CS7038 - Malware Analysis - Wk01.2

VirtualBox Lab Setup and Crash Course

Coleman Kane
kaneca@mail.uc.edu

January 12, 2017

UNIVERSITY OF Cincinnati

VirtualBox

Virtualization is a feature present in nearly all consumer CPUs these days that enables you to virtualize a whole hardware system, using software features of your computer, yet still run at near-native execution speeds.

I'll be using Oracle's VirtualBox product (`https://virtualbox.org`) for the labs in this class, and you are strongly encouraged to learn this tool as well, even if you already are more familiar with the competing VMWare product from Dell. Some reasons I prefer VirtualBox:

- Open-sourced GPL community-driven project, with minimal usage restrictions
- Same UI across Windows, Linux, FreeBSD, MacOS
- Supports creating complex virtual network layouts
- Supports some Paravirtualization (PVM) features
- Full command-line interface alternative, fully scriptable

VirtualBox Resources

Since it is a community-supported project, with some vendor assistance from Oracle, there's an awful lot of documentation.

- **Download:**
  `https://www.virtualbox.org/wiki/Downloads`
- **User Manual:**
  `https://www.virtualbox.org/manual/UserManual.html`
- **Community Support:**
  `https://www.virtualbox.org/wiki/Community`

Once installed, a good place to visit would be the walkthroughs and overview in Chapter 1:

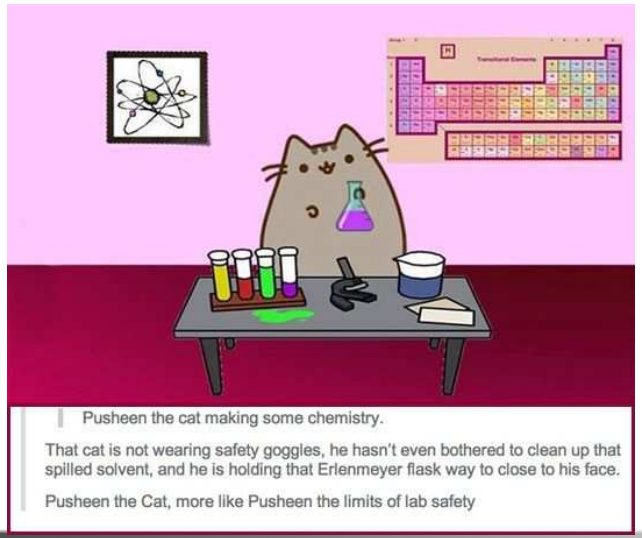
- `https://www.virtualbox.org/manual/ch01.html`

VirtualBox for Your Lab

The following features of VirtualBox will be helpful for your lab environment:

- Snapshots & Cloning
- Import/Export of Appliances
- Virtual networking options
- Shared Folders
- VirtualBox Guest Additions
- 2D (& even 3D) acceleration

A well organized malware lab is an effective malware lab. Always handle with care.



UNIVERSITY OF Cincinnati

Getting a Virtual Machine

There are three common methods to getting a Virtual Machine set up:

- **Import an appliance:** Imports disk image + VirtualBox system config in one bundle
- **Install from media:** Create an empty disk, custom VM, and use installation media supplied by OS provider, just like building a new computer
- **Attach Disk Image:** OS and software already installed on disk image, but need to configure a new VM to attach it to

The method used will vary depending upon how the media is provided to you from your software provider. We will cover the first two methods in our lecture.

Most Familiar: Install from media

It is most common to install software into VirtualBox using installation media, such as an ISO9660 DVD image.
We can easily get copies of GNU/Linux from various distribution websites.
For Windows images, those of you with a UC account can download the Windows 10 student edition using DreamSpark. Others may be able to download Windows 10 images from Microsoft's website, and use the product code provided with their laptop/desktop to install a "registered" OS.
GNU/Linux offerings:

- **Ubuntu Desktop:** `https://www.ubuntu.com/download/desktop`
- **Linux Mint:** `https://www.linuxmint.com/download.php`
- **LUbuntu Desktop:** `http://lubuntu.net/`

Unless specifically noted, it is recommended to always download the 64-bit images. No significant resource savings are gained by trying to use a 32-bit OS, and this might introduce complications.

