



track0 Install_Details

Joe Thielen edited this page on 30 Sep 2016 · 204 revisions

These are the detailed steps required to set up a track0 install.

⚠ IMPORTANT NOTE! These instructions are currently in *DRAFT* form. These may change significantly until they are stabilized. *Track0 will never be meant for production use!*

Table of Contents (TOC)

- [Before starting...](#)
 - [Timezone considerations](#)
- [Server hardware selection](#)
- [Install QubesOS](#)
 - [Qubes 3.1 Initial Setup](#)
 - [Qubes 3.2 Initial Setup](#)
 - [Post-Qubes Install](#)
 - [Updating Qubes](#)
 - [Determine Network Info](#)
- [Create TPF Firewall VM](#)
- [Create TPF Work VM](#)
 - [Enable VM automatic shutdown on system shutdown](#)
- [Create stock CentOS HVM](#)
 - [Obtain CentOS 7 Minimal ISO and put on work VM.](#)
 - [Create a blank Qubes HVM](#)
 - [Install CentOS 7 Minimal on blank HVM](#)
 - [Fix an issue with Qubes and CentOS HVMs](#)
 - [Install updates](#)
 - [Install additional recommended software by this project](#)
 - [Further setup](#)
 - [Wrapping up](#)
 - [Enable networking between tpf-stock_centos_7 and tpf-work](#)
- [Create TPF Proxy VM](#)
 - [Clone stock CentOS HVM to new TPF Proxy VM](#)
 - [Configure networking / hostname / hosts](#)
 - [Enable networking between tpf-proxy and tpf-work](#)
 - [Set up ModSecurity](#)
 - [Set up mod_proxy](#)
 - [Enable VM autostart on boot and crash](#)
- [Create TPF IPA Server VM](#)
 - [Clone stock CentOS HVM to new TPF IPA Server VM](#)
 - [Configure networking / hostname / hosts](#)
 - [Set up FreeIPA](#)
 - [Enable VM autostart on boot and crash](#)

Pages 15

- [Home](#)
- [Images and Logos](#)
- [Installing Red Hat 389 Directory Server \(DS389\) on CentOS 7 \(1511\) Minimal](#)
- [Qubes Commands](#)
- [track0](#)
- [track0 Ideas_and_Questions](#)
- [track0 Install_Details](#)
- [track0 Install_Overview](#)
- [track0 Overview](#)
- [track0 settings_cheat_sheet](#)
- [track1](#)
- [track1 Ideas_and_Questions](#)
- [Track1 Install_Details](#)
- [track1 settings_cheat_sheet](#)
- [Tracks](#)

Clone this wiki locally


<https://github.com/JoeThielen/ThePlatformFramework/wiki>

Clone in Desktop

- [Create TPF Test Project VM](#)
 - [Clone stock CentOS HVM to new Test Project VM](#)
 - [Configure networking / hostname / hosts](#)
 - [Enable networking between *tpf-test* and *tpf-work* and *tpf-proxy* and *tpf-ipa*](#)
 - [Enable VM autostart on boot and crash](#)
- [Create test web app](#)
 - [Install Apache & PHP](#)
- [Configure networking / port forwarding](#)
- [Other / Optional VMs](#)

Before starting...


Installing *TPF* is not a simple process. There are many, many steps. And they must be done mostly in order. Therefore it may be advisable to print out the following two documents:

- [Track0-Install Overview](#) - This can be used to *check off* items as you accomplish them. Also for taking notes.
-  [Track0-Settings Cheat Sheet](#) - This can be used to write down important information about your install. *PLEASE PRINT THIS BEFORE STARTING!*
 - Passwords - There are more than a few!
 - IP addresses - There are more than a few of these too!
 - Other various settings - There are more than a *more than a* few of these.

[Return to TOC](#)

Timezone considerations

Before starting this project you should think about what timezone you wish to use. It's recommended for uniformity that you use https://en.wikipedia.org/wiki/Coordinated_Universal_Time, especially if your project is going to be used by people from outside of your local area. In the USA, too often, we tend to think very locally within our own timezone and not realize others live in different timezones, as well the fact that when dealing with computers and data it's good to have and use a standard. So, if you're project is only ever going to be used by one organization / business and all it's employees are local, it may be fine to choose your local timezone. However, it's recommended, especially for a new project starting from scratch, to use *UTC* for all internal functions, then when displaying or reporting a date to transform that into whatever the current users local/preferred timezone is.

- Find the name and offset for your timezone from the [Wikipedia list of timezones](#).
 - Note the values in the *TZ* and *UTC offset* fiends (not *UTC DST offset*).
 - If you're going to use *UTC*, the *TZ* value is *Etc/UTC* and the offset is *+00:00*
 -  If you're using the [track0 settings cheat sheet](#), write these down for the *TPF-TZ* and *TPF-TZ-OFFSET* setting values.

Server hardware selection

This install requires a dedicated hardware computer. You can not use a VM, since the primary system OS is a VM hypervisor itself.

Since there will be more than a few VMs running on this machine, it should be as fast a system as you can get, with as much memory as you can pack into it.

- Qubes Compatibility
 - Because the system will be based around the Qubes OS, it's extremely important to try and ensure the hardware you select is known to be compatible with it. The Qubes project maintains it's own *Hardware Compatibility List (HCL)* which is a great reference.
 - [Qubes System Requirements](#)
 - [Qubes Hardware Compatibility List](#)
 - It is very helpful to the Qubes project and other Qubes users to know about working (and non-working) hardware configurations! This will also benefit other *The Platform Framework* users.
 - [Qubes: How do I Submit a Report to the Hardware Compatibility List?](#)

- CPU
 - In order to use the hardware VM features, you'll need to get a CPU with:
 - [Intel VT-x or AMD-V](#)
 - [Intel VT-d/IOMMU](#)
 - It may be necessary to enable these settings in the system BIOS / setup before they can be used by the OS!
 - For Intel, this means a Core i5 or Core i7.
 - For AMD, this means 🚩 ??? - *need information here*
 - Additionally, due to multiple VMs running simultaneously, it's recommend *at least* to use a quad-core CPU.
- RAM
 - The more memory the better. It's recommended *at least* to use 16GB of RAM
- Storage
 - 🚩 ??? - *need information here*
- Networking
 - 🚩 ??? - *need information here*
 - Multiple networking adapters may enable the complete separation of TPF network traffic within qubes. i.e., creating a totally separate *net VM*. See [Track0-Ideas and Questions wiki page for more information](#)

[Return to TOC](#)

Install QubesOS

🚩⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *Install QubesOS*), are not yet complete!

- First, of course, you must download a copy of Qubes. The downloads and instructions are here: <https://www.qubes-os.org/downloads/>
 - These instructions were created for Qubes version 3.1 and 3.2(-rc2 & -rc3). They may very well work for future versions as well.
- ⚠️ **IMPORTANT NOTE!** If the target machine has *Intel VT-x/AMD-V* or *Intel VT-d/IOMMU* you may need to turn these on in the computers *BIOS*. The instructions for doing so are beyond the scope of this document.
 - ⚠️ If you do not do this, you will be losing out on important functionality related to security!!!
 - i.e., it may be possible for VMs to break into each other and/or the base hypervisor. This is not advisable in any way shape or form.
- After putting Qubes on a USB drive (or DVD), boot the target machine with the Qubes media.
- On the first screen choose *Install Qubes R3.x*.
 - Alternatively, you may of course also choose *Test media and install Qubes R3.x* option.
- Choose your language.
- On the *INSTALLATION SUMMARY* screen:
 - Click on *TIME & DATE*
 - Choose your timezone as discussed above in the [Timezone considerations](#) section.
 - If selecting *UTC*, select *Etc* in *Region*, then *Coordinated Universal Time* in *City*.
 - Click the *Done* button in the upper-left hand corner.
 - Click on *SOFTWARE SELECTION*
 - On the right-hand side, under *Add-Ons for Selected Environment*, uncheck *ALL* items.
 - Including *Debian 8* and *Whonix*.
 - We will not be using these.
 - Click the *Done* button in the upper-left hand corner.
 - Click on *INSTALLATION DESTINATION* under *SYSTEM*.
 - Choose which disk(s) you wish to install Qubes to by clicking on them.
 - If a given disk has been marked correctly, it will have a checkmark on it.
 - ⚠️ Qubes 3.2 Specific: Uncheck the *Encrypt my data*. checkbox on the left near the bottom!
 - ⚠️ **IMPORTANT NOTE!** If you leave this box checked, then your system will not be able to boot by itself, without someone entering a password. So if there is a storm

or the power is interrupted for any reason and the machine reboots, it will just sit there awaiting a human being to enter the password!

