

Tutorial: CANVAS 101 Part 1 (host selection, launching modules)

I've gotten a bit of feedback from some of our CANVAS users asking for a reference on basic CANVAS usage, well their wish has been granted! The next (most likely three) tutorials will focus on the basics of using CANVAS and hopefully serve as a reference for folks who don't use CANVAS every day.

At the end of this tutorial you will be able to

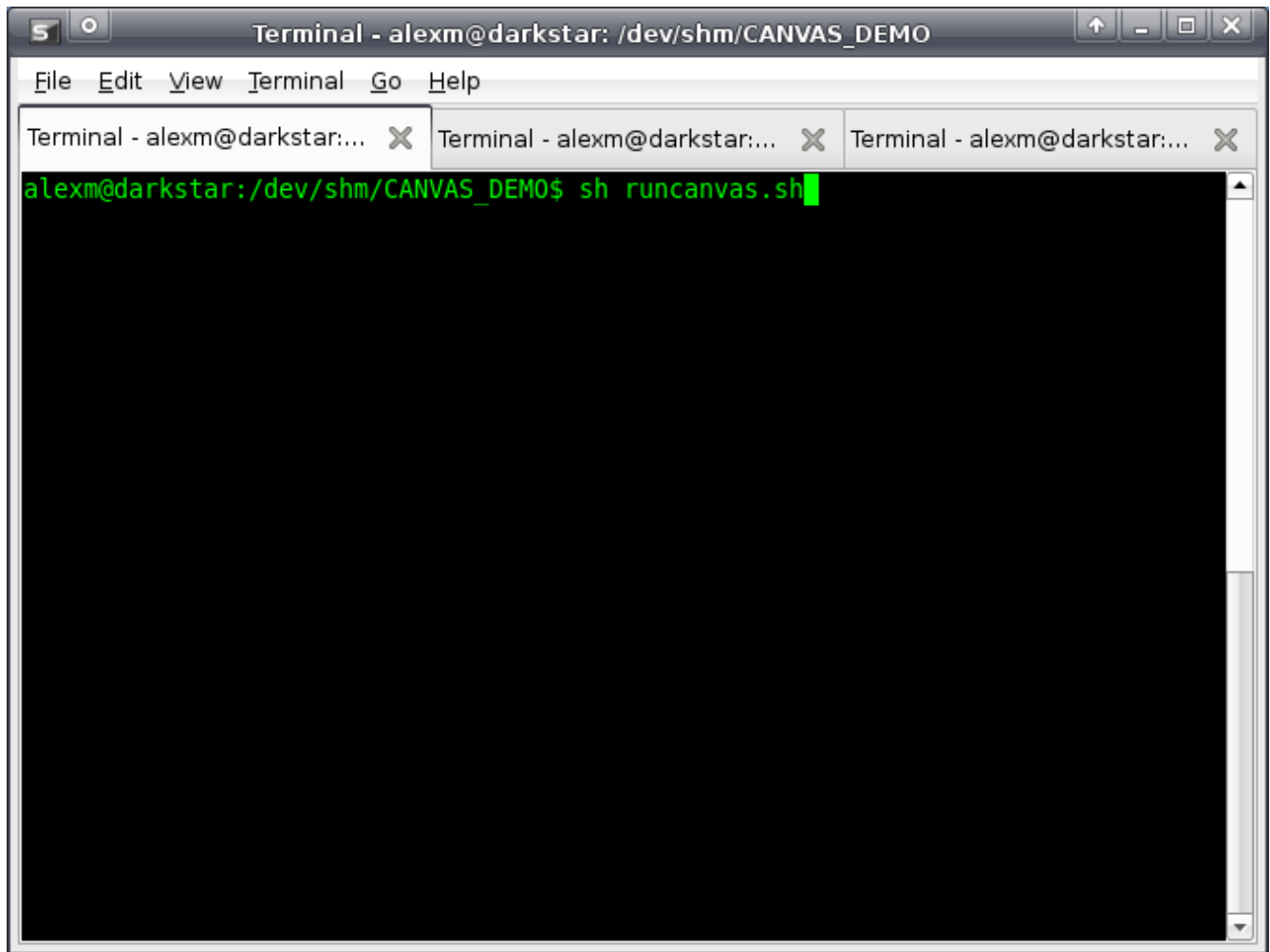
- Start CANVAS
- Understand the GUI organization
- Select hosts
- Launch modules

Introduction

