Homework 4: Censorship with CensorLab

26 November, 2024

CensorLab is a censorship simulation platform that allows researchers to test censorship circumvention tools against various censorship techniques. For this assignment, we have provided a VirtualBox VM with all the tools necessary to use CensorLab.

To get started,

1. Install VirtualBox
2. If you are using an x86_64 system (e.g., most Windows/Linux PCs, Macs with Intel processor), download this file
   - Import the OVA file into VirtualBox using the import feature (x86_64)
3. If you are using an aarch64 system (e.g., m1,2,3 Macs), download this file
   - NOTE: due to a bug in VirtualBox for aarch64 macs, OVA import will not be available. To create a VM on these systems, download the above file, create a new virtual machine with the following settings
     - 4+ CPU cores (Figure 2)
     - 4GB+ Memory (8-12GB recommended to avoid update issues)(Figure 2)
     - OS: Arch Linux (arm64)(Figure 1)
     - You may wish to adjust resolution and video memory for a larger screen
   - When choosing the disk, choose the VMDK file you downloaded above as the drive image instead of creating a new one (See Figure 3
4. Start the VM
5. On the desktop there will be two links
   - Firefox - the homepage of firefox will be a documentation page for using CensorLab. All information about CensorLab will be documented here
   - Terminal - This opens a terminal for running CensorLab, proxies, etc.
6. You will want to run an initial update on the system. Please run `censorlab-update` in the terminal. The password for `censorlab` is `c3ns0rl4b612@@!`. You may be requested to run this command again if bugs are found.

7. `vim` is preinstalled, but if you prefer vscode, you may install and run it on the VM by running `NIXPKGS_ALLOW_UNFREE=1 nix run --impure nixpkgs#vscode`.

8. For each assignment, create a folder (e.g. question1, question2) containing censor.toml and censor.py. Ensure censor.toml refers to censor.py. On the VM, /etc/censorlab-demos contains examples of such projects. For Q5, simply include question5.txt. Submit a zip file of everything to Gradescope.

9. Make sure to refer to the documentation (Firefox link on desktop) for more info on the APIs in CensorLab

10. To run CensorLab with the given censor.toml, use `censorlab -c censor.toml nfq`

11. If you have questions or issues with CensorLab, contact`jsheffey@cs.umass.edu` or ask questions on Piazza

Question 1 [20pt]. You are a censorship programmer for Repressistan. The Massachusetts Institute of Technology (MIT) has released a scathing condemnation of your internet censorship policies. Write a CensorLab program that drops any DNS packets requesting addresses for mit.edu or any subdomain. You may test DNS lookups using the `nslookup` command:

```
nslookup example.com 8.8.8.8
```

Non-exhaustive examples of domains for which DNS requests that should time out due to droped packets:

- `mit.edu`
- `scripts.mit.edu`
- `stuff.mit.edu`

Non-exhaustive examples of DNS requests that should successfully receive a response (even if that response is canonically NXDOMAIN):

- `kermit.edu`
- `umass.edu`
- `google.com`

Question 2 [20pt]. The citizens of Repressistan have begun using encrypted DNS schemes to circumvent your censorship. However, they aren't using TLSv1.3 with ESNI, so their HTTP and HTTPS requests may contain plaintext domains. Write a CensorLab program to drop HTTP or HTTPS packets containing `mit.edu` or a subdomain of `mit.edu`.

Question 3 [20pt]. Repressistan's intelligence agencies have discovered that MIT is using alternative domain names to circumvent blocking. Analysts have discovered that MIT uses AS3 for its operations. Write a CensorLab program that drops all traffic to IPs associated with AS3. (`packet.ip.src` and `packet.ip.dst` may be useful)

Question 4 [20pt]. The citizens of Repressistan are using fully encrypted circumvention protocols to hide their traffic to all sorts of forbidden websites. Write a CensorLab program to drop ShadowSocks proxy traffic while leaving HTTPS traffic untouched (Hint - See Wu et al. [2023]). A configuration file to start a ShadowSocks proxy can be found here. You can use `wget` in the CensorLab VM to download it. It can be used by starting `sslocal -c shadowsocks.json` in a terminal. To make connections over this proxy, run `curl --proxy socks5h://localhost:1080 https://umass.edu`