- Click the *DONE* button.
- On the *INSTALLATION OPTIONS* pop-up window:
 - ⚠ Qubes 3.1 Specific: Uncheck the *Encrypt my data* checkbox!
 - ⚠ *IMPORTANT NOTE!* If you leave this box checked, then your system will not be able to boot by itself, without someone entering a password. So if there is a storm or the power is interrupted for any reason and the machine reboots, it will just sit there awaiting a human being to enter the password!
 - ⚠ If there are existing partitions, then you should click the *Reclaim space* button in the lower-right hand corner.
 - ⚠ You should not leave existing data on this new server. You should be installing Qubes from scratch and let it take over the whole system.
 - On the *RECLAIM DISK SPACE* screen, click on the *Delete all* button on the right-hand side near the bottom.
 - The click the *Reclaim space* button in the lower-right hand corner.
 - Click the *Continue* button.
- Back on the *INSTALLATION SUMMARY* screen click on the *Begin Installation* button in the lower-right hand corner.
- ⚠ On the *CONFIGURATION* screen you DO NOT need to click on *ROOT PASSWORD*.
 - ➡ DO NOT SET A ROOT PASSWORD
- ⚠ Qubes 3.2 Specific: Click on *USER CREATION*
 - Create a user named *tpfadmin*
 - Uncheck the *Require a password to use this account* option.
 - ➡ DO NOT create a password for this user!
 - 🐭 If you're using the [track0 settings cheat sheet](#), write down *NONE* for the *QUBES-USERPASS* setting value.
 - Click the *Done* button in the upper-left hand corner.
- *Wait until installation has completed...*
- When it has competed click the *Reboot* button in the lower-right hand corner.
 - ⚠ You may need to remove the Qubes installation media, or set your device boot order to the correct boot drive in BIOS.

[Return to TOC](#)

Qubes 3.1 Initial Setup

⚠ This is Qubes 3.1 Specific!

- After reboot you will arrive at the initial *Welcome to Qubes* screen.
- Click the *Forward* button in the lower-right hand corner.
- On the *License Information* screen click the *Forward* button in the lower-right hand corner.
- On the *Create User* screen create a user named *tpfadmin*
 - Create a password for this user.
 - 🐭 If you're using the [track0 settings cheat sheet](#), write this down for the *QUBES-USERPASS* setting value.
 - Click the *Forward* button in the lower-right hand corner.
- On the *Date and Time* screen, ensure the date and time are correct, then click the *Forward* button in the lower-right hand corner.
- On the *Create VMs* screen:
 - Ensure the *Create default system qubes* checkbox is checked.
 - *Uncheck* the *Create default application qubes* checkbox.
 - ⚠ NOTE: If you are not familiar with Qubes and/or you're only exploring *TPF* right now, you may leave this checkbox checked if you wish. It will simply create some extra VMs that we aren't going to use for *TPF*, but you may find them helpful in learning Qubes. However, when setting up Qubes for production *TPF* use, it is advised to *NOT* have these VMs.
 - Click the *Finish* button in the lower-right hand corner.
 - *Wait until configuration has completed...*

Qubes 3.2 Initial Setup

⚠ This is Qubes 3.2 Specific!

- After reboot you will arrive at the *INITIAL SETUP* screen.
- Click on *QUBES OS* under *SYSTEM*.
 - Ensure the *Create default system qubes* checkbox is checked.
 - *Uncheck* the *Create default application qubes* checkbox.
 - ⚠ NOTE: If you are not familiar with Qubes and/or you're only exploring *TPF* right now, you may leave this checkbox checked if you wish. It will simply create some extra VMs that we aren't going to use for *TPF*, but you may find them helpful in learning Qubes. However, when setting up Qubes for production *TPF* use, it is advised to *NOT* have these VMs.
 - Click the *Done* button in the upper-left hand corner.
 - *Wait until configuration has completed...*
- Back on the *INITIAL SETUP* screen, click the *FINISH CONFIGURATION* button in the lower-right hand corner.

[Return to TOC](#)

Post-Qubes Install

- You will now be asked to log in. The username should already be pre-populated with the *tpfadmin* user.
 - ⚠ Qubes 3.1 Specific: Enter the password we created above.
 - 🐭 If you're using the [track0 settings cheat sheet](#), this is *QUBES-USERPASS* setting value.
 - ⚠ Qubes 3.2 Specific: We did not create a password, so just click the *Log In* button.
- After logging in you should now be at the Qubes graphical user interface.
 - ⚠ Qubes 3.2 Specific: On the *Welcome to the first start of the panel* dialog, click the *Use default config* button.
 - The *Qubes VM Manager* window should be open which shows *ALL* VMs & HVMs.
 - Click the last icon on the right... it looks like a circle which is half green. This will show *ALL* VMs, regardless of if they're active or not.
 - Click *View* then Check the box for *IP*.
 - This is helpful, as it will give you the Qubes IP address of each Qube/VM/HVM.
 - There should only be four VMs listed at this point (unless you let Qubes create the other VMs above):
 - *dom0* is the user interface VM. i.e., what you're using now.
 - *sys-net* is the VM which handles network traffic and is the *ONLY* VM which will have direct contact with the physical network interface.
 - *sys-firewall* is the firewall VM. We will actually be copying this and creating our own firewall VM specifically for *TPF*. This will be explained later.
 - *fedora-23* is also called a *template VM* because it can be used as a template for other VMs.
- The first thing we need to do is set Qubes up to automatically log in as the *tpfadmin* user.
 - NOTE: Helpful information taken from: <https://groups.google.com/forum/#!topic/qubes-users/CTSzbNHSgBU>
 - If we do not do this then the server will not be able to boot by itself, i.e., after a power outage.
 - Click on the *Application Launcher Menu*
 - ⚠ Qubes 3.1 Specific: This is in the lower-left hand corner of the screen. This is a small square icon with a Q in it. Then click on *System Tools* and *Konsole (Terminal)*
 - ⚠ Qubes 3.2 Specific: This is in the upper-left hand corner of the screen. This is a small square icon with a Q in it. Click on *Terminal Emulator*.
 - ⚠ This is the *dom0* terminal. This is where all major Qubes configuration is done.

- You may need to become the superuser to do certain tasks.

- To do so, issue this command:

```
sudo su
```

- Edit the `/etc/lightdm/lightdm.conf` file.

- Find these lines and uncomment them.




```
pam-service=lightdm
pam-autologin-service=lightdm-autologin
```

- Then find these lines and uncomment them. Also, ensure to add `tpfadmin` to the `autologin-user=` line as noted below.

```
autologin-user=tpfadmin
autologin-user-timeout=0
```


- Now reboot the machine to ensure what we've done works.

- To do so:

- Click on the *Application Launcher Menu*.
 - Click on the *Leave* option (Qubes 3.1) or *Log Out* option (Qubes 3.2) at the bottom of the menu.
 -  Qubes 3.2 Specific: Uncheck the *Save session for future logins* checkbox.
 - Click on *Restart*.
 -  Qubes 3.1 Specific: Click on the *Restart Computer* text/icon.
 - The server will now reboot.
 - When finished you should be brought back to the Qubes GUI without having to log in!
 -  If you are asked to log in again, you must go back and ensure the `/etc/lightdm/lightdm.conf` file has been edited correctly! DO NOT PROCEED UNTIL THIS HAS BEEN SUCCESSFULLY COMPLETED!!!

- Now we need to set up a static IP address. This machine will be a server with other users interacting with it from outside the machine. We will also be setting up firewall rules later which require a static IP.

- To configure a static IP:

- Right-click on the on the networking icon in the system tray
 - Click *Edit Connections*.
 - Click on your active network connection.
 - i.e., *Ethernet - Wired connection 1*
 - Click the *Edit* button to the right.
 - Click on *IPv4 Settings*.
 - In the *Method* drop-down select *Manual*.
 - In the *Addresses* section click *Add*.
 - Add the IP address, netmask (typically `255.255.255.0`) and gateway.
 -  If you're using the [track0 settings cheat sheet](#), write down the IP address for the `QUBES-EXTIP` setting value.
 - Add one or more DNS servers in *DNS servers*.
 - Usually you can use the gateway IP address, or, if failing that, use Google's free DNS service at the following IP addresses:
 - `8.8.8.8` (*Primary*)
 - `8.8.4.4` (*Secondary*)
 - Click the *Save* button.
 - Click the *Close* button.

- Reboot!

Updating Qubes

- ⚠️ DO NOT SKIP THIS PROCESS! It is *extremely* important to keep your system up to date!
- NOTE: Helpful information taken from: [Updating Software in dom0](#)
 - Additional helpful Qubes documentation: [Common Tasks](#)
- Update *dom0*
 - Start a *dom0* terminal.
 - Issue this command:

```
sudo qubes-dom0-update -y
```

 - *This may take a while!*
 - If you are asked *Is this ok*, type `y` and press ENTER.
 - Because we've updated a major portion of the system we will now need to reboot and make sure everything still functions correctly.
 - Reboot!
- Update the *Fedora Template*
 - In the *Qubes VM Manager* window, find the *fedora-23* VM.
 - If you do not see it, click the last icon on the right... it looks like a circle which is half green. This will show ALL VMs, regardless of if they're active or not.
 - Right-click on the *fedora-23* VM and select *Update VM*.
 - ⚠️ If *Update VM* is grayed out and not available, first try *Start/Resume VM*... after a bit the *Update VM* option should become available.
 - A new window should pop up which runs the package update manager.
 - If you are asked *Is this ok*, type `y` and press ENTER.
 - *This may take a while!*
 - ⚠️ *This may take a VERY long while!*
 - Because we've updated a major portion of the system we will now need to reboot and make sure everything still functions correctly.
 - Reboot!

[Return to TOC](#)

Determine Network Info

Before you get too far along, it will be helpful to know the following network-related info.

