Capturing Windows Passwords using the Network Provider API

A step-by-step guide to building your own password capture DLL

Sergey Polak
Senior Programmer
Fish & Neave
Spolak@fishneave.com
Agenda

- Introduction
  - Password capturing in general
  - What is Network Provider API?
- Details about the Network Provider API functions
- Writing your own Network Provider
  - Minimum requirements
  - Differences between the versions of Windows (NPLogon vs. NPLogonNotify)
  - Installing your provider DLL
- Demonstration
- Conclusion
  - Q&A
Introduction

- Password capturing in general
  - Capture Methods
    - Keystroke logging software
    - Keystroke logging hardware
    - Social Engineering
    - Camera
  - Are there legitimate reasons for capturing user passwords on a corporate network?
    - Administrative (or user) convenience
    - Password complexity enforcement
    - Surveillance
Introduction

- The Windows Network Provider API – what is it and what is it good for?
  - Logon network authentication
  - Network connections, device redirection and enumeration
Digging Deeper into the Network Provider API

- Capabilities
  - User Credential Management
  - Network Connections (Device Redirection)
  - Searching and Enumeration
  - “Administrative” Functions
Digging Deeper into the Network Provider API

- Credential Management
  - Logon Notifications
    - NPLLogon and NPLLogonNotify
  - Password Change Notifications
    - NPPasswordChangeNotify
  - Current User Query
    - NPGetUser
Digging Deeper into the Network Provider API

- Network Connections (Device Redirection)
  - NPAAddConnection and NPAAddConnection3
  - NPGetConnection
  - NPCancelConnection
  - NPGetConnectionPerformance
Digging Deeper into the Network Provider API

- Searching and Enumeration
  - NPOpenEnum, NPEnumResource and NPCloseEnum
  - NPSearchDialog
Digging Deeper into the Network Provider API

- “Administrative” Functions
  - NPGetDirectoryType
  - NPDirectoryNotify
Writing Your Own Network Provider

- Minimum Requirements

```c
DWORD NPGetCaps(DWORD nIndex);
```

`nIndex` values and responses:
- `WNNC_SPEC_VERSION`: `WNNC_SPEC_VERSION51`
- `WNNC_NET_TYPE`: value from predefined list
- `WNNC_USER`
- `WNNC_CONNECTION`
- `WNNC_DIALOG`
- `WNNC_ADMIN`
- `WNNC_ENUMERATION`
- `WNNC_START`: time until started
- `WNNC_AUTHENTICATION`: `WNNC_AUTH_LOGON`

Export function by name and as ordinal 13
Writing Your Own Network Provider

- Differences between the versions of Windows
  - Windows 95, 98 and ME
    - NPLLogon (exported as ordinal 43)
  - Windows NT, 2000, XP and 2003 Server
    - DWORD APIENTRY NPLLogonNotify( PLUID lpLogon, LPCWSTR lpAuthentInfoType, LPVOID lpAuthentInfo, LPCWSTR lpPreviousAuthentInfoType, LPVOID lpPreviousAuthentInfo, LPWSTR lpStationName, LPVOID StationHandle, LPWSTR* lpLogonScript );

- Password change notification
  - DWORD APIENTRY NPPasswordChangeNotify( LPCWSTR lpAuthentInfoType, LPVOID lpAuthentInfo, LPCWSTR lpPreviousAuthentInfoType, LPVOID lpPreviousAuthentInfo, LPWSTR lpStationName, LPVOID StationHandle, DWORD dwChangeInfo );
NPLLogonNotify Parameters

- **lpLogon**
  - Pointer to the session ID

- **lpAuthentInfoType**
  - A string that identifies the type of login. The values are:
    - MSV1_0:Interactive
    - Kerberos:Interactive

- **lpAuthentInfo**
  - Depending on the value of lpAuthentInfoType, this is either a MSV1_0_INTERACTIVE_LOGON or a KERB_INTERACTIVE_LOGON structure
NPLogonNotify Parameters – lpAuthentInfo details

- MSV1_0_INTERACTIVE_LOGON or KERB_INTERACTIVE_LOGON structure members:
  - MessageType
  - LogonDomainName
    - Type UNICODE_STRING
  - UserName
    - Type UNICODE_STRING
  - Password
    - Type UNICODE_STRING
NPLLogonNotify Parameters

- **lpPreviousAuthentInfoType**
  - NULL unless user was forced to change authentication information (such as password)

- **lpPreviousAuthentInfo**

- **lpStationName**
  - WinSta_0 or WinSta0
    - Interactive user logon
  - SvcCtl
    - Service logon

- **StationHandle**
  - If interactive login, this is a handle to dialog box currently on screen

- **IpLogonScript**
  - Return a pointer to a logon script to execute
Installing Your Provider DLL

- `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\NetworkProvider\Order`
  - ProviderOrder
- `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`
  - Key with name of provider
    - Group = “NetworkProvider”
  - NetworkProvider subkey
    - Class = 2 (WN_CREDENTIAL_CLASS)
    - ProviderPath
    - Name
    - Description (optional)
    - NetNotLoading (optional)
    - CallOrder (optional)
    - NetID (optional)
Demonstration
References

- Microsoft Developer Network Library (MSDN)
  - Network Provider API is under Security\SDK Documentation\Authentication
Q&A?