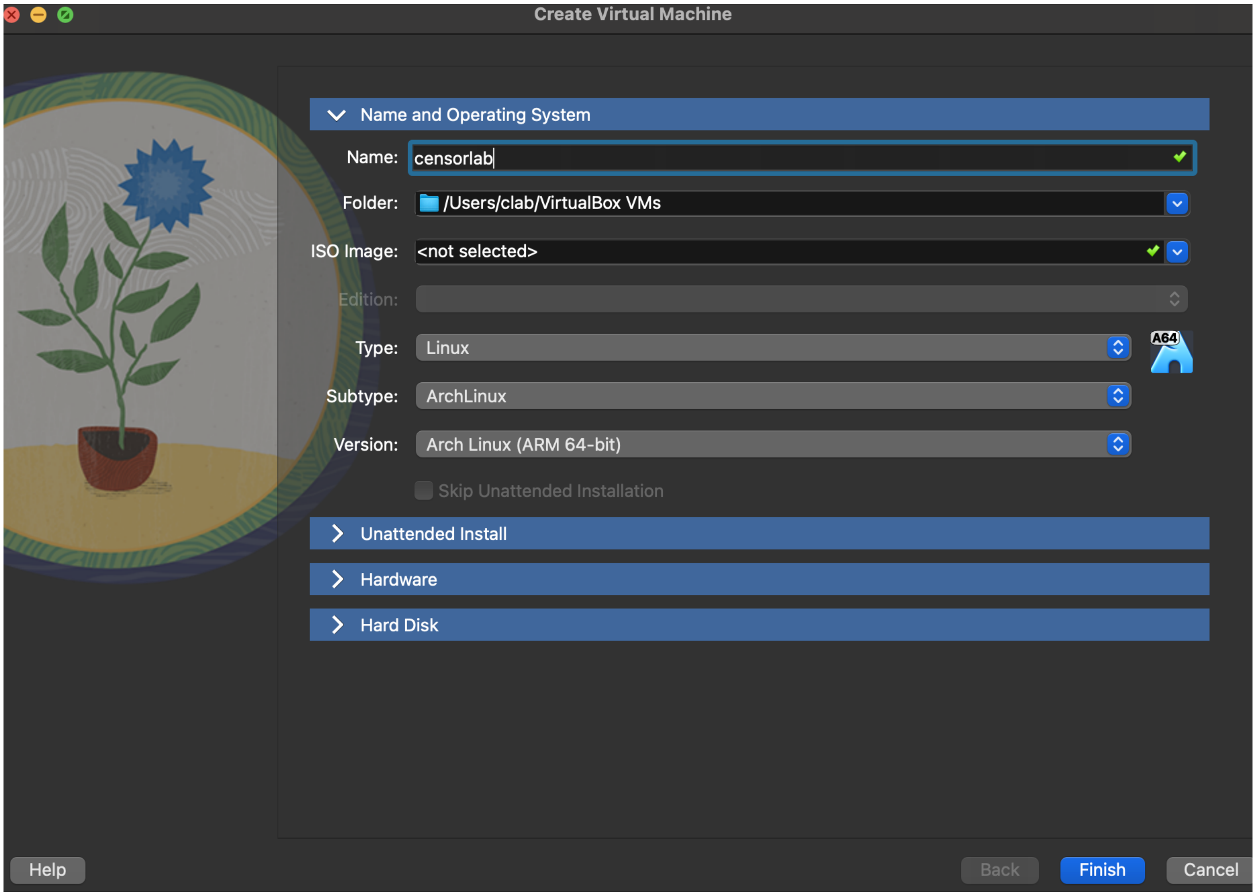
redPlease be mindful when using this server and do not send excessive bandwidth through it. To test whether your program successfully censors ShadowSocks, simple curl commands as shown above will suffice.

Question 5 [20pt]. How might the developers of fully encrypted circumvention protocols modify their protocol to defeat the censor program you've written in Question 4?