- Open up a terminal in the *sys-net* VM
 - To open up a terminal in the *sys-net* VM (or *sys-firewall* VM)..
 - Click on the *Application Launcher Menu*.
 - Click on *ServiceVM: sys-net*
 - Click on *sys-net: Terminal*
 - Run this command to get a list of network interfaces and their IP addresses..

```
sudo ifconfig | grep -i cast
```
 - Find the interface which has the machines IP address and note it's name.
 - This is the one with the static IP you set up during Qubes setup.
 - This is the outermost IP which gets you out of the machine and onto your physical network.
 - Example: Interface name: `enp0s1` - IP address: `10.0.1.104`
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write down the interface name for the `QUBES-SYSNET-IFNAME` setting value.

[Return to TOC](#)

Create TPF Firewall VM

After installing and updating Qubes, we're going to create a new firewall VM which is specific to *TPF*.

But wait, doesn't Qubes already come with a firewall VM (*sys-firewall*)? Yes, it sure does. However, *especially during track0 research and development*, it's likely that you will wish to explore Qubes and its functionality. Since Qubes can automatically come with other VMs (*work, personal, vault, etc...*), we want to separate and compartment/isolate them from *TPF*.

- Open a terminal in Dom0
- Create the firewall VM as a proxy VM:
 - `qvm-create tpf-firewall --proxy --label gray`
- Then link our new firewall VM to the system net VM.
 - `qvm-prefs -s tpf-firewall netvm sys-net`
- Set the firewall to automatically start when Qubes boots up
 - `qvm-prefs -s tpf-firewall autostart true`
- Get the IP address for the firewall VM:
 - `qvm-ls -n tpf-firewall`
 - The firewall IP address is in the *ip* column
 - 📌 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-FIREWALLVM-IP* setting value.
 - The IP address in the *ip_back* column is the one which all other TPF VMs will use as a gateway
 - 📌 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-NETWORK-GW* setting value.
- Start the VM
 - In the *Qubes VM Manager* right-click on *tpf-firewall* and click *Start/Resume VM*
 - Alternatively you can issue the following command in a *dom* terminal:

```
qvm-start tpf-firewall
```

[Return to TOC](#)

Create TPF Work VM

NOTE: This step is not strictly necessary. This step is also not recommended if you are setting up a production environment. However, especially during *track0* creating a *work* VM for use strictly within the TPF environment can aid you when doing development. It can provide you with a method to use Firefox and other programs.

- Clone the system *fedora-23* VM and set netvm to *tpf-firewall*
 - From *dom0* console:

```
qvm-clone fedora-23 tpf-work
```

- This will take a few moments to complete as it is copying files.

```
qvm-prefs -s tpf-work netvm tpf-firewall
qvm-prefs -s tpf-work label gray
```

- Get the IP address for the new VM:

```
qvm-ls -n tpf-work
```

- The IP address is in the *ip* column
 - 📌 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-WORKVM-IP* setting value.
- Allow network/Internet access:

```
qvm-firewall tpf-work --policy=allow --icmp=allow --dns=allow
```


- Start the VM

```
qvm-start tpf-work
```

- Create a shortcut to a browser in the *tpf-work* VM.
 - Click the *Application Launcher Menu*
 - Click on *Template: tpf-work*.
 - Click on *tpf-work: Add more shortcuts...*
 - From the items in the *Available* box click on *Firefox* then click the `>` button.
 - Click the *OK* button.
 - ⚠ NOTE: You can now access the *Firefox* web browser within the *TPF* network by:
 - Click the *Application Launcher Menu*
 - Click on *Template: tpf-work*.
 - Click on *tpf-work: Firefox*
- Enable X11 forwarding for the SSH client on *tpf-work*.
 - Start a terminal in the *tpf-work* VM.
 - Edit the `/etc/ssh/ssh_config` file.
 - Find this line:

```
# ForwardX11 no
```

- Change it to:

```
ForwardX11 yes
```

[Return to TOC](#)

Enable VM automatic shutdown on system shutdown

Qubes has an issue if certain VMs/HVMs are left running during system shutdown. Therefore, we need a custom way to have it automatically shutdown properly. We can achieve this via *systemd*.

- In a terminal in *dom0*:
 - Create a file named `/usr/lib/systemd/user/tpf-work-hvm-shutdown.service` with the following contents:

```
[Unit]
# This script ensures that the HVM will be stopped at user logout/shutdown,
# otherwise the system will hang on shutdown/reboot.
Description=Stop the tpf-work VM on user logout/system shutdown.
Before=systemd-exit.service

[Service]
# This is a oneshot type of service, as we need it to pay attention to our
# "Before" requirement in the above "Unit" section. A simple type of service
# will not do so.
Type=oneshot
# We need a dummy program to run at startup, as we're only actually focused on
# shutdown.
ExecStart=/bin/true
# In order for the service to "stay running" we need to set "RemainAfterExit".
RemainAfterExit=yes
# This is what actually shuts down the HVM.
ExecStop=/bin/sudo /bin/qvm-shutdown tpf-work --force --wait

[Install]
WantedBy=default.target
```

- Then enable these service using this command:
 - ⚠ **IMPORTANT NOTE!** These commands should be executed in the *dom0* terminal when NOT in sudo nor su mode (*i.e., not as root or a user with elevated privileges*)! This is because the second command needs to run as the *tpfadmin* user itself.

```
systemctl --user enable tpf-work-hvm-shutdown
systemctl --user start tpf-work-hvm-shutdown
```

[Return to TOC](#)

Create stock CentOS HVM

Obtain CentOS 7 Minimal ISO and put on *work VM*.

- Direct method (via wget)
 - Start terminal from the *tpf-work* VM
 - Start Menu -> Template: tpf-work -> tpf-work: Terminal
 - Issue these commands:
 - `cd ~/Downloads`
 - `wget http://mirrors.kernel.org/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso`
 - Verify the SHA256 checksum/hash (from the [CentOS 7 ISOs checksum page](#)):
 - `wget http://buildlogs.centos.org/rolling/7/isos/x86_64/sha256sum.txt.asc`
 - `sha256sum -c sha256sum.txt.asc 2>/dev/null | grep CentOS-7-x86_64-Minimal-1511.iso`
 - This should return:
 - `CentOS-7-x86_64-Minimal-1511.iso: OK`
 - Manual method
 - Start Firefox from the *tpf-work* VM
 - Start Menu -> Template: tpf-work -> tpf-work: Firefox
 - [List of mirrors for CentOS 7 \(1511\) Minimal](#)
 - Choose a mirror and download to */home/user/Downloads* (*~/Downloads*)
 - Alternatively: [CentOS download page](#)

[Return to TOC](#)

Create a blank Qubes HVM

- From a terminal in *Dom0*:
 - Create the new HVM named *tpf-stock_centos_7*
 - `qvm-create tpf-stock_centos_7 --hvm --label gray`
 - Link the new HVM to Net VM *tpf-firewall*
 - `qvm-prefs -s tpf-stock_centos_7 netvm tpf-firewall`
 - Allow the HVM to have access to 1GB of RAM.
 - Your *CentOS* will thank you for this.

```
qvm-prefs -s tpf-stock_centos_7 memory 1024
```

- Get the IP addresses assigned to this HVM (*assigned by Qubes regardless of if OS is loaded yet*)
 - `qvm-ls -n tpf-stock_centos_7`
 - The IP is in the *ip* column
 - 🛠 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-STOCKVM-IP* setting value.

- The gateway (and DNS IP) in the *gateway/DNS* column should match the value we wrote down for *TPF-NETWORK-GW*

[Return to TOC](#)

Install CentOS 7 Minimal on blank HVM

- Start the HVM using the CentOS 7 ISO as a virtual CDROM drive (from *dom0* terminal):
 - `qvm-start tpf-stock_centos_7 --cdrom=tpf-work:/home/user/Downloads/CentOS-7-x86_64-Minimal-1511.iso`
- By default, the first screen for CentOS 7 asks you if you wish to test the media. Since we already did a sha256 checksum/hash it's not likely there is an issue. Of course, you can choose to do this if you wish.
 - Choose *Install CentOS 7*
 - *Screenshot 01*
- Choose your language
 - *Screenshot 02*
- You should now see the *INSTALLATION SUMMARY* screen.
 - *Screenshot 03*
 - Depending upon your screen resolution, there may be a vertical scroll bar leading to additional items (especially *NETWORK & HOST NAME*)
 - *Screenshot 04*
- Configure Networking
 - Click on *NETWORK & HOSTNAME*, near the bottom of the *INSTALLATION SUMMARY* screen.
 - *Screenshot 05*
 - Enter the hostname *stock.theplatformframework.com* in the *Host name:* field in the bottom-left hand corner of the screen.
 - *Screenshot 06*
 - Click the *Configure* button in the lower-right hand corner of the screen.
 - Click the *General* tab near the top.
 - Check the checkbox which reads *Automatically connect to this network when it is available*
 - *Screenshot 07*
 - Click the *IPv4 Settings* tab near the top. *Screenshot 08*
 - From the *Method* drop-down, select the *Manual* option.
 - Click the *Add* button to the right of the *Addresses* section.
 - Enter the IP address assigned by Qubes.
 - 📌 If you're using the [track0 settings cheat sheet](#), this is *TPF-STOCKVM-IP* setting value.
 - Enter the *Netmask* value of *255.255.255.255*
 - ⚠ DO NOT USE *255.255.255.0*!
 - Enter the *Gateway* value, which is from the TPF Firewall VM
 - *Not the external IP from the firewall, but the ip_back value*
 - 📌 If you're using the [track0 settings cheat sheet](#), this is *TPF-NETWORK-GW* setting value.
 - In the *DNS servers:* field, enter the same IP we just entered for *Gateway*.
 - 📌 If you're using the [track0 settings cheat sheet](#), this is *TPF-NETWORK-GW* setting value.
 - *Screenshot 09*
 - Click the *Save* button.
 - The network interface should now show as *ON* and *Connected*, using the details we just set up.
 - ⚠ Note: If it does not show as *ON* nor *Connected*, then click the *OFF* to turn it *ON*.
 - *Screenshot 10*
 - Click the *Done* button in the upper-left hand corner of the screen.
- Configure Date/Time
 - Click on *DATE & TIME*, near the top of the *INSTALLATION SUMMARY* screen.
 - *Screenshot 11*

