

Forum:[Hack Forums](#)

Thread: [PowerShell Empire for Pentester: Mimikatz](#)

chirag.8347 - January 6, 2025, 8:15 pm

This article will showcase various attacks and tasks that can be performed on a compromised Windows Machine which is a part of a Domain Controller through PowerShell Empire inbuilt Mimikatz Module.

PowerShell Empire is one of those tools that keep on giving to the Penetration Community for as long as it was first introduced. Any other tool that we could remember that has more utility than anything is the Mimikatz. It has been years since the release of both of these tools but their ability to consistently attack the Windows Machine is unmatched. We know that neither PowerShell Empire nor Mimikatz is being used in the wild currently because of their signature that has been added to almost all of the Anti-Virus Software and across Virus Total. This has although made them less useful as compared to Cobalt Strike and other alternatives but when it comes to understanding the basics of Windows Authentication Systems such as SAM and LSASS and attack them and extract credentials there is no tool that can work as efficiently as Mimikatz.

We covered various forms of Credential Dumping with Mimikatz in our Series but we didn't present a consolidated guide to use Mimikatz with PowerShell Empire. Hence, we created this resource.

DC Sync Attack

The Mimikatz DCSYNC-function allows an attacker to replicate Domain Controller (DC) behaviour. Typically impersonates as a domain controller and request other DC's for user credential data via GetNCChanges. But compromised account should be a member of administrators, Domain Admin or Enterprise Admin to retrieve account password hashes from the others domain controller. As a result, the intruder will build Kerberos forged tickets using a retrieved hash to obtain any of the Active Directory's resources. We have compromised the machine and its user who is a member of the privilege account (Administrators, Domain Admin or Enterprise Admin).

```
usemodule credentials/mimikatz/dcsync
set user krbtgt
execute
```

Loading the dcsync module will invoke the mimikatz PowerShell script to execute the dcsync attack to obtain the credential by asking from an other domain controller in the domain. Here, we are requesting KRBGTG account Hashes and as result, it will retrieve the KRBGTG NTLM HASH.

Similar to the DC Sync attack we just performed on a particular user so the NTLM hash returned is also of that particular user. But in case the attacker wants to extract the hash of the entirety of all the users created on the Domain Controller. This is when the hashdump module comes into action. It will perform the DC Sync attack for each and every user and then provide the hashes for all of them in a consolidated view as shown in the image below.

```
usemodule credentials/mimikatz/dcsync_hashdump
execute
Golden Ticket
```

Golden Ticket attack is a famous technique of impersonating users on an AD domain by abusing Kerberos authentication.

Default local accounts are built-in accounts that are created automatically when a Windows Server

domain controller is installed, and the domain is created. These default local accounts have counterparts in Active Directory. The default local accounts in the Users container include: Administrator, Guest, and KRBTGT. In the Active Directory domain, every domain controller runs a KDC (Kerberos Distribution Center) service that processes all requests for tickets to Kerberos. For Kerberos tickets, AD uses the KRBTGT account in the AD domain. A legitimate user begins the communication for a service request to the Application Server. The KRBTGT account acts as a service account for the Key Distribution Center (KDC) and separated into three parts: Database (db), Authentication Server (AS) and Ticket Granting Server (TGS). The Authentication Server (AS) verifies client authentication. If the logged user is authenticated successfully the AS issues a ticket called TGT which confirms to other servers that the user has been authenticated. Then the User request TGS from the KDC that will be used to access the service of the application server.

Forging Kerberos tickets depends on the password hash available to the attacker. Golden Tickets requires the KRBTGT password hash. Golden Tickets are forged Ticket-Granting Tickets (TGTs), also called authentication tickets, Attacker escapes authentication and initializes communication with KDC. Since a Golden Ticket is a forged TGT, it is sent to the Domain Controller as part of the TGS-REQ to get a service ticket. The TGT is used mainly to inform KDC's domain controller that another domain controller has authenticated the users. The reality is that the TGT has the hash KRBTGT password encrypted and any KDC service inside the domain may decrypt to prove it is valid.

If an intruder has access to an Active Directory forest/domain administrator/local administrator account, he/she can exploit Kerberos tickets for identity theft. A golden ticket attack is something that he/she creates a ticket created by Kerberos that is valid for 10 years. However, if any other user has changed its password, the attacker may use the KRBTGT account to stay on the network. The attacker may also create accessible user/computer/service tickets from Kerberos for a non-existent Active Directory account. As we know, there is some basic requirement to create a forged TGT i.e., extract the "domain Name, SID, krbtgt Hash", Once an attacker has admin access to a Domain Controller, the KRBTGT account password hashes can be extracted using Mimikatz.

Once we have compromised the victim machine who is a member of AD, then we can use the following module directly without an admin privilege session.

```
usemodule credentials/mimikatz/golden_ticket
set domain <Domain_name>
set sid <SID>
set group 500
set user pavan
set krbtgt_hash <ntlm_hash>
set id 500
execute
kerberos::golden /user:pavan /domain:ignite.local /sid: SID
back
shell dir \\DC1.ignite.local\\c$
```

Extracting Tickets

We saw how to forge tickets. Tickets last longer than a normal persistence. Golden Tickets can last up to 10 years. Hence, we should have the ability to extract those tickets for usage down the road.

```
usemodule credentials/mimikatz/extract_tickets
execute
Domain Cache
Microsoft Windows stores previous users' logon information locally so that they can log on if a logon
```

server is unreachable during later logon attempts. This is known as Domain Cache credential (DCC) but in-actually it is also known as MSCACHE or MSCASH hash. It sorted the hash of the user's password that you can't perform pass-the-hash attacks with this type of hash. It uses the MSCACHE algorithm for generating password hash and that are stored locally in the Windows registry of the Windows operating system. These hashes are stored in the Windows registry, by default the last 10 hashes.

There two versions of MSCASH/MSCACHE or DCC [warn]

credit: hackingarticles.in

evilginx - January 11, 2025, 4:08 am

Isn't this hectic to keep up with cmds and consoles when you already have starkiller? my opinion tho

Sociopath - January 14, 2025, 11:10 pm

(Jan 11, 2025 05:08 AM)evilginx Wrote: [\[CITE\]](#) Isn't this hectic to keep up with cmds and consoles when you already have starkiller? my opinion tho

Yea I share the same opinion here. Starkiller would be way more efficient.