

Forum:[Hack Forums](#)

Thread: [opsec for begginers book](#)

itzRealWubaDubaLubb - December 1, 2024, 3:21 pm

i found this book might help beginner to stay anonymous

<https://filetransfer.io/data-package/UcD8lCN4#link>

Charlie Sheen - December 1, 2024, 3:47 pm

[Bastille](#) can you check this out

Tsar - December 1, 2024, 6:35 pm

(Dec 1, 2024 04:47 PM)Charlie Sheen Wrote: [\[CITE\]](#) [Bastille](#) can you check this out

It is safe pdf. Not as in depth as I would hope lol but i guess it is for beginners.

File is clean though

Bastille - December 2, 2024, 2:13 am

(Dec 1, 2024 04:47 PM)Charlie Sheen Wrote: [\[CITE\]](#) [Bastille](#) can you check this out

(Dec 1, 2024 07:35 PM)Rapid Wrote: [\[CITE\]](#) It is safe pdf. Not as in depth as I would hope lol but i guess it is for beginners.

File is clean though

Cannot make an analysis in depth, it looks like the file is obfuscated, but there is a sign of an embedded JavaScript (/JS)

I started with pdfid.py to get a preview :

The /Encrypt field indicates that the PDF is encrypted. This can be used to protect sensitive data, but in a malicious context, it could also hide malicious content or embedded JavaScript code.

The /JS field shows that there is an object containing JavaScript. Even if the /JavaScript field is set to 0, the mere presence of /JS is a red flag, as JavaScript scripts in PDFs are often used to execute malicious actions (e.g., downloads or exploits).

This has been detected by VirusTotal too :

<https://www.virustotal.com/gui/file/6734e3f485e91278bfd505559d50f88218b597f69f6e04825da870ad0b48a5f9/detection>

Next step, I run pdf-parser.py to see through the PDF file and how it's composed. Here is a preview of the content :

We can see that some objects don't have important information, this could be legit, or non-legit in case the file is malicious. It has been flagged here in the TRIA.GE report that the PDF is malformed (due to the obfuscated content) : <https://tria.ge/241202-kgfgratjbq>

If we go back to our /Encrypt field we can find a correspondence for Obj 102 which contains the settings of the encryption :

Code obj 102 0

Type:

Referencing:

```
<<
  /Filter /Standard
  /V 1
  /R 2
  /O '(Vó\x00\x8b}0\x9c\x00¹#\x95³+P\x9a\x8a\x97=â3ï9ùSdG\\z\x17Í\x82)'
  /U '(\x89\x02:\x80"\x1cì\xadâzšïQô\x8cÂï!Yµ\\tí\x1c²\x9dê^\x80B{Áª)'
  /P -60
>>
```

['/Standard']

AI Quote:

Analysis of the fields

/Filter /Standard:

Indicates that the security filter used is the PDF standard filter. This typically corresponds to password-based or key-based encryption.

/V 1:

The encryption version is 1. This corresponds to 40-bit RC4 encryption (weak security). It is an outdated and vulnerable method.

/R 2:

Revision 2 of the standard encryption. This means that a user password (/U) and an owner password (/O) are used to control access to the file.

/O (owner key) and /U (user key):

These fields contain encrypted keys. They are used to verify permissions and unlock the document. Encoded sequences like (Vó\x00\x8b... and (\x89\x02:...) represent the keys but cannot be used directly without the password.

/P -60:

Permissions associated with the document. This negative value indicates that restrictions are applied. For example:

- Printing or copying content is disabled.
- Modifications are limited.
- Behavior depends on the PDF reader and its enforcement of these restrictions.

What this means

The file is encrypted with weak encryption (40-bit RC4) using the standard filter. This type of encryption is often used to protect legitimate documents but can also be employed to hide malicious content.

If the file contains malicious scripts, this encryption protects them to prevent automated analysis tools from examining the content.

We can see in Adobe Reader that the permissions and the version of the encryption match :

In order to get more readable data, and as the encryption method is weak, we can use the following command from qpdf to decrypt some content of the file :

Code `qpdf --decrypt DiabloOpsecV1.pdf Decrypted.pdf`

From the encrypted PDF file we have an encrypted URI :

And from the decrypted file we have a decrypted URI :

The objective is probably to avoid tools scanning the URL and getting a match from databases with malicious links. But the website seems under construction :

Domain name :

Expires On : 2024-12-08

Registered On : 2023-12-08

Cert issued On : 2024-12-02

Registrar : NAMECHEAP INC

I've ran the PDF file in different online virtual environments and nothing suspicious. Even malicious PDF files can detect virtual environments, but don't know if it's advanced enough to do this in this case.

any.run PDF file : <https://app.any.run/tasks/bc2bfe92-8633-4c1d-b6a3-c49cc895a1a6>

any.run with a clean PDF file : <https://app.any.run/tasks/716a6182-d152-471d-b79a-65f5ad553d0d>

The DNS requests and connections seems ok, the website seems to be hosted on hostinger, but nothing in the logs of any.run.

Maybe the /JS was flagged because of the URI, the PDF file has been made with FPDF, which is not conventional tho :

Code `/CreationDate (D:20241109212843)`

`/Producer (FPDF 1.7)`

This was obfuscated but I was able to get this info after decrypting the pdf file.

If someone wants to take a deeper look or bring some info feel free, this /JS triggers me but can't say for sure if it's malicious or not.