- Ensure the *Network Time* in the upper-right hand corner of the screen is set to *ON*.
 - If it is not, you can NOT change it manually, there is something wrong with the networking setup, you must go back and fix the networking, then come back to this step!
 - ☹ DO NOT SKIP THIS STEP! ENSURING COORDINATED TIME BETWEEN ALL VMs IS EXTREMELY IMPORTANT AND MAY CAUSE SERIOUS PROBLEMS AND HEADACHES DOWN THE ROAD IF IGNORED HERE!
 - Choose the timezone you selected at the start of the project.
 - 📁 If you're using the [track0 settings cheat sheet](#), this is *TPF-TZ* setting value.
 - Click the *Done* button in the upper-left hand corner of the screen.
- Configure storage
 - Click on *INSTALLATION DESTINATION*, near the bottom of the *INSTALLATION SUMMARY* screen.
 - *Screenshot 12*
 - In the *Local Standard Disks* section there should be two drives.
 - The first one should be something like *20GiB* in size.
 - The second one will be much smaller, something like *2048 MiB* (2GiB).
 - Click on the *20 GiB* drive.
 - After doing so a checkmark should appear on it.
 - *Other Storage Options* section, check the *I will configure partitioning* option.
 - *Screenshot 13*
 - Click the *Done* button in the upper-left hand corner of the screen.
 - You will now see a screen for *MANUAL PARTITIONING*
 - *Screenshot 14*
 - In the section labeled *New CentOS 7 Installation*:
 - In the middle there is a drop-down that usually defaults to *LVM*.
 - Change this, select *Standard Partition* from the drop-down list.
 - Click the link for *Click here to create them automatically*
 - It should now have created three entries:
 - */boot*
 - */*
 - *swap*
 - *Screenshot 15*
 - Click the *Done* button in the upper-left hand corner of the screen.
 - A window labeled *SUMMARY OF CHANGES* will pop up. Click the *Accept Changes* button.
 - At this point we should be ready to continue with the installation. Click the *Begin Installation* in the bottom-right hand corner of the screen.
 - *Screenshot 16*
 - Now we need to set up two users.
 - First we need to set up the *root* admin user.
 - Click on *ROOT PASSWORD*
 - *Screenshot 17*
 - Enter a password for the root user.
 - 📁 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-STOCKVM-ROOTPASS* setting value.
 - Click the *Done* button in the upper-left hand corner of the screen.
 - Click on *USER CREATION*
 - ☹ DO NOT SKIP THIS STEP - We will be disabling network root access to our HVMs!
 - Therefore we will require a secondary user.
 - *Screenshot 18*
 - Enter *tpf* in the *Full Name* field.
 - Enter a password for this user.
 - 📁 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-STOCKVM-USERPASS* setting value.

- Click the *Done* button in the upper-left hand corner of the screen.
- NOW WE WAIT - Until CentOS 7 is finished installing.
- When it's finished press the *Reboot* button in the lower-right hand corner of the screen.
 - *Screenshot 19*
 - ⚠ NOTE: Qubes will not actually allow the HVM to reboot, it will just shutdown and disappear. This is OK, we don't want it started just yet...

[Return to TOC](#)

Fix an issue with Qubes and CentOS HVMs

- There is an issue with Qubes and CentOS HVMs which may prevent the CentOS HVM from booting every time.
 - [Issue documentation on GitHub](#)
- The temporary fix is to edit the `/usr/share/qubes/vm-template-hvm.xml` file (in a *Dom0* terminal after becoming *root*) and do the following:
 - Near the bottom, change this line:
 - `<input type='tablet' bus='usb' />`
 - To
 - `<input type='mouse' bus='ps2' />`
- ⚠ NOTE: This file is reported to get replaced on a Qubes update.
 - 🚀 Issue to be followed up [here](#)

[Return to TOC](#)

Install updates

- Start the *tpf-stock_centos_7* HVM
 - This can be done either from a terminal in *dom0* OR, more easily, from the *Qubes VM Manager* in the Qubes GUI.
 - Via terminal:
 - `qvm-start tpf-stock_centos_7`
 - Via *Qubes VM Manager*:
 - Open the *Qubes VM Manager*
 - Right-click on the entry for *tpf-stock_centos_7*
 - Click *Start/Resume VM*
- Log in as the *root* user using the password you set up during the stock CentOS 7 install.
 - 🐭 If you're using the [track0 settings cheat sheet](#), this would be the *TPF-STOCKVM-ROOTPASS* setting value.
- Verify networking works

```
ping -c 4 www.google.com
```

- 🚫 IF NETWORKING IS NOT WORKING, YOU MUST RESOLVE THIS ISSUE BEFORE CONTINUING!
- Install a utility which will decrease the amount of stuff we need to download (*deltarpm*)
 - Doing a system update can mean downloading A LOT of stuff. This utility will help decrease that amount.

```
yum install -y deltarpm
```

- Do a system update

```
yum update -y
```

- *This may take awhile*
- Add the [Extra Packages for Enterprise Linux \(or EPEL\)](#) software repository. This gives us access to

certain additional packages not available in the Fedora stock repo.

```
yum install -y epel-release
```

[Return to TOC](#)

Install additional recommended software by this project

- Install additional packages which provide additional possibly helpful functionality
 - ⚠ NOTE: It may not be 100% necessary to install all of these packages. This is just a list of extra packages which the project author has found helpful from time to time. *It is assumed in all TPF instructions that these packages will be installed. If you choose not to install these packages, you may need to do so manually at some point in the future if functionality is missing!*

```
yum install -y nmap lynx xorg-x11-xauth vim-enhanced bzip2 smartmontools polipo
```

- If you have any other software you personally use, you may install it here. However, please take the following considerations into mind:
 - ⚡ DO NOT INSTALL Apache, mod_ssl, any database (MySQL, MariaDB, etc...) - Doing so will cause problems when we clone this stock VM for other purposes!
 - ⚠ Consider the security ramifications of the particular software you are installing!
 - Examples of things you would add here:
 - Text editors! *vim* & *gvim* are installed via the commands above.

[Return to TOC](#)

Further setup

- FIREWALLD - The minimal version of CentOS does not install firewalld by default!
 - `yum -y install firewalld`
 - `systemctl start firewalld`
 - `systemctl enable firewalld`
- SSH
 - Turn off *root* user access
 - Edit `/etc/ssh/sshd_config`
 - Find this line:
 - `#PermitRootLogin yes`
 - Change it to:
 - `PermitRootLogin no`
 - Start & enable SSHD
 - `systemctl restart sshd`
 - `systemctl enable sshd`
 - Open SSH port in firewall
 - `firewall-cmd --add-service ssh`
 - `firewall-cmd --add-service ssh --permanent`
- Disable other unnecessary services
 - `systemctl stop postfix`
 - `systemctl disable postfix`
- Install & enable [sysstat - Performance monitoring tools for Linux](#)
 - `yum install -y sysstat`
 - `systemctl start sysstat`
 - `systemctl enable sysstat`
- Enable *rc.local* functionality
 - `chmod +x /etc/rc.d/rc.local`
- Edit `/etc/hosts`

- Edit the `/etc/hosts` file

- Add a line like this:

- `10.137.3.9 stock.theplatformframework.com stock`

- ⚠ NOTE! There are *TWO* spaces between `10.137.3.9` and `stock.theplatformframework.com`
- ⚠ NOTE! `stock` is repeated twice. First is for the *Fully Qualified Domain Name* and second is for just the hostname itself.
- Where `10.137.3.9` Is the IP address for this HVM as assigned by Qubes.
 - 🐭 If you're using the [track0 settings cheat sheet](#), this is *TPF-STOCKVM-IP* setting value.
- This command may work:

```
sudo echo `ifconfig | grep -A 1 eth0 | tail -1 | tr -s " " | cut -
```

- Fix an issue with framebuffer driver causing VM delay during boot...

- The following is a fix to remove a VM delay during boot, causing a message like `BUG: soft lockup - CPU#0 stuck for 23s! [systemd-udevd:244]` to occur.
- Edit the file `/etc/default/grub`
 - Find the line which starts:
 - `GRUB_CMDLINE_LINUX=`
 - Remove this text from that line:
 - `rhgb`
 - Add this text to that line:
 - `modprobe.blacklist=bochs_drm consoleblank=0 acpi=off apm=off`

- Run this command:

```
grub2-mkconfig --output=/boot/grub2/grub.cfg
```

[Return to TOC](#)

Wrapping up

- Finally, perform the following actions to ensure we have a good solid stock HVM:
 - Shutdown the HVM
 - Start the HVM
 - Ensure networking is functional

```
ping -c 4 www.google.com
```

[Return to TOC](#)

Enable networking between *tpf-stock_centos_7* and *tpf-work*.

