

Forum:[Hack Forums](#)

Thread: [Bug Report PDF Generator.py](#)

source - May 10, 2023, 8:26 pm

Decided to see the capabilities of PDF generating in python.

These fields can be changed of course.

Features:Background color customization

Application name, vulnerability description, steps to reproduce, impact, exploitability, affected scope, severity, remediation, reward, and date input

Styling customization for fonts, font sizes, font colors, and document layout

Report is generated in a PDF file format that can be easily shared with clients or project managers.

I personally wouldn't use this for reports but rather personal documentation to save and go back to.

```
Code from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter
from reportlab.lib.units import inch
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.enums import TA_CENTER
from reportlab.lib.colors import HexColor
```

```
# create a new PDF with Reportlab
def create_pdf(filename, data):
    # set up the document
    pdf = canvas.Canvas(filename, pagesize=letter)

    # set background color
    pdf.setFillColorRGB(25/255, 25/255, 25/255)
    pdf.rect(0, 0, letter[0], letter[1], fill=True)

    # set the font and font size
    styles = getSampleStyleSheet()
    pdf.setFont(styles['Normal'].fontName, 10)

    # set font color
    font_color = HexColor('#2E8B57')
    pdf.setFillColor(font_color)

    width, height = letter
```

```
# write the title
pdf.drawCentredString(width/2.0, height-0.5*inch, "Bug Bounty Report")

# write the application name
pdf.drawCentredString(width/2.0, height-1.0*inch, data['app_name'])

# write the report
pdf.drawString(0.5*inch, height-1.5*inch, "## Vulnerability Description\n")
pdf.drawString(0.5*inch, height-1.75*inch, data['vulnerability_description'])

# write the steps to reproduce
pdf.drawString(0.5*inch, height-2.25*inch, "## Steps to Reproduce\n")
pdf.drawString(0.5*inch, height-2.5*inch, data['steps_to_reproduce'])

# write the impact
pdf.drawString(0.5*inch, height-3.0*inch, "## Impact\n")
pdf.drawString(0.5*inch, height-3.25*inch, data['impact'])

# write the exploitability
pdf.drawString(0.5*inch, height-3.75*inch, "## Exploitability\n")
pdf.drawString(0.5*inch, height-4.0*inch, data['exploitability'])

# write the affected scope
pdf.drawString(0.5*inch, height-4.5*inch, "## Affected Scope\n")
pdf.drawString(0.5*inch, height-4.75*inch, data['affected_scope'])

# write the severity
pdf.drawString(0.5*inch, height-5.25*inch, "## Severity\n")
pdf.drawString(0.5*inch, height-5.5*inch, data['severity'])

# write the remediation
pdf.drawString(0.5*inch, height-6.0*inch, "## Remediation\n")
pdf.drawString(0.5*inch, height-6.25*inch, data['remediation'])

# write the reward
text = f"## Reward: {data['reward']}"
pdf.drawString(0.5*inch, height-6.75*inch, text)

# write the date
text = f"## Date: {data['date']}"
pdf.drawString(0.5*inch, height-7.0*inch, text)

# save the PDF
pdf.showPage()
pdf.save()
print(f"PDF generated successfully as {filename}")
```

```
# ask for the filename
filename = input("Enter filename (without extension): ") + ".pdf"
```

```
# ask for the data
```

```
data = {
    'app_name': input("Enter application name: "),
    'vulnerability_description': input("Enter vulnerability description: "),
    'steps_to_reproduce': input("Enter steps to reproduce: "),
    'impact': input("Enter impact: "),
    'exploitability': input("Enter exploitability: "),
    'affected_scope': input("Enter affected scope: "),
    'severity': input("Enter severity: "),
    'remediation': input("Enter remediation: "),
    'reward': input("Enter reward: "),
    'date': input("Enter date (MM/DD/YYYY): ")
}
```

```
# create the PDF
create_pdf(filename, data)
screenshot:
```

Maxie - May 11, 2023, 12:11 am

Have you tried using puppeteer for generating pdfs? In Python there is a wrapper called pyppeteer.

I haven't used the python version, but we use a similar thing with nodejs for generating most text layout pdfs.

Fortunate - May 11, 2023, 10:48 am

Never experimented with pdf generation in python but this seems very useful.

How heavy is to run?

source - May 11, 2023, 4:18 pm

(May 11, 2023 10:48 AM)Fortunate Wrote: [\[CITE\]](#) Never experimented with pdf generation in python but this seems very useful.

How heavy is to run?

Not heavy at all, it's instant

(May 11, 2023 12:11 AM)Maxie Wrote: [\[CITE\]](#) Have you tried using puppeteer for generating pdfs? In Python there is a wrapper called pyppeteer.

I haven't used the python version, but we use a similar thing with nodejs for generating most text layout pdfs.

I attempted puppeteer but could not get it working, I'm a noob still.

puppateer is good for scraping.

Nightmare - July 15, 2023, 10:25 pm

BTW the script you made is really good for fast PDF generation,thank you for share!