References

Mingshi Wu, Jackson Sippe, Danesh Sivakumar, Jack Burg, Peter Anderson, Xiaokang Wang, Kevin Bock, Amir Houmansadr, Dave Levin, and Eric Wustrow. How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2653–2670, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. URL `https://www.usenix.org/conference/usenixsecurity23/presentation/wu-mingshi`.

1 Appendix



Create Virtual Machine

Name and Operating System

Name: censorlab

Folder: /Users/clab/VirtualBox VMs

ISO Image: <not selected>

Edition:

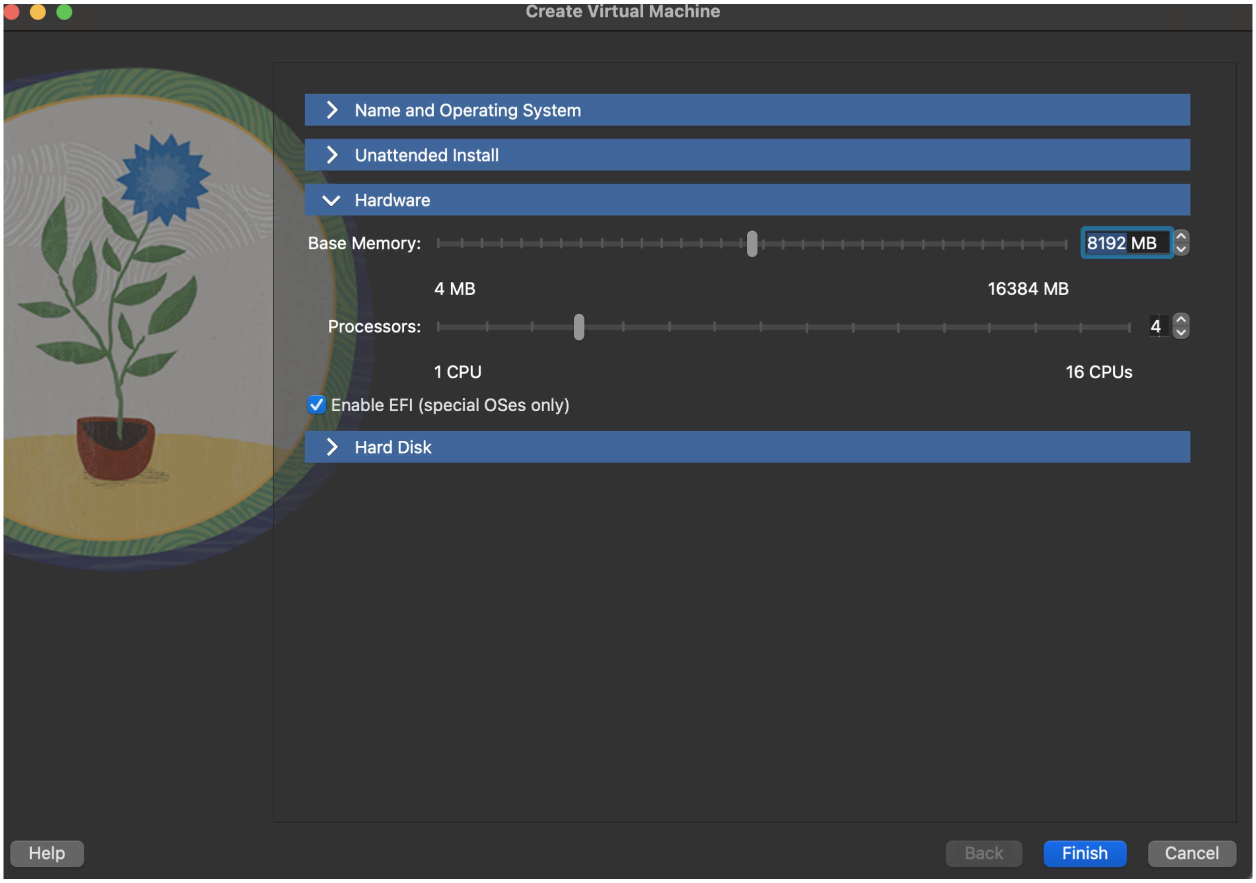
Type: Linux

Subtype: ArchLinux

Version: Arch Linux (ARM 64-bit)