By default, VMs/HVMs in Qubes are not allowed to talk to each other. They can only speak to their *net VM*. In our case this is *tpf-firewall*. So we need to add rules in *tpf-firewall* to allow them to speak to each other. This is quite helpful, as it enables you to do things like SSH from the *tpf-work* VM to an HVM like *tpf-stock_centos_7*. It also enables you to run commands on the HVM which may require an X / GUI. We don't install an X server / GUI in our *TPF* HVMs.

- Start a new terminal in *tpf-firewall*.
- Edit the file `/rw/config/qubes-firewall-user-script`
 - Add the following rules:

- Don't forget to replace the values for TPF-WORKVM-IP and TPF-STOCKVM-IP below!

- 🛠️ If you're using the [track0 settings cheat sheet](#), these are the *TPF-WORKVM-IP* and TPF-STOCKVM-IP setting values.

```
# Allow tpf-work and tpf-stock_centos_7 to talk to each other.
iptables -I FORWARD 2 -s TPF-WORKVM-IP -d TPF-STOCKVM-IP -j ACCEPT
iptables -I FORWARD 2 -s TPF-STOCKVM-IP -d TPF-WORKVM-IP -j ACCEPT
```

- For example, if TPF-WORKVM-IP is `10.137.3.8` and TPF-STOCKVM-IP is `10.137.3.9` then the rules would look like:

```
# Allow tpf-work and tpf-stock_centos_7 to talk to each other.
iptables -I FORWARD 2 -s 10.137.3.8 -d 10.137.3.9 -j ACCEPT
iptables -I FORWARD 2 -s 10.137.3.9 -d 10.137.3.8 -j ACCEPT
```

- By default *qubes-firewall-user-script* will only run when VMs / HVMs change. However, it also need to run on system startup. Therefore we need to run *qubes-firewall-user-script* from *rc.local*.

- Edit `/rw/config/rc.local`, this line:

```
sudo /rw/config/qubes-firewall-user-script
```

- Ensure `/rw/config/rc.local` and `/rw/config/qubes-firewall-user-script` are executable.

```
sudo chmod +x /rw/config/rc.local /rw/config/qubes-firewall-user-script
```

- Now run the `/rw/config/qubes-firewall-user-script` script to enable the rules right now for the current session, as this file only runs on bootstrap.

```
sudo /rw/config/qubes-firewall-user-script
```

- The *tpf-work* VM and *tpf-stock_centos_7* HVM should now to be able to talk to each other.
 - You can now start a terminal in *tpf-work* and SSH to *tpf-stock_centos_7* whenever you wish.
 - The network forwarding rules should also fix themselves automatically if either VM/HVM is restarted.
- Next we want to add a hostname entry in `/etc/hosts` on *tpf-work* for *tpf-stock_centos_7*.
 - From a terminal in *tpf-work* edit the `/etc/hosts` file.

- Add a line like the following:

- Don't forget to replace the value for TPF-STOCKVM-IP below!

- 🛠️ If you're using the [track0 settings cheat sheet](#), this is the `_TPF-STOCKVM-IP` setting value.

```
TPF-STOCKVM-IP stock.theplatformframework.com stock
```

- Now you should be able to do this to SSH to the *tpf-stock_centos_7* HVM:

```
ssh tpf@stock
```

[Return to TOC](#)

Create TPF Proxy VM

⚠️🚧 **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF Proxy VM*), are not yet complete!

[Return to TOC](#)

Clone stock CentOS HVM to new TPF Proxy VM

✳️⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF Proxy VM*), are not yet complete!

- Before we can clone a new HVM from an existing one, we must ensure the one we're copying is not actually running.

- Shutdown the *tpf-stock_centos_7* HVM.
 - This can be done in one of two ways:
 - From a terminal in *Dom0*:

```
qvm-shutdown tpf-stock_centos_7
```

- By right-clicking on the HVM in the *Qubes VM Manager* and selecting *Shutdown VM*.

- From a terminal in *Dom0*:
 - Clone the new HVM named *tpf-proxy* from *tpf-stock_centos_7*

```
qvm-clone tpf-stock_centos_7 tpf-proxy
```

- Set the HVM's firewall to *tpf-firewall* and set the HVM label.

```
qvm-prefs -s tpf-proxy netvm tpf-firewall
qvm-prefs -s tpf-proxy label gray
```

- Get the IP address for the new VM:

```
qvm-ls -n tpf-proxy
```

- The IP address is in the *ip* column
 - 📌 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-PROXYVM-IP* setting value.
- Start the HVM manually (unless you are going to reboot)

```
qvm-start tpf-proxy
```

[Return to TOC](#)

Configure networking / hostname / hosts

✳️⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF Proxy VM*), are not yet complete!

- From within the HVM, log in as the *root* user and perform the following operations.
 - Change the user passwords
 - Because we just cloned the *stock CentOS 7 HVM*, The *root* and *tpf* user passwords are the same.
 - 📌 If you're using the [track0 settings cheat sheet](#), these are the values for the *TPF-STOCKVM-ROOTPASS* and *TPF-STOCKVM-TPFPASS* settings.
 - Change these passwords:

```
passwd root
```

- ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
- 📋 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-PROXYVM-ROOTPASS* setting value.

```
passwd tpf
```

- ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
- 📋 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-PROXYVM-TPFPASS* setting value.
- Change the hostname
 - Run the `nmtui` command. Choose the option to *Set system hostname*.
 - Set the hostname to `proxy.theplatformframework.com`
- Change the IP address
 - Run the `nmtui` command again. Choose the option to *Edit a connection*.
 - Choose the connection `eth0` and then *Edit...*
 - Change the IP address to the address we found above.
 - In *IPv4 CONFIGURATION* in *Addresses*
 - 📋 If you're using the [track0 settings cheat sheet](#), use the value for the *TPF-PROXYVM-IP* setting.
 - ⚠️ *Ensure to append /32 to the end of the IP Address!*
 - ⚠️ NOT /24 , otherwise Qubes networking will have issues!
 - The other information should remain the same.
- Update the `/etc/hosts` file
 - Edit the `/etc/hosts` file
 - Change the line for `stock.theplatformframework.com`
 - Change the hostname from `stock` to `proxy` (should be done twice on that line, for each instance of the hostname `stock`).
 - Change the IP address to what we just set up.
- Now reboot the proxy HVM.
 - ⚠️ Qubes doesn't actually allow VMs to be automatically rebooted, so just shut it down, then you'll have to start it again manually.

[Return to TOC](#)

Enable networking between *tpf-proxy* and *tpf-work*.

Just like we did previously with *tpf-stock_centos_7* we're going to enable *tpf-work* to talk to *tpf-proxy*.

- Start a new terminal in *tpf-firewall*.
- Edit the file `/rw/config/qubes-firewall-user-script`
 - Add the following rules:
 - Don't forget to replace the values for *TPF-WORKVM-IP* and *TPF-STOCKVM-IP* below!
 - 📋 If you're using the [track0 settings cheat sheet](#), these are the *TPF-WORKVM-IP* and *TPF-STOCKVM-IP* setting values.

```
# Allow tpf-work and tpf-proxy to talk to each other.
iptables -I FORWARD 2 -s TPF-WORKVM-IP -d TPF-PROXY-IP -j ACCEPT
iptables -I FORWARD 2 -s TPF-PROXY-IP -d TPF-WORKVM-IP -j ACCEPT
```

- Now run the `/rw/config/qubes-firewall-user-script` script to enable the rules right now for the current session, as this file only runs on bootup.

```
sudo /rw/config/qubes-firewall-user-script
```

- The *tpf-work* VM and *tpf-proxy* HVM should now be able to talk to each other.

- Next we want to add a hostname entry in `/etc/hosts` on *tpf-work* for *tpf-proxy*.
 - From a terminal in *tpf-work* edit the `/etc/hosts` file.
 - Add a line like the following:
 - Don't forget to replace the value for TPF-PROXYVM-IP below!
 - 📌 If you're using the [track0 settings cheat sheet](#), this is the `_TPF-PROXYVM-IP` setting value.

```
TPF-PROXYVM-IP proxy.theplatformframework.com proxy
```

[Return to TOC](#)

Set up ModSecurity

⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF Proxy VM*), are not yet complete!