Steps to install from media

This approach typically takes a few steps:

- Download install media
- Create new VM with proper configuration for your OS choice, and sufficient hard disk space
- Attach the downloaded ISO media to the virtual CD drive of the VM
- Boot the VM (you may need to press **F12** on boot if the VM doesn't automatically try booting off the CD)
- Walk through the install, *just like if you did it on a real computer*
- Upon completion, shut down the VM and then modify the VM configuration to "eject" the virtual CD

Notes on Installing Windows 10 Education Edition

When you install Windows 10 in your lab, you will want to be sure to turn off a lot of the telemetry features and some common security mitigations that may interfere with malware analysis.

During the first stage of setup, after install from media completed and the system reboots the first time, I chose the **customize settings** option and explicitly turned off **ALL** of the features that it asked about.

It may ask how you intend to connect to a network. I chose **Join a local Active Directory domain**. After that, it let me create a new local user account and never asked about Domain-specific credentials.

Easiest: OVA Import

In general, you download a file from the provider that has a `*.ova` extension. This single file contains the hard disk image(s) and all the configuration options that were chosen by the original VM author.

The process for importing these is straightforward:

- Download the OVA file
- In `VirtualBox` go to the File menu, and choose the *Import Appliance...* option
- Use the form to select the downloaded OVA file for import
- Tweak the configuration (if desired) prior to import
- Click *Import* button

Sources for OVA Images

Useful OVA sources:

- Microsoft Modern.IE Website `https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/`
- Offensive Security Kali Download `https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/`
- Remnux Distribution `https://remnux.org/#distro`

Remember - these are installed OS images, and typically have very basic default usernames/passwords and even running services that you may want to tweak before using them.

OVA Export

You may also choose to create your own OVA files. This can be very helpful after you've set up your VirtualBox labs with tools and configurations.

The process for exporting:

- In `VirtualBox` go to the File menu, and choose the *Export Appliance...* option
- Use the form to select the VM you'd like to export
- Choose the *OVF 1.0* format (the 2.0 format is experimental, as of 2017-01-12)
- Add descriptive information about your appliance
- Click *Export* button

You now have an appliance that can be imported into another VirtualBox session, or even re-imported into this one.

Snapshots

A VM execution instance consists of three components:

- Persistent storage (disk)
- VM configuration (HW to simulate, behavior, etc.)
- Run-time state (what's going on in memory)

The **snapshots** feature in VirtualBox allows us to save copies of varying run-time versions of suspended and powered-off virtual machines, enabling us to revert back to them in the future.

This functionality will be used heavily when doing run-time execution analysis of malware samples. Since malware frequently can make irreperable and even unexpected changes to a system, the feature enabling reverting to a "*known clean*" state is very helpful, in that it eliminates unnecessary VM rebuilds.

Virtual Networking

VirtualBox offers numerous network configurations, and each VM can have up to 4 virtualized network interfaces configured each to be any one of the following:

- Detached
- **NAT**
- NAT Network
- Bridged
- **Internal Network**
- **Host-only Network**
- 3rd-party driver support

Ones that we will primarily focus on during class are highlighted in bold above.

More elaborate discussion of the above modes is provided here:
`https://www.virtualbox.org/manual/ch06.html`

Functional OVA Images

As with OVA files we can now distribute pre-installed OS images, we also have the capability to prepare the OS environment for specific purposes. It is becoming increasingly popular to distribute function-specific OS configurations.

Some of these are listed below:

- **Kali**: A linux distribution that is preloaded with a number of common penetration-testing tools, including malware creation - `https://www.kali.org`
- **Remnux**: A linux distribution pre-loaded with tools for analyzing malware and system forensics - `https://remnux.org`
- **VulnHub**: An archive of a large amount of pre-configured vulnerable system images that can be attacked with malware. Many distributed as OVA files - `https://www.vulnhub.com`