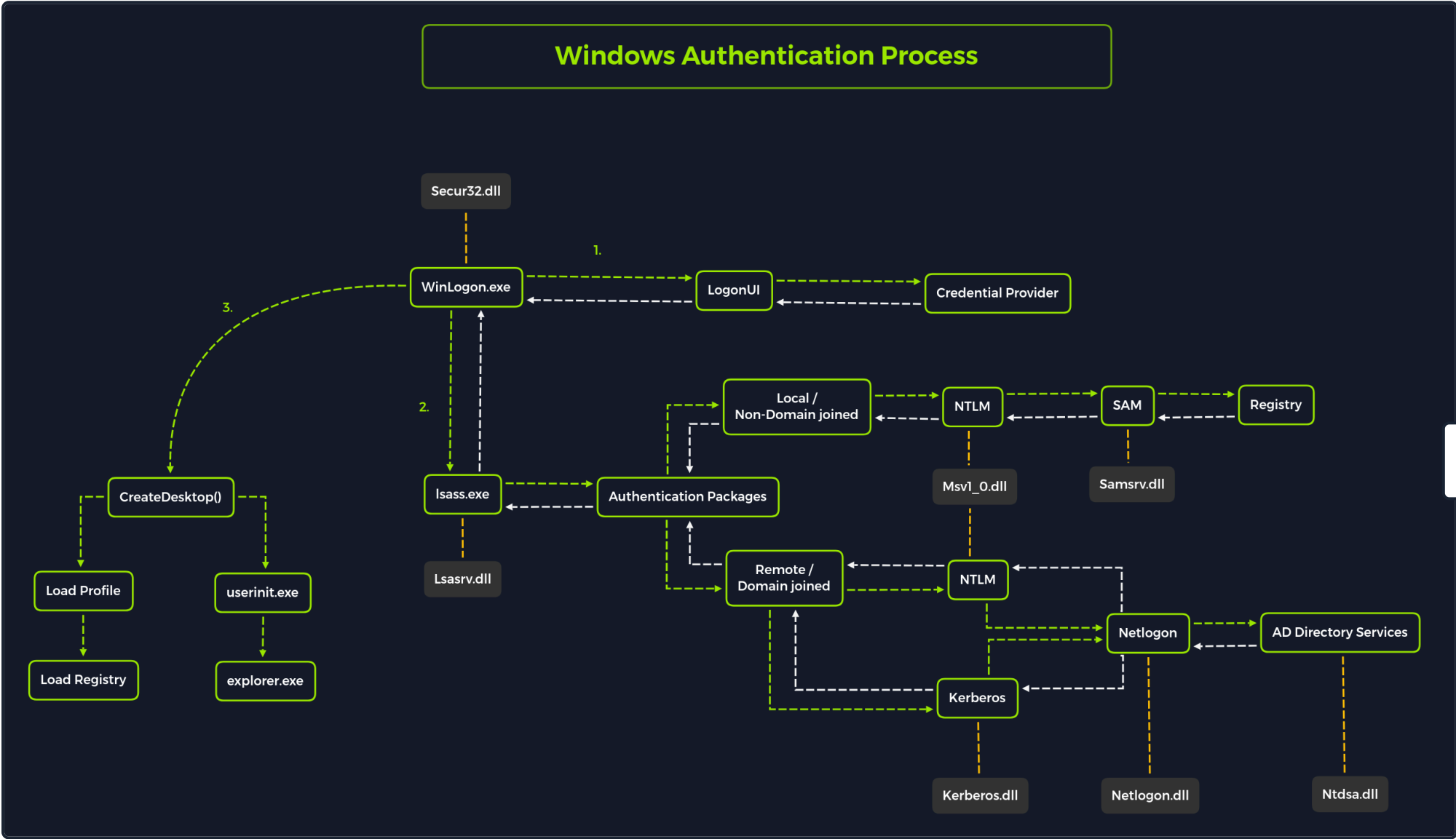


# Windows Authentication Process

The [Windows client authentication process](#) involves multiple modules responsible for logon, credential retrieval, and verification. Among the various authentication mechanisms in Windows, Kerberos is one of the most widely used and complex. The [Local Security Authority \(LSA\)](#) is a protected subsystem that authenticates users, manages local logins, oversees all aspects of local security, and provides services for translating between user names and security identifiers (SIDs).

The security subsystem maintains security policies and user accounts on a computer system. On a Domain Controller, these policies and accounts apply to the entire domain and are stored in Active Directory. Additionally, the LSA subsystem provides services for access control, permission checks, and the generation of security audit messages.

## Windows authentication process diagram



Local interactive logon is handled through the coordination of several components: the logon process ([WinLogon](#)), the logon user interface process ([LogonUI](#)), credential providers, the Local Security Authority Subsystem Service ([LSASS](#)), one or more authentication packages, and either the Security Accounts Manager ([SAM](#)) or Active Directory. Authentication packages, in this context, are Dynamic-Link Libraries (DLLs) responsible for performing authentication checks. For example, for non-domain-joined and interactive logins, the [Msv1\\_0.dll](#) authentication package is typically used.

