
[REPORT TITLE]

[SUBTITLE]

student@youremailaddress.com, [Other info, e.g. OSID]

2023-07-13

Contents

1	Overview	1
1.1	Intent	1
2	Requirements	2
2.1	Important Note	2
2.2	Software	2
2.2.1	Obsidian plugin breakdown	2
2.2.2	Minimum LaTeX installation	3
3	Installation	4
3.1	Optional - version control	4
4	Configuration	5
4.1	templates/frontmatter.yml	5
5	Suggested Usage	6
5.1	Take notes	6
5.2	Organize the flow of the report	6
5.3	Generate the report	7
5.3.1	Usage	7
6	How do I ...	9
6.1	...use Markdown if I'm not very familiar with it?	9
6.2	...get my name and stuff on the report?	9
6.3	...do the automatic image labeling?	9
6.4	...automatically reference images in the report?	10
6.5	...do something not covered here?	10
7	Coming Soon!	11
7.1	Tutorial	11

1 Overview

This is a fork of [noraj/OSCP-Exam-Report-Template-Markdown](#).

The goal is to help streamline report writing during labs/pentesting, by reducing the number of “passes” needed to create the report.

The idea is that you can take your notes organically during the test, and with some minimal editing, quickly assemble your text into a clean, professional looking report with

- an automatically generated title page
- an automatically generated table of contents
- automatically generated image labels
- automatically generated references to figures within the document
- automatically generated chapter/heading/section/subsection numbers

This project makes use of the Obsidian Markdown Editor, The Eisvogel LaTeX template, pandoc, and some custom scripts to help save you time writing reports.

1.1 Intent

The intent for this repo is for it to be used as a “starter” for reporting on tests, labs, etc. This was designed for the Obsidian Markdown editor in order to take advantage of some of the native features as well as community plugins.

The `.obsidian` folder is included in this repo, so you should have all the plugins and settings enabled from the get-go. Please make whatever adjustments you need to work comfortably.

2 Requirements

2.1 Important Note

This is intended for use only on Linux. I have not tested this on Windows, and I am not planning to any time soon. If you'd like to be a guinea pig, feel free to fork and submit a pull request!

2.2 Software

- Obsidian Markdown Editor , with the following plugins:
 - Paste image rename plugin (included in this repo)
 - Obsidian Link Converter plugin (included in this repo)
 - Obsidian Heading Shifter plugin (included in this repo)
- pandoc
- The Eisvogel LaTeX template
- LaTeX/TexLive - see below
- awk - should come standard with Linux distro

2.2.1 Obsidian plugin breakdown

plugin name	what it does	why it's needed
Paste Image Rename	Renames images as they're pasted into the project.	better attachment organization; image filenames become captions in report

plugin name	what it does	why it's needed
Obsidian Link Converter	ensures links are CommonMark markdown spec with relative paths	Pandoc only recognizes CommonMark link formats. Also makes it trivially easy to convert older notes you may have with Wiki style links (! [[some link]])
Obsidian Heading Shifter	quickly promotes/demotes Markdown headers	Allows you to quickly promote and demote headings in an extracted section to match what they need to be in the final report

2.2.2 Minimum LaTeX installation

A full LaTeX installation is 1GB+, so if space is tight (or you just don't want bloat), this is the minimum set of packages you need to get off the ground.

```
apt install texlive-base texlive-binaries
texlive-fonts-extra texlive-fonts-extra-links
texlive-fonts-recommended texlive-latex-base
texlive-latex-extra texlive-latex-recommended
texlive-pictures texlive-plain-generic
```

3 Installation

1. Ensure you have all the required software and plugins listed above.
2. Clone this repo.
3. In Obsidian, open this repo as a vault.
4. Start taking notes!

3.1 Optional - version control

After cloning, remove the `.git` directory from the root of the repo. Then, to create a new repository, run the commands below:

```
# initializes a new git session in the current directory  
# with the default branch 'main'  
git init -b main
```

Now you have version history, and you can back up your personal fork of this repo to any of the popular Git hosting services, e.g. GitLab, GitHub, your own private git server, etc.

4 Configuration

This is a sampling of some of the configuration settings in this repo. This list is not comprehensive, and you can totally adjust anything you like to fit your needs.

- vim keybindings
- line numbers
- images/attachments are stored in `./img` by default
- suggested Heading Shifter hotkeys

4.1 `templates/frontmatter.yml`

This file contains the metadata that populates the Title, Subtitle, Date, etc. fields in the report. Simply update this file with your information and it will automatically appear in the report.

Alternatively, you can specify a custom frontmatter file with the `-y` flag.

5 Suggested Usage

5.1 Take notes

Take your notes in markdown, however you like.

5.2 Organize the flow of the report

There's tons of flexibility here, and no wrong way to do this. My general approach is to take notes in one big document as I'm testing, and then use Obsidian's "extract selection into new file" feature to "slice and dice" my notes into lots of smaller files, which will later be assembled into the final report.