- Make sure the system is updated

```
yum -y update
```

- Make sure Apache is installed and enabled on 443 (HTTPS)

```
yum -y install httpd mod_ssl
systemctl start httpd
systemctl enable httpd
firewall-cmd --add-service https
firewall-cmd --add-service https --permanent
```

- Install `mod_security` and `mod_evasive`

```
yum -y install mod_security mod_evasive
```

- Edit `/etc/httpd/conf.d/mod_security.conf`

- Add this at the top of the file:

```
LoadModule security2_module modules/mod_security2.so
```

- Add *OWASP Core Rule Set* for `mod_security`

```
cd /etc/httpd
mkdir crs
cd crs
wget https://github.com/SpiderLabs/owasp-modsecurity-crs/tarball/master
tar xzf master
mv SpiderLabs-owasp-modsecurity-crs-* owasp-modsecurity-crs
cd owasp-modsecurity-crs
cp modsecurity_crs_10_setup.conf.example modsecurity_crs_10_setup.conf
```

- Disable the `OWASP_CRS/PROTOCOL_VIOLATION/IP_HOST` rule (#960017) which prohibits accessing a server by it's IP...
 - Edit `/etc/httpd/crs/owasp-modsecurity-crs/base_rules/modsecurity_crs_21_protocol_anomalies.conf`
 - Search for `960017 (OWASP_CRS/PROTOCOL_VIOLATION/IP_HOST)` and comment it out.
- Edit `/etc/httpd/conf/httpd.conf`:
 - Add these lines at the end of the file:

```
<IfModule security2_module>
    Include crs/owasp-modsecurity-crs/modsecurity_crs_10_setup.conf
    Include crs/owasp-modsecurity-crs/base_rules/*.conf
</IfModule>
```

- Restart Apache

```
systemctl restart httpd
```

[Return to TOC](#)

Set up mod_proxy

★⚠️⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF Proxy VM*), are not yet complete!

- Install *mod_ssl* & *mod_proxy_html*

```
yum install -y mod_ssl mod_proxy_html
```

- Create a file named */etc/httpd/conf.d/reverse-proxy.conf* and add the following contents:
 - ⚠️ **NOTE:** The *SSLProxyVerify* & *SSLProxyCheck* items are necessary if the proxy does not recognize the SSL cert of the target machine!
 - ⚠️ **NOTE:** The *ProxyPass* and *ProxyPassReverse* entries will be set up later on in this install guide.

```
ProxyRequests Off
SSLProxyEngine On
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
#ProxyPass / https://TARGETSERVER/
#ProxyPassReverse / https://TARGETSERVER/
```

- Copy template file */usr/share/doc/httpd-2.4.6/proxy-html.conf* to */etc/httpd/conf.d/*

```
cp /usr/share/doc/httpd-2.4.6/proxy-html.conf /etc/httpd/conf.d/
```

- Restart Apache

```
systemctl restart httpd
```

[Return to TOC](#)

Enable VM autostart on boot and crash

Because this VM is custom we need a custom way to have it automatically start on boot, restart automatically on failure/crash, and shutdown properly. We can achieve this via *systemd*.

Unfortunately, for several reasons related to how services/sessions in Qubes work, we need to create separate systemd service files for startup and shutdown in order to ensure reliable operation. In particular we need to create a systemd timer to ensure the HVM does not start until the GUI is fully up and running.

- In a terminal in *dom0*:
 - Create a file named */usr/lib/systemd/system/tpf-proxy-hvm-startup.service* with the following contents:

```
[Unit]
Description=Start the tpf-proxy HVM, restart on HVM crash/shutdown.
```

```
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
Requires=user-1000.slice lightdm.service

[Service]
# This is a forking type process, NOT oneshot NOR simple.
Type=forking
# We need to set a DISPLAY or the HVM won't show up on the GUI.
Environment=DISPLAY=:0
ExecStart=/bin/qvm-start tpf-proxy
# We need to specify a dummy handler for ExecStop, as we handle HVM shutdown in
# a systemd user service, so we do not want this service to potentially mess
# with that.
ExecStop=/bin/true
# We want this service to monitor the HVM and restart it if it crashes or shut
# down.
Restart=always
# We need a high value here to ensure the HVM doesn't attempt to restart durin
# system shutdown.
RestartSec=120
# We are monitoring the PID of the following process/file, NOT the status of
# the qvm-start process itself.
PIDFile=/var/run/qubes/qubesdb.tpf-proxy.pid

[Install]
WantedBy=graphical.target
```

- o Create a file named `/usr/lib/systemd/system/tpf-proxy-hvm-startup.timer` with the following contents:

```
[Unit]
Description=Start the tpf-proxy HVM with a timer, ensuring it starts when the
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
Requires=user-1000.slice lightdm.service

[Timer]
# Let's give the GUI 20 seconds to start and settle before we attempt to start
# our HVM.
OnActiveSec=20
# This is the HVM systemd service we want to start.
Unit=tpf-proxy-hvm-startup.service

[Install]
WantedBy=graphical.target
```

- o Create a file named `/usr/lib/systemd/user/tpf-proxy-hvm-shutdown.service` with the following contents:

```
[Unit]
# This script ensures that the HVM will be stopped at user logout/shutdown,
# otherwise the system will hang on shutdown/reboot.
Description=Stop the tpf-proxy VM on user logout/system shutdown.
Before=systemd-exit.service

[Service]
# This is a oneshot type of service, as we need it to pay attention to our
# "Before" requirement in the above "Unit" section. A simple type of service
# will not do so.
Type=oneshot
# We need a dummy program to run at startup, as we're only actually focused on
# shutdown.
ExecStart=/bin/true
# In order for the service to "stay running" we need to set "RemainAfterExit".
RemainAfterExit=yes
```

```
# This is what actually shuts down the HVM.
ExecStop=/bin/sudo /bin/qvm-shutdown tpf-proxy --force --wait

[Install]
WantedBy=default.target
```

- Then enable these service using this command:

- ⚠️ **IMPORTANT NOTE!** These commands should be executed in the *dom0* terminal when NOT in sudo nor su mode (i.e., not as root or a user with elevated privileges)! This is because the second command needs to run as the *tpfadmin* user itself.

```
systemctl enable tpf-proxy-hvm-startup.timer
systemctl --user enable tpf-proxy-hvm-shutdown
systemctl --user start tpf-proxy-hvm-shutdown
```

[Return to TOC](#)

Create TPF FreeIPA VM

⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF FreeIPA VM*), are not yet complete!

[Return to TOC](#)

Clone stock CentOS HVM to new TPF FreeIPA VM

⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF FreeIPA VM*), are not yet complete!

- From a terminal in *Dom0*:
 - Clone the new HVM named *tpf-ipa* from *tpf-stock_centos_7*

```
qvm-clone tpf-stock_centos_7 tpf-ipa
```

- Set the HVM's firewall to *tpf-firewall* and set the HVM label.

```
qvm-prefs -s tpf-ipa netvm tpf-firewall
qvm-prefs -s tpf-ipa label gray
```

- Get the IP address for the new VM:

```
qvm-ls -n tpf-ipa
```

- The IP address is in the *ip* column
 - 📌 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-IP* setting value.

- Start the HVM manually (unless you are going to reboot)

```
qvm-start tpf-ipa
```

[Return to TOC](#)

Configure networking / hostname / hosts

⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF FreeIPA VM*), are not yet complete!

- From within the HVM, log in as the *root* user and perform the following operations.
 - Change the user passwords

- Because we just cloned the *stock CentOS 7 HVM*, The *root* and *tpf* user passwords are the same as those used in that HVM (*stock CentOS 7 HVM*).
 - 🛠️ If you're using the [track0 settings cheat sheet](#), these are the values for the *TPF-STOCKVM-ROOTPASS* and *TPF-STOCKVM-TPFPASS* settings.
- Change these passwords:
 - `passwd root`
 - ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-ROOTPASS* setting value.
 - `passwd tpf`
 - ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-TPFPASS* setting value.
- Change the hostname
 - Run the `nmtui` command. Choose the option to *Set system hostname*.
 - Set the hostname to `ipa.theplatformframework.com`
- Change the IP address
 - Run the `nmtui` command again. Choose the option to *Edit a connection*.
 - Choose the connection `eth0` and then *Edit...*
 - Change the IP address to the address we found above.
 - In *IPv4 CONFIGURATION* in *Addresses*
 - 🛠️ If you're using the [track0 settings cheat sheet](#), use the value for the *TPF-IPAVM-IP* setting.
 - ⚠️ *Ensure to append /32 to the end of the IP Address!*
 - ⚠️ NOT /24 , otherwise Qubes networking will have issues!
 - The other information should remain the same.
- Update the `/etc/hosts` file
 - Edit the `/etc/hosts` file
 - Change the line for `stock.theplatformframework.com`
 - Change the hostname from `stock` to `ipa` (should be done twice on that line, for each instance of the hostname `stock`).
 - Change the IP address to what we just set up.
- Now reboot the FreeIPA HVM.
 - ⚠️ Qubes doesn't actually allow VMs to be automatically rebooted, so just shut it down, then you'll have to start it again manually.

[Return to TOC](#)

Set up FreeIPA

🛠️⚠️ **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF FreeIPA VM*), are not yet complete!