□ Skip Unattended Installation

Figure 1: Initial VM config for OSX import

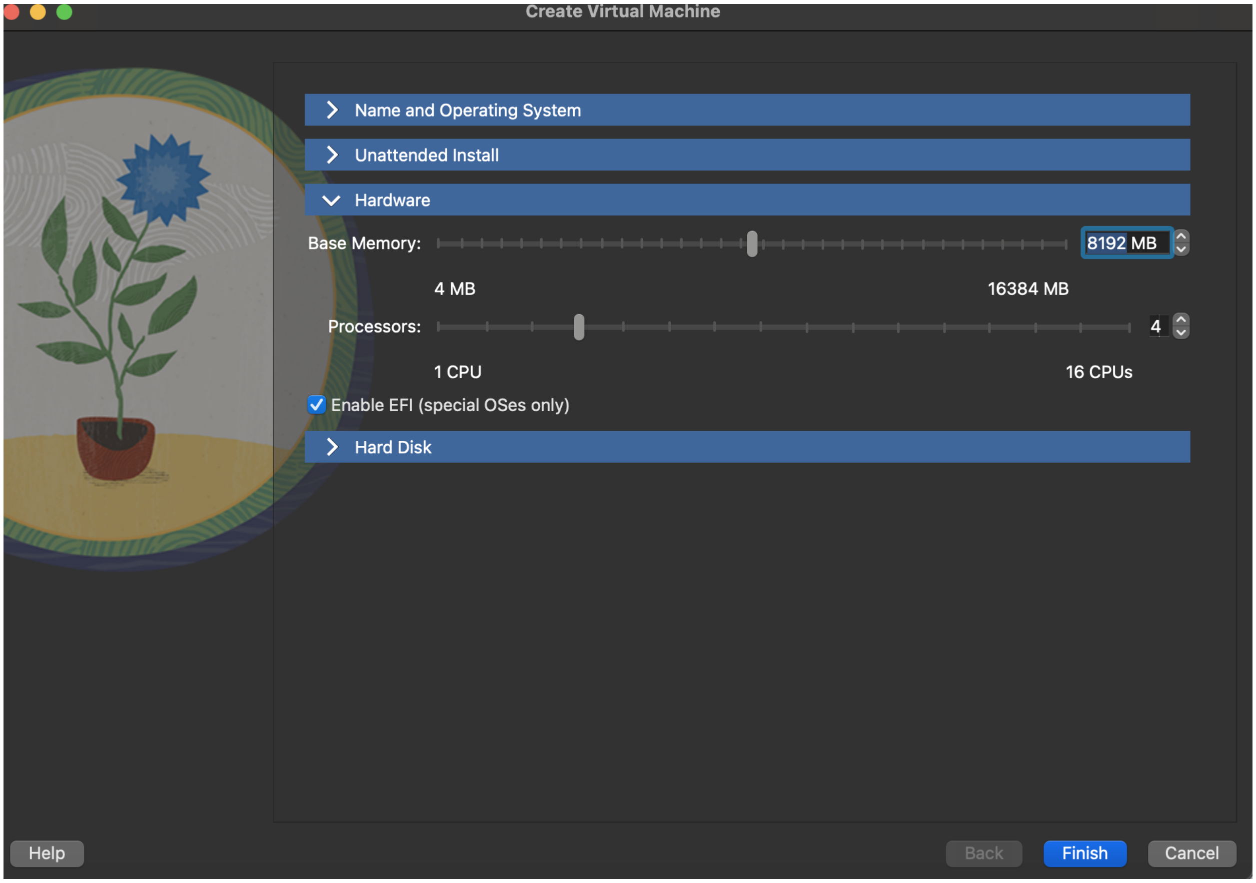


Base Memory: 8192 MB

Processors: 4

☑ Enable EFI (special OSes only)

Figure 2: VM spec config for OSX import



Base Memory: 8192 MB

Processors: 4

☑ Enable EFI (special OSes only)

Figure 3: VM hard drive config for OSX import