[WinLogon](#) is a trusted system process responsible for managing security-related user interactions, such as:

- Launching [LogonUI](#) to prompt for credentials at login
- Handling password changes
- Locking and unlocking the workstation

To obtain a user's account name and password, WinLogon relies on credential providers installed on the system. These credential providers are [COM](#) objects implemented as DLLs.

WinLogon is the only process that intercepts login requests from the keyboard, which are sent via RPC messages from `Win32k.sys`. At logon, it immediately launches the `LogonUI` application to present the graphical user interface. Once the user's credentials are collected by the credential provider, WinLogon passes them to the Local Security Authority Subsystem Service (`LSASS`) to authenticate the user.

## LSASS

The `Local Security Authority Subsystem Service` (`LSASS`) is comprised of multiple modules and governs all authentication processes. Located at `%SystemRoot%\System32\lsass.exe` in the file system, it is responsible for enforcing the local security policy, authenticating users, and forwarding security audit logs to the `Event Log`. In essence, LSASS serves as the gatekeeper in Windows-based operating systems. A more detailed illustration of the LSASS architecture can be found [here](#).

Authentication Packages	Description
<code>Lsasrv.dll</code>	The LSA Server service both enforces security policies and acts as the security package manager for the LSA. The LSA contains the Negotiate function, which selects either the NTLM or Kerberos protocol after determining which protocol is to be successful.
<code>Msv1_0.dll</code>	Authentication package for local machine logons that don't require custom authentication.
<code>Samsrv.dll</code>	The Security Accounts Manager (SAM) stores local security accounts, enforces locally stored policies, and supports APIs.
<code>Kerberos.dll</code>	Security package loaded by the LSA for Kerberos-based authentication on a machine.
<code>Netlogon.dll</code>	Network-based logon service.
<code>Ntdsa.dll</code>	This library is used to create new records and folders in the Windows registry.

Source: [Microsoft Docs](#).

Each interactive logon session creates a separate instance of the WinLogon service. The `Graphical Identification and Authentication` (`GINA`) architecture is loaded into the process area used by WinLogon, receives and processes the credentials, and invokes the authentication interfaces via the `LSALogonUser` function.



