Luke Stephens (@lodestarluke)
Pentester | Hubby | Musician | On a mission to free my thoughts and actions from the limits which are imposed on them by society.
Feb 15 · 6 min read

# Luke's Ultimate OSCP Guide: Part 2— Workflow and documentation tips


Man walks through door with large shadow. OFFENSIVE security logo dramatically appears in a red abyss.

At the start of my labs, I wasn't really prepared for how much documentation I would be writing. I opted to document ALL of the exercises and 10 lab machines (while this is not compulsory, it earns you an extra 5 points in the exam). On top of that, I also documented the exam boxes. In total, this came to around 280 pages. Within the first couple of weeks, my reporting process changed dramatically, and I wasted a lot of time getting a good workflow happening.

Hopefully, you can make the most of your lab time by learning from my mistakes. Read on, fellow hackers!

. . .

## Environment

### The Virtual Machine

Offsec provides you with a modified Kali Linux virtual machine. I'm not going to run through setting it up here, but I will say this:

The virtual machine provided by Offsec is meant to be run with VMWare, not VirtualBox.

Generally, I'm more of a VirtualBox guy. It's free, and I like Vagrant. Unfortunately, converting a VMWare VM to VirtualBox, and vice versa, is dodgy. If you're using Linux or Windows as your host machine—you're in luck! Download VMWare workstation for free and install away. Unfortunately for Apple users—you will need to buy VMWare Fusion, or convert the Kali PWK VM to VirtualBox and use that. I used VirtualBox for the entire time including the exam. It worked pretty well but I ran into issues copy/pasting between the the host and guest machines. After being converted, the Kali PWK VM also lagged a lot more than the regular Kali in VirtualBox—perhaps that is just an isolated incident though.

## To update? Or not to update?

If you've spent much time trawling the #offsec channel on IRC, or any OSCP related chats, you will know that some of the most common problems arise from people updating their Kali VM. Be careful about what you update because it will cause problems with the exercises, running `apt upgrade` will break stuff. One exception to the rule is searchsploit which you can update by doing a `searchsploit -u`.

If you really want to check out the latest upgrades to Kali, I recommend having a seperate Kali VM which is fully updated, not the PWK Kali version.

## Sharing Files

Sharing files between your host machine and your Kali PWK VM is important for two reasons. Firstly, you'll find yourself wanting to share your documentation between the two regularly. Secondly, it makes backups easier.

Protip: Create a shared folder between your host and guest machine. Put the shared folder on the host machine inside a folder which syncs to the cloud (such as Dropbox or Google Drive). That way, your files are automatically being backed up to the cloud as you create them.

Another protip: Got some tools or files that you use regularly? Pop them in a private git repo. If you bork your VM, it won't take long to spin up a new one and pull down your custom tools.

## Applications that will make your life easier

- Shutter: For taking screenshots. When you take the screenshot, it's automatically added to the clipboard for an easy paste into your docs.
- CherryTree: CherryTree took over when KeepNote stopped being maintained. It's much better for a number of reasons. Ditch KeepNote and use CherryTree instead. CherryTree is also included in the latest Kali, while KeepNote is not.
- Terminator: Mainly for the lovely split-screen terminals.
- Burp: From memory, it's not covered in much detail in the PWK course. It's my favourite tool for hacking webapps and it comes pre-installed. There's plenty of tutorials out there for learning how to use it. Try YouTube.

# Documentation

## The Workflow

My main machine is a humble 13" Macbook Air from 2015. It's powerful enough for me, but it's no super-computer. Once my documentation reached about 80 pages of screenshot heavy material in "Pages" (the apple equivalent of Microsoft Word), my computer became so slow that it was virtually unresponsive. From then on my documentation process changed, I used CherryTree within Kali to create the initial documentation and screenshots without any formatting. When it came time to format it nicely, I would export the entire CherryTree file to HTML, share it with my host machine, copy the text over to Google docs, and add some nice formatting. Voila!

## CherryTree Setup

CherryTree allows for hierachical notetaking. I found it useful to set up the notes the same way as the network—with each subnet as a parent node, then machines as subnodes. If you want to go even more hierachical, you could create another layer under this for each stage of the hack. Perhaps "Enumeration", "Exploitation", "Privilege Escalation" and "Flags" would work well for you.

I also made a few top level nodes. One was a "cheatsheet" which kept a lot of command syntaxes that were hard to remember, one was "credentials" which stored credentials I had found throughout the network that I might want to try later, and one was called "flyover", which contained my full network enumeration scans, DNS

enumeration, a log of which boxes I had already owned, and which ones were yet to fall.

### What should be included in the documentation?

Basically, we can split the documentation in to three parts:

- The lab exercises (Not compulsory, but will earn an extra 5 points in the exam if you submit these alongside a write-up of 10 lab machines)
- 10 lab machines (Not compulsory, but will earn an extra 5 points in the exam if you submit these alongside the lab exercises write-up)
- The exam machines (Compulsory!)

When you're documenting the lab exercises, you need to show just enough to document the steps you took to achieve the task. No more, no less. If in doubt, leave it in—because it's better to have too much documentation than miss an important step!

When documenting machines (both in the labs AND in the exam), you need to include every step taken to exploit the machine including screenshots if necessary. There is no need to include things you tried that didn't work. You must also include one single screenshot which displays the output of the `proof.txt` file, the output of `ifconfig/ipconfig`, and the output of `whoami`, `id`, or a similar command to demonstrate your current user.

### Official Support

If you don't trust me, or if you want more info—the official support for the course documentation requirements can be found here. You can also contact the admins to clarify anything you're not 100% on.

You can also download a sample lab report here, although I didn't use it in the end, it didn't really fit with my reporting style.

## RTFM

The most important part of all this is to read the official documentation on how to report, the correct formats, etc. If the report is amazing but you send it in the wrong format, you will automatically receive a fail. Read it and read it again!

If you read these two pages 10 times over, you should be safe:

- Exam requirements (also contains reporting requirements): https://support.offensive-security.com/#!oscp-exam-guide.md
- PWK support page: https://support.offensive-security.com/#!pwk-support.md

## Where are the other parts to this guide?

If you haven't already read part 1 of my "Ultimate Guide to OSCP" series, it's here. Part 3 includes my approach to hacking new machines in the labs, a cheatsheet, and some other useful hacking tricks. You can find that here.

## Get in touch

If you have any suggestions or would like to stay in touch—the best way is to follow me on Twitter.