Generally, you'll want to "extract" anything you think might appear in more than one place, whether that's one or more places within the same report, or in two or more documents – think "walkthrough" versus "report".

Future releases will include a tutorial with some working examples to give you a good idea of how to use this repo.

5.3 Generate the report

5.3.1 Usage

```
(kali㉿kali)-[~/reports]
$ ./generate.sh
./generate.sh [-vdp] [-f LIST [-f LIST ...]] [-o PDF] [-y FRONTMATTER] FILE1 [FILE2 FILE3 ...]

Process and pass FILE(s) to pandoc for rendering into a report with the
Eisvogel LaTeX template. FILEs are processed in the order in which they
appear in the list. If FILE is missing or - will read from STDIN.

-v          verbose; log debug info to STDERR

-d          update FRONTMATTER metadata file with today's date
           in YYYY-MM-DD (%F) format

-p          preview: renders the markdown that will
           be passed to pandoc

-f LIST     a file containing the list of files to process
           in the order in which they should appear in the
           report. multiple files are processed in the order
           in which they appear in the invocation. contents of
           files in LIST(s) will appear in the report before the
           contents of FILE(s)

-o PDF      specify output file name with extension, e.g. path/to/report.pdf
           default is report.pdf

-y FRONTMATTER specify a custom .yaml frontmatter metadata file
```

Figure 5.1: usage

The `generate.sh` script was written to allow you the most flexibility in assembling the report. In its most basic form, you can invoke it with the following command:

```
./generate.sh README.md
```

The `report.pdf` file included this repo was generated with the commands shown in the screenshot below. Note the `-v` flag which logs debug info to STDERR.

```
(kali㉿kali)-[~/reports]
$ ./generate.sh -v README.md
[+] debug enabled
[+] num list files: 0
[+] output file: /home/kali/reports/report.pdf
[+] using /home/kali/reports/templates/frontmatter.yml for metadata file
[+] remaining args: README.md
[+] added README.md to $FILE_LIST
[+] # input files: 1
[+] STANDALONE: -s
[+] generating report
[+] file /home/kali/reports/report.pdf created
```

Figure 5.2: generating report from README file

6 How do I ...

6.1 ...use Markdown if I'm not very familiar with it?

Obsidian ships with a read-only “Sandbox Vault” - this is a handy reference to have open as you write or edit your notes.

6.2 ...get my name and stuff on the report?

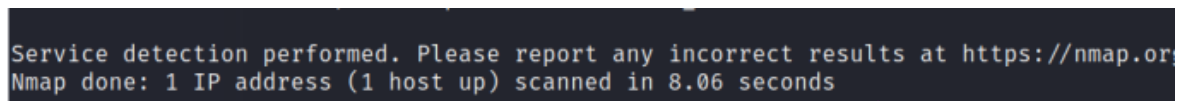
Provided in this repo is the file `templates/frontmatter.yml`. Simply update the relevant sections with your information, and it will appear in the report.

6.3 ...do the automatic image labeling?

When you paste an image into Obsidian, the Paste Image Rename plugin prompts you for a filename for the image. Whatever you put as the filename will become the label/caption for that image in the report.

As long as you choose descriptive names for the images you paste into your notes, the caption will automatically populate in the report.

For example, suppose you paste an image and rename it to `nmap-scan-leaking-domain-name.png`. When the report is rendered, you'll see this:



```
Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 8.06 seconds
```

Figure 1.1: nmap scan leaking domain name

file name (stripped of dashes and dots)

pasted image

Figure 6.1: rendered pasted image with filename

6.4 ...automatically reference images in the report?

As you're writing your report, you may want to refer to specific images/figures, but there's no way to be sure how it will be labeled (e.g. Figure 2.1) until the report is generated.

By adding `\label{someLabel}` to the alt-text part of the embed link – that is, the text between the square brackets – you can refer back to that figure in the text of the report with `ef{someLabel}`, and Pandoc will take care of the rest.

For example, suppose your markdown notes contain the following embed link:

```
![[some link \label{someUniqueLabel}](uri/to/resource.png)]
```

If you'd like to refer back to this figure later in your markdown, simply write

```
See Figure ef{someUniqueLabel}
```

The rendered report will look like this:

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 8.06 seconds
```

Figure 1.1: nmap scan leaking domain name

automatically generated matching figure numbers

An nmap scan of the 1000 most common TCP ports reveals a web server running on Port 80, with leaked domain name of `jupiter.htb`. See Figure 1.1

Figure 6.2: auto matching label numbers

6.5 ...do something not covered here?

I've included the FAQ from the original project. If something you want to do isn't covered here, submit an issue and we'll try and figure it out!

7 Coming Soon!

7.1 Tutorial

Future releases will include a tutorial for orientation and practice. Topics to be covered will include

- extracting note contents into new notes
- organizing the “flow” of the report
- converting from CherryTree format to Markdown