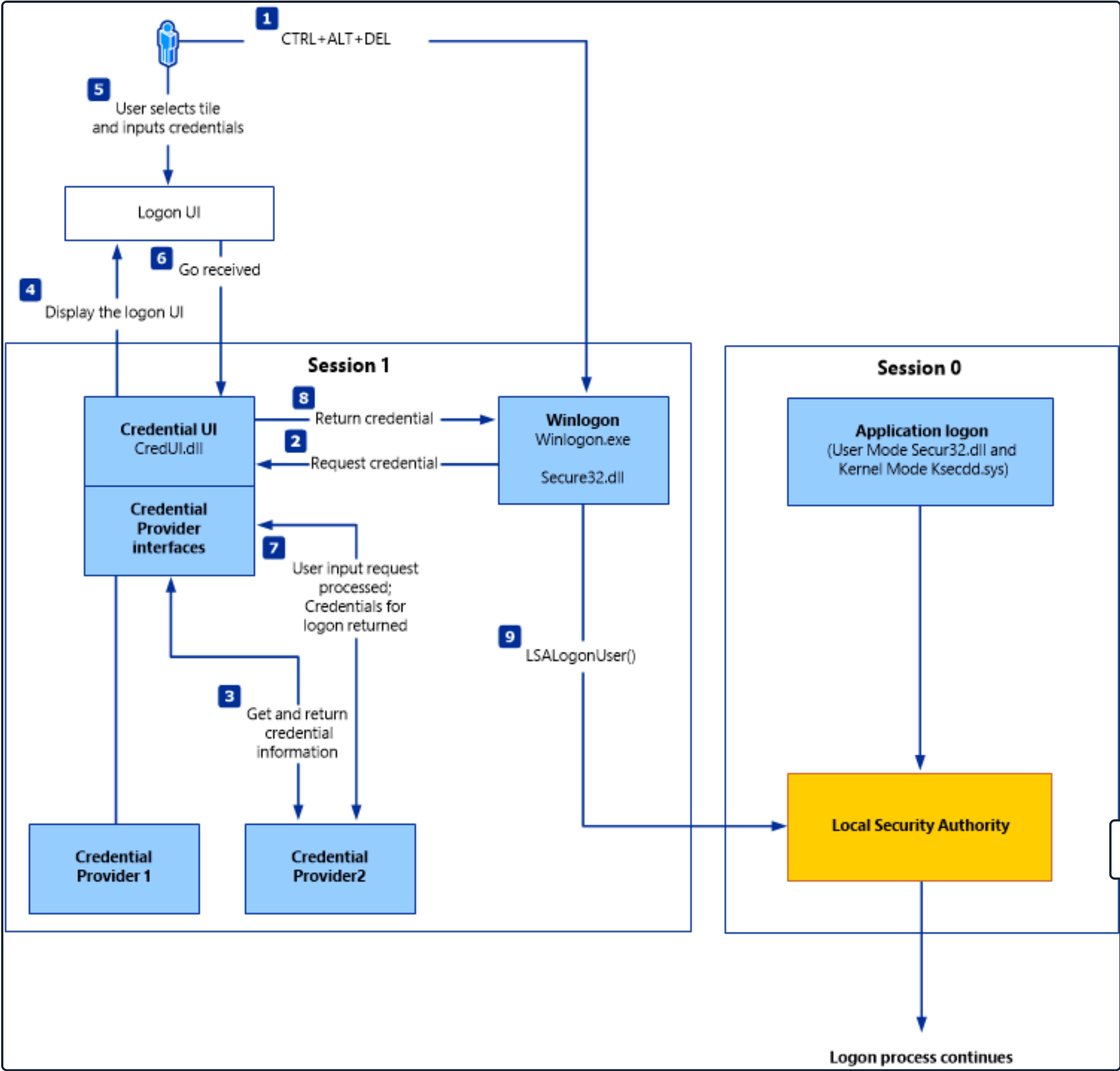
## SAM database

The `Security Account Manager` (`SAM`) is a database file in Windows operating systems that stores user account credentials. It is used to authenticate both local and remote users and uses cryptographic protections to prevent unauthorized access. User passwords are stored as hashes in the registry, typically in the form of either `LM` or `NTLM` hashes. The SAM file is located at `%SystemRoot%\system32\config\SAM` and is mounted under `HKLM\SAM`. Viewing or accessing this file requires `SYSTEM` level privileges.

Windows systems can be assigned to either a workgroup or domain during setup. If the system has been assigned to a workgroup, it handles the SAM database locally and stores all existing users locally in this database. However, if the system has been joined to a domain, the Domain Controller (`DC`) must validate the credentials from the Active Directory database (`ntds.dit`), which is stored in `%SystemRoot%\ntds.dit`.

To improve protection against offline cracking of the SAM database, Microsoft introduced a feature in Windows NT 4.0 called `SYSKEY` (`syskey.exe`). When enabled, SYSKEY partially encrypts the SAM file on disk, ensuring that password hashes for all local accounts are encrypted with a system-generated key.

## Credential Manager



Source: [Microsoft Docs](#).

Credential Manager is a built-in feature of all Windows operating systems that allows users to store and manage credentials used to access network resources, websites, and applications. These saved credentials are stored per user profile in the user's **Credential Locker**. The credentials are encrypted and stored at the following location:

Windows Authentication Process

```
PS C:\Users\[Username]\AppData\Local\Microsoft\[Vault/Credentials]\
```

There are various methods to decrypt credentials saved using Credential Manager. We will practice hands-on with some of these methods in this module.

NTDS

It is very common to encounter network environments where Windows systems are joined to a Windows domain. This setup simplifies centralized management, allowing administrators to efficiently oversee all systems within their organization. In such environments, logon requests are sent to Domain Controllers within the same Active Directory forest. Each Domain Controller hosts a file called **NTDS.dit**, which

is synchronized across all Domain Controllers, with the exception of [Read-Only Domain Controllers \(RODCs\)](#).

**NTDS.dit** is a database file that stores Active Directory data, including but not limited to:

- User accounts (username & password hash)
- Group accounts
- Computer accounts
- Group policy objects

Later in this module, we will explore methods for extracting credentials from the **NTDS.dit** file.

Now that we have gone through a primer on credential storage concepts, let's study the various attacks we can perform to extract credentials and further our access during assessments.

← Previous

Next →

+10 Streak pts

✔ Mark Complete & Next


 Cheat Sheet

Table of Contents

Introduction

Introduction

✔

Password Cracking Techniques

 Introduction to Password Cracking

✔

 Introduction to John The Ripper

✔

 Introduction to Hashcat

✔

 Writing Custom Wordlists and Rules

✔

 Cracking Protected Files

✔

 Cracking Protected Archives

✔

Remote Password Attacks

 Network Services

✔

 Spraying, Stuffing, and Defaults

✔

Extracting Passwords from Windows Systems

[Windows Authentication Process](#)

 Attacking SAM, SYSTEM, and SECURITY

 Attacking LSASS

 Attacking Windows Credential Manager

 Attacking Active Directory and NTDS.dit



 Credential Hunting in Windows

Extracting Passwords from Linux Systems





 Linux Authentication Process

 Credential Hunting in Linux

Extracting Passwords from the Network

-  Credential Hunting in Network Traffic
-  Credential Hunting in Network Shares

Windows Lateral Movement Techniques

-  Pass the Hash (PtH)
-  Pass the Ticket (PtT) from Windows
-  Pass the Ticket (PtT) from Linux
-  Pass the Certificate

Password Management


- Password Policies
- Password Managers


Skills Assessment


-  Skills Assessment - Password Attacks

My Workstation



 Interact

 Terminate

 Reset

Life Left: 74m