Besides, 4MB for a 5 pages PDF file seems huge, some OBJ such as 99 have huge length compared to the others :

itzRealWubaDubaLubb - December 3, 2024, 11:56 am

(Dec 2, 2024 03:13 AM) Bastille Wrote: [\[CITE\]](#) Cannot make an analysis in depth, it looks like the file is obfuscated, but there is a sign of an embedded JavaScript (/JS)

I started with pdfid.py to get a preview :

The /Encrypt field indicates that the PDF is encrypted. This can be used to protect sensitive data, but in a malicious context, it could also hide malicious content or embedded JavaScript code.

The /JS field shows that there is an object containing JavaScript. Even if the /JavaScript field is set to 0, the mere presence of /JS is a red flag, as JavaScript scripts in PDFs are often used to execute malicious actions (e.g., downloads or exploits).

This has been detected by VirusTotal too :

<https://www.virustotal.com/gui/file/6734e3f485e91278bfd505559d50f88218b597f69f6e04825da870ad0b48a5f9/detection>

Next step, I run pdf-parser.py to see through the PDF file and how it's composed. Here is a preview of the content :

We can see that some objects don't have important information, this could be legit, or non-legit in case the file is malicious. It has been flagged here in the TRIA.GE report that the PDF is malformed (due to the obfuscated content) : <https://tria.ge/241202-kgfqratjbq>

If we go back to our /Encrypt field we can find a correspondence for Obj 102 which contains the settings of the encryption :

Code obj 102 0

Type:

Referencing:

```
<<
  /Filter /Standard
  /V 1
  /R 2
  /O '(Vó\x00\x8b}0\x9c\x00¹f#\x95³+P\x9a\x8a\x97=â3İ9ùSdG\\z\x17Í\x82)'
  /U '(\x89\x02:\x80"\x1cì\xadâzšİQô\x8cÂî!Yµ\|tÍ\x1c²\x9dê^\x80B{Áª)'
  /P -60
>>
```

[/Standard']

AI Quote:

Analysis of the fields

/Filter /Standard:

Indicates that the security filter used is the PDF standard filter. This typically corresponds to password-based or key-based encryption.

/V 1:

The encryption version is 1. This corresponds to 40-bit RC4 encryption (weak security). It is an

outdated and vulnerable method.

/R 2:

Revision 2 of the standard encryption. This means that a user password (/U) and an owner password (/O) are used to control access to the file.

/O (owner key) and /U (user key):

These fields contain encrypted keys. They are used to verify permissions and unlock the document. Encoded sequences like (Vó\x00\x8b... and (\x89\x02:...) represent the keys but cannot be used directly without the password.

/P -60:

Permissions associated with the document. This negative value indicates that restrictions are applied. For example:

- Printing or copying content is disabled.
- Modifications are limited.
- Behavior depends on the PDF reader and its enforcement of these restrictions.

What this means

The file is encrypted with weak encryption (40-bit RC4) using the standard filter. This type of encryption is often used to protect legitimate documents but can also be employed to hide malicious content.

If the file contains malicious scripts, this encryption protects them to prevent automated analysis tools from examining the content.

We can see in Adobe Reader that the permissions and the version of the encryption match :

In order to get more readable data, and as the encryption method is weak, we can use the following command from qpdf to decrypt some content of the file :

Code `qpdf --decrypt DiabloOpsecV1.pdf Decrypted.pdf`

From the encrypted PDF file we have an encrypted URI :

And from the decrypted file we have a decrypted URI :

The objective is probably to avoid tools scanning the URL and getting a match from databases with malicious links. But the website seems under construction :

Domain name :

Expires On : 2024-12-08

Registered On : 2023-12-08

Cert issued On : 2024-12-02

Registrar : NAMECHEAP INC

I've ran the PDF file in different online virtual environments and nothing suspicious. Even malicious PDF files can detect virtual environments, but don't know if it's advanced enough to do this in this case.

any.run PDF file : <https://app.any.run/tasks/bc2bfe92-8633-4c1d-b6a3-c49cc895a1a6>
any.run with a clean PDF file : <https://app.any.run/tasks/716a6182-d152-471d-b79a-65f5ad553d0d>

The DNS requests and connections seems ok, the website seems to be hosted on hostinger, but nothing in the logs of any.run.

Maybe the /JS was flagged because of the URI, the PDF file has been made with FPDF, which is not conventional tho :

Code /CreationDate (D:20241109212843)

/Producer (FPDF 1.7)

This was obfuscated but I was able to get this info after decrypting the pdf file.

If someone wants to take a deeper look or bring some info feel free, this /JS triggers me but can't say for sure if it's malicious or not.

Besides, 4MB for a 5 pages PDF file seems huge, some OBJ such as 99 have huge length compared to the others :

i think bcz the owner used a software to protect the pdf from copy and edit . i have it in my computer i dont think is malicious

Bastille - December 3, 2024, 10:40 pm

(Dec 3, 2024 12:56 PM)itZRealWubaDubaLubb Wrote: [\[CITE\]](#) i think bcz the owner used a software to protect the pdf from copy and edit . i have it in my computer i dont think is malicious

Possibly, but this was an interesting threat hunt, first time I'm doing this on a PDF file, I'll monitor the website to see what's coming

ds47 - December 6, 2024, 7:44 am

Going to check this out thanks. Still looking for a definitive OPSEC guide

Xiu97yd0 - January 18, 2025, 4:32 pm

Thank you i appreciate it