- Install freeipa-server packages:
 - This will install a TON of stuff!

```
yum install -y ipa-server ipa-server-dns bind-dyndb-ldap
```

- Configure FreeIPA!

```
ipa-server-install --setup-dns
```

- ⚠️ **IMPORTANT NOTE:** If this process fails at any point, you must UNINSTALL and then try the process again.
 - To uninstall:

```
ipa-server-install --uninstall
```

- If this fails also, try to correct whatever the error is and run the UNINSTALL process again.
- First thing it asks is for server host name - If the local hostname has been set up correctly in */etc/hostname* & */etc/hosts*, this should be automatic and you should only have to press ENTER to accept the default
 - Otherwise this must be the Fully Qualified Domain Name (FQDN)
 - i.e.: ipa.theplatformframework.com
- Confirm the domain name - Again, if the local hostname has been set up correctly this should automatically show.
 - Otherwise this must ONLY be the domain name
 - i.e.: theplatformframework.com
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-IPADOMAIN* setting value.
- Realm name (Kerberos) - Again, if the local hostname has been set up correctly this should automatically show.
 - Otherwise this must be the domain name as entered above, EXCEPT IN ALL UPPER-CASE.
 - i.e.: THEPLATFORMFRAMEWORK.COM
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-IPAREALM* setting value.
- Create a password for the Directory Manager. This is not as commonly used as the next item, the admin user. However, it's very important this is a secure password!
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-IPADIRMANPASS* setting value.
- Create a password for the admin user (IPA admin). This is the password you will use for most administrative tasks. It MUST be a secure password!
 - 🛠️ If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-IPAVM-IPAADMINPASS* setting value.
- Answer the following questions as directed below:
 - Existing BIND confirmation detected, overwrite? [no]: yes
 - Do you want to configure DNS forwarders? [yes]: PRESS ENTER
 - Enter an IP address for a DNS forwarder, or press Enter to skip: 8.8.8.8
 - Enter an IP address for a DNS forwarder, or press Enter to skip: 8.8.4.4
 - Enter an IP address for a DNS forwarder, or press Enter to skip: PRESS ENTER
 - Do you want to configure the reverse zone? [yes]: PRESS ENTER
 - Please specify the reverse zone name [...SOMETHING HERE...]: PRESS ENTER
 - Continue to configure the system with these values? [no]: yes
 - IMPORTANT NOTE: If the following question is asked:
 - 🛠️⚠️ *Need exact question wording here*
 - 🛠️ Something about *cannot use IP network address...*
 - Then the following needs done:
 - Edit */usr/lib/python2.7/site-packages/ipapython/ipautil.py*
 - Comment out the 4 lines (around 177) which look like (or similar to) (add the THREE double-quotes as noted below):

```
"""
if not allow_network and addr == net.network:
    raise ValueError("cannot use IP network address")
if not allow_broadcast and addr.version == 4 and addr == net:
    raise ValueError("cannot use broadcast IP address")
"""
```


- From: <https://www.redhat.com/archives/freeipa-users/2012-February/msg00064.html>

- NOW WAIT! This part may take a *long*, long, time! (30 minutes or more)

- Open necessary ports on firewall:

```
firewall-cmd --permanent --add-service=ntp
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --permanent --add-service=ldap
firewall-cmd --permanent --add-service=ldaps
firewall-cmd --permanent --add-service=kerberos
firewall-cmd --permanent --add-service=kpasswd
```

- Use authconfig to ensure home directories are created and enable the *sssd* service.

```
authconfig --enablemkhomedir --update
systemctl enable sssd
```

- Now reboot the FreeIPA HVM.

- ⚠ Qubes doesn't actually allow VMs to be automatically rebooted, so just shut it down, then you'll have to start it again manually.

- TEST IT!

- Using a browser, try to access the FreeIPA server.
 - 🚀 This needs to be changed for Qubes, since it will not be accessible via the outside world...
- i.e.: <https://ipa.theplatformframework.com/ipa/ui/>
- ⚠ NOTE! You must have the IP of the FreeIPA server in your hosts file (or truly set up correctly in DNS)
 - This includes in Windows too!!!
 - Windows hosts file (NEED ADMIN PRIVS - START NOTEPAD AS ADMIN!):
 - `C:\Windows\System32\drivers\etc\hosts`
- ⚠ NOTE! Chrome may present an Apache-like username/password box. Entering the admin user/pass here does not seem to work, but if you CANCEL this, you will be sent to the IPA admin website.

[Return to TOC](#)

Enable VM autostart on boot and crash

Because this VM is custom we need a custom way to have it automatically start on boot, restart automatically on failure/crash, and shutdown properly. We can achieve this via *systemd*.

Unfortunately, for several reasons related to how services/sessions in Qubes work, we need to create separate *systemd* service files for startup and shutdown in order to ensure reliable operation. In particular we need to create a *systemd* timer to ensure the HVM does not start until the GUI is fully up and running.

- In a terminal in *dom0*:

- Create a file named `/usr/lib/systemd/system/tpf-ipa-hvm-startup.service` with the following contents:

```
[Unit]
Description=Start the tpf-ipa HVM, restart on HVM crash/shutdown.
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
```

```

Requires=user-1000.slice lightdm.service

[Service]
# This is a forking type process, NOT oneshot NOR simple.
Type=forking
# We need to set a DISPLAY or the HVM won't show up on the GUI.
Environment=DISPLAY=:0
ExecStart=/bin/qvm-start tpf-ipa
# We need to specify a dummy handler for ExecStop, as we handle HVM shutdown i
# a systemd user service, so we do not want this service to potentially mess
# with that.
ExecStop=/bin/true
# We want this service to monitor the HVM and restart it if it crashes or shut
# down.
Restart=always
# We need a high value here to ensure the HVM doesn't attempt to restart durin
# system shutdown.
RestartSec=120
# We are monitoring the PID of the following process/file, NOT the status of
# the qvm-start process itself.
PIDFile=/var/run/qubes/qubesdb.tpf-ipa.pid

[Install]
WantedBy=graphical.target

```

- Create a file named `/usr/lib/systemd/system/tpf-ipa-hvm-startup.timer` with the following contents:

```

[Unit]
Description=Start the tpf-ipa HVM with a timer, ensuring it starts when the GUI
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
Requires=user-1000.slice lightdm.service

[Timer]
# Let's give the GUI 20 seconds to start and settle before we attempt to start
# our HVM.
OnActiveSec=20
# This is the HVM systemd service we want to start.
Unit=tpf-ipa-hvm-startup.service

[Install]
WantedBy=graphical.target

```

- Create a file named `/usr/lib/systemd/user/tpf-ipa-hvm-shutdown.service` with the following contents:

```

[Unit]
# This script ensures that the HVM will be stopped at user logout/shutdown,
# otherwise the system will hang on shutdown/reboot.
Description=Stop the tpf-ipa VM on user logout/system shutdown.
Before=systemd-exit.service


[Service]
# This is a oneshot type of service, as we need it to pay attention to our
# "Before" requirement in the above "Unit" section. A simple type of service
# will not do so.
Type=oneshot
# We need a dummy program to run at startup, as we're only actually focused on
# shutdown.
ExecStart=/bin/true
# In order for the service to "stay running" we need to set "RemainAfterExit".
RemainAfterExit=yes
# This is what actually shuts down the HVM.
ExecStop=/bin/sudo /bin/qvm-shutdown tpf-ipa --force --wait

[Install]

```

```
WantedBy=default.target
```


- Then enable these service using this command:

-  **IMPORTANT NOTE!** These commands should be executed in the *dom0* terminal when NOT in sudo nor su mode (*i.e., not as root or a user with elevated privileges*)! This is because the second command needs to run as the *tpfadmin* user itself.

```
systemctl enable tpf-ipa-hvm-startup.timer
systemctl --user enable tpf-ipa-hvm-shutdown
systemctl --user start tpf-ipa-hvm-shutdown
```


[Return to TOC](#)

Create TPF TestProject VM

 **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF TestProject VM*), are not yet complete!

[Return to TOC](#)

Clone stock CentOS HVM to new TPF TestProject VM

 **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF TestProject VM*), are not yet complete!

- From a terminal in *Dom0*:
 - Clone the new HVM named *tpf-test* from *tpf-stock_centos_7*


```
qvm-clone tpf-stock_centos_7 tpf-test
```

- Set the HVM's firewall to *tpf-firewall* and set the HVM label.

```
qvm-prefs -s tpf-test netvm tpf-firewall
qvm-prefs -s tpf-test label gray
```

- Get the IP address for the new VM:


```
qvm-ls -n tpf-test
```

- The IP address is in the *ip* column
 -  If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-TESTVM-IP* setting value.
- Start the HVM manually (unless you are going to reboot)

```
qvm-start tpf-test
```

[Return to TOC](#)

Configure networking / hostname / hosts

 **IMPORTANT NOTE!** This entire section, and all sub-sections (for *TPF TestProject VM*), are not yet complete!

- From within the HVM, log in as the *root* user and perform the following operations.
 - Change the user passwords
 - Because we just cloned the *stock CentOS 7 HVM*, The *root* and *tpf* user passwords are the same as those used in that HVM (*stock CentOS 7 HVM*).

