikatz.

With the NTLM hash of one or more service accounts, the attacker can create Silver Tickets using Mimikatz. The information required to create these tickets includes:

Domain SID — This can be obtained easily by issuing the command "Whoami /User" from a command prompt and copying the SID value without the Relative ID (RID) at the end.

- **Target**— This is the host, which can be copied from the SPN value. In our case, this is **jefflab-sql02.jefflab.local:1433**.
- **Service**— This is the name of the service that the tickets will be created for. It must be a service running as the compromised service account. Our example will use MSSQLSvc.
- **User**— This is the user that the ticket will be created for. It can be any user account whatsoever, even a user account that does not exist.
- **Groups**— This is the list of groups that will be added to the PAC for the account. By default, it includes Domain Admins, but a custom set of groups can be used instead if needed.

Here is the complete command issued to create a Silver Ticket for MSSQLSvc with Domain Admin rights as a user with the name **FakeUser**.

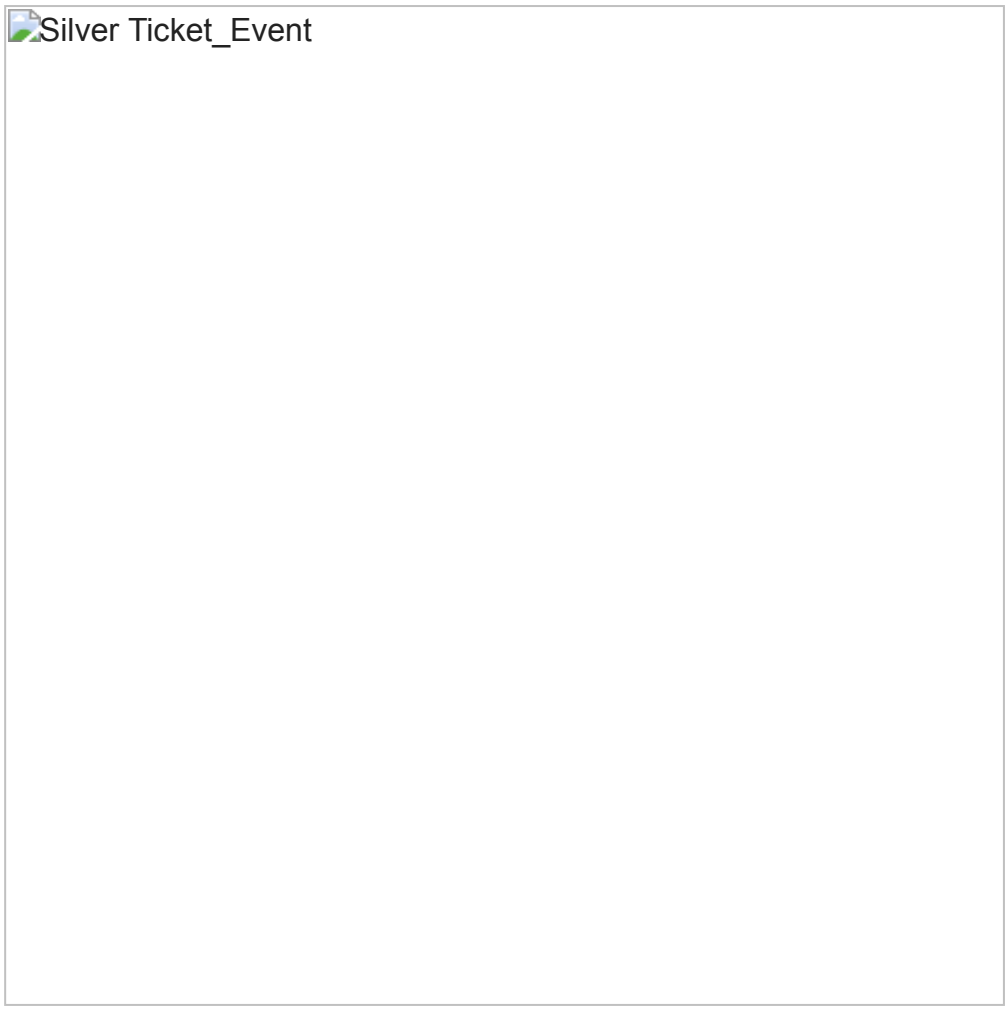
Step 3. Pass the ticket.

The **/ptt** flag is specified to automatically inject the fake ticket into memory.

The attacker now has a forged Kerberos ticket for the FakeUser account:

With that ticket in memory, the adversary just needs a way to issue SQL commands against the target host, which will support passing the ticket. They can use the Sqlcmd.exe utility provided by Microsoft.

As you can see in the above command, the adversary is able to make a SQL connection to the target host, which sees them as JEFFLABFakeUser. Now they are connected to a SQL database with full admin rights as an account that doesn't even exist! That makes it much harder to investigate any actions the attacker performs and to understand how this access was compromised in the first place.



You can see by looking at the SQL server's security logs that it will track logon events as FakeUser as well:

Protecting against Silver Ticket attacks

Detecting Silver Ticket attacks can be very difficult since they bypass the entire TGT portion of authentication and cannot be monitored by looking at domain controller logs. Monitoring for logon anomalies using local logon events, such as the one shown above, can help protect your organization. But the best option is to block these attacks by enforcing proper security over service accounts so they are not compromised to begin with.

Jeff Warren