- 📖 If you're using the [track0 settings cheat sheet](#), these are the values for the *TPF-STOCKVM-ROOTPASS* and *TPF-STOCKVM-TPFPASS* settings.
- Change these passwords:
 - `passwd root`
 - ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
 - 📖 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-TESTVM-ROOTPASS* setting value.
 - `passwd tpf`
 - ⚠️ **IMPORTANT NOTE!** DO NOT USE THE SAME PASSWORDS! Each and every VM password should be unique. Use established best practices here.
 - 📖 If you're using the [track0 settings cheat sheet](#), write this down for the *TPF-TESTVM-TPFPASS* setting value.
- Change the hostname
 - Run the `nmtui` command. Choose the option to *Set system hostname*.
 - Set the hostname to *test.theplatformframework.com*
- Change the IP address
 - Run the `nmtui` command again. Choose the option to *Edit a connection*.
 - Choose the connection *eth0* and then *Edit...*
 - Change the IP address to the address we found above.
 - In *IPv4 CONFIGURATION* in *Addresses*
 - 📖 If you're using the [track0 settings cheat sheet](#), use the value for the *TPF-TESTVM-IP* setting.
 - ⚠️ *Ensure to append /32 to the end of the IP Address!*
 - ⚠️ NOT /24 , otherwise Qubes networking will have issues!
 - The other information should remain the same.
- Update the */etc/hosts* file
 - Edit the */etc/hosts* file
 - Change the line for *stock.theplatformframework.com*
 - Change the hostname from *stock* to *test* (should be done twice on that line, for each instance of the hostname *stock*).
 - Change the IP address to what we just set up.
 - Also add a new line for *proxy.theplatformframework.com*
 - 📖 If you're using the [track0 settings cheat sheet](#), use the value for the *TPF-PROXYVM-IP* setting.

```
TPF-PROXYVM-IP    proxy.theplatformframework.com proxy
```

- Now reboot the TestProject HVM.
 - ⚠️ Qubes doesn't actually allow VMs to be automatically rebooted, so just shut it down, then you'll have to start it again manually.

Return to TOC

Enable networking between *tpf-test*, *tpf-work*, *tpf-proxy* and *tpf-ipa*.

Just like we did previously with *tpf-stock_centos_7* we're going to enable *tpf-work*, *tpf-proxy*, and *tpf-ipa* to talk to *tpf-test*.

- Start a new terminal in *tpf-firewall*.
- Edit the file */rw/config/qubes-firewall-user-script*
 - Add the following rules:
 - Don't forget to replace the values for *TPF-WORKVM-IP*, *TPF-PROXYVM-IP*, *TPF-IPAVM-IP*, and *TPF-TESTVM-IP* below!

- 📖 If you're using the [track0 settings cheat sheet](#), these are the *TPF-WORKVM-IP*, *TPF-PROXYVM-IP*, *TPF-IPAVM-IP* AND *TPF-TESTVM-IP* setting values.

```
# Allow tpf-work and tpf-test to talk to each other.
iptables -I FORWARD 2 -s TPF-WORKVM-IP -d TPF-TESTVM-IP -j ACCEPT
iptables -I FORWARD 2 -s TPF-TESTVM-IP -d TPF-WORKVM-IP -j ACCEPT
# Allow tpf-proxy and tpf-test to talk to each other.
iptables -I FORWARD 2 -s TPF-PROXYVM-IP -d TPF-TESTVM-IP -j ACCEPT
iptables -I FORWARD 2 -s TPF-TESTVM-IP -d TPF-PROXYVM-IP -j ACCEPT
# Allow tpf-ipa and tpf-test to talk to each other.
iptables -I FORWARD 2 -s TPF-IPAVM-IP -d TPF-TESTVM-IP -j ACCEPT
iptables -I FORWARD 2 -s TPF-TESTVM-IP -d TPF-IPAVM-IP -j ACCEPT
```

- Now run the */rw/config/qubes-firewall-user-script* script to enable the rules right now for the current session, as this file only runs on bootup.

```
sudo /rw/config/qubes-firewall-user-script
```

- The *tpf-work* VM and *tpf-ipa*, *tpf-proxy*, and *tpf-test* HVMs should now be able to talk to each other.
- Next we want to add a hostname entry in */etc/hosts* on *tpf-work* for *tpf-test*.

- From a terminal in *tpf-work* edit the */etc/hosts* file.

- Add a line like the following:

- Don't forget to replace the value for TPF-TESTVM-IP below!

- 📖 If you're using the [track0 settings cheat sheet](#), this is the *_TPF-TESTVM-IP* setting value.

```
TPF-TESTVM-IP test.theplatformframework.com test
```

- Now we need to do the same thing in */etc/hosts* on *tpf-proxy* for *tpf-test*.

- From a terminal in *tpf-proxy* edit the */etc/hosts* file.

- Add a line like the following:

- Don't forget to replace the value for TPF-TESTVM-IP below!

- 📖 If you're using the [track0 settings cheat sheet](#), this is the *_TPF-TESTVM-IP* setting value.

```
TPF-TESTVM-IP test.theplatformframework.com test
```

- Now we need to do the same thing in */etc/hosts* on *tpf-ipa* for *tpf-test*.

- From a terminal in *tpf-ipa* edit the */etc/hosts* file.

- Add a line like the following:

- Don't forget to replace the value for TPF-TESTVM-IP below!

- 📖 If you're using the [track0 settings cheat sheet](#), this is the *_TPF-TESTVM-IP* setting value.

```
TPF-TESTVM-IP test.theplatformframework.com test
```

[Return to TOC](#)

Enable VM autostart on boot and crash

Because this VM is custom we need a custom way to have it automatically start on boot, restart automatically on failure/crash, and shutdown properly. We can achieve this via *systemd*.

Unfortunately, for several reasons related to how services/sessions in Qubes work, we need to create

separate systemd service files for startup and shutdown in order to ensure reliable operation. In particular we need to create a systemd timer to ensure the HVM does not start until the GUI is fully up and running.

- In a terminal in *dom0*:
 - Create a file named `/usr/lib/systemd/system/tpf-test-hvm-startup.service` with the following contents:

```
[Unit]
Description=Start the tpf-test HVM, restart on HVM crash/shutdown.
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
Requires=user-1000.slice lightdm.service

[Service]
# This is a forking type process, NOT oneshot NOR simple.
Type=forking
# We need to set a DISPLAY or the HVM won't show up on the GUI.
Environment=DISPLAY=:0
ExecStart=/bin/qvm-start tpf-test
# We need to specify a dummy handler for ExecStop, as we handle HVM shutdown i
# a systemd user service, so we do not want this service to potentially mess
# with that.
ExecStop=/bin/true
# We want this service to monitor the HVM and restart it if it crashes or shut
# down.
Restart=always
# We need a high value here to ensure the HVM doesn't attempt to restart durin
# system shutdown.
RestartSec=120
# We are monitoring the PID of the following process/file, NOT the status of
# the qvm-start process itself.
PIDFile=/var/run/qubes/qubesdb.tpf-test.pid

[Install]
WantedBy=graphical.target
```

- Create a file named `/usr/lib/systemd/system/tpf-test-hvm-startup.timer` with the following contents:

```
[Unit]
Description=Start the tpf-test HVM with a timer, ensuring it starts when the GUI
# We require (and start after) user-1000.slice because that's the user
# (tpfadmin) that is set for autologin. We also want to make sure the
# lightdm.service is running.
After=user-1000.slice lightdm.service
Requires=user-1000.slice lightdm.service

[Timer]
# Let's give the GUI 20 seconds to start and settle before we attempt to start
# our HVM.
OnActiveSec=20
# This is the HVM systemd service we want to start.
Unit=tpf-test-hvm-startup.service

[Install]
WantedBy=graphical.target
```

- Create a file named `/usr/lib/systemd/user/tpf-test-hvm-shutdown.service` with the following contents:

```
[Unit]
# This script ensures that the HVM will be stopped at user logout/shutdown,
# otherwise the system will hang on shutdown/reboot.
```

```

Description=Stop the tpf-test VM on user logout/system shutdown.
Before=systemd-exit.service

[Service]
# This is a oneshot type of service, as we need it to pay attention to our
# "Before" requirement in the above "Unit" section. A simple type of service
# will not do so.
Type=oneshot
# We need a dummy program to run at startup, as we're only actually focused on
# shutdown.
ExecStart=/bin/true
# In order for the service to "stay running" we need to set "RemainAfterExit".
RemainAfterExit=yes
# This is what actually shuts down the HVM.
ExecStop=/bin/sudo /bin/qvm-shutdown tpf-test --force --wait

[Install]
WantedBy=default.target

```

o Then enable these service using this command:

- **⚠ IMPORTANT NOTE!** These commands should be executed in the *dom0* terminal when NOT in sudo nor su mode (*i.e., not as root or a user with elevated privileges*)! This is because the second command needs to run as the *tpfadmin* user itself.

```

systemctl enable tpf-test-hvm-startup.timer
systemctl --user enable tpf-test-hvm-shutdown
systemctl --user start tpf-test-hvm-shutdown

```

[Return to TOC](#)

Create test web app

Now, finally, we're might get to do something fun. We're going to tie everything together and create a sample app which ties into FreeIPA and can be accessed via the proxy.

[Return to TOC](#)

Install Apache & PHP

- Open a terminal in the *tpf-test* VM.
- Make sure the system is updated

```
yum -y update
```

- Make sure Apache is installed and enabled on 443 (HTTPS)

```

yum -y install httpd mod_ssl
systemctl start httpd
systemctl enable httpd
firewall-cmd --add-service https
firewall-cmd --add-service https --permanent

```

- Install PHP5.6 for CentOS 7
 - o This requires the addition of the [IUS Repo](#)

```

yum -y install https://centos7.iuscommunity.org/ius-release.rpm
yum -y install php56u php56u-cli php56u-json

```

- Restart the Apache web server.

```
systemctl restart httpd
```

TO BE CONTINUED...

[Return to TOC](#)

Other / Optional VMs

- [LDAP/Red Hat 389 Directory Server](#)

[Return to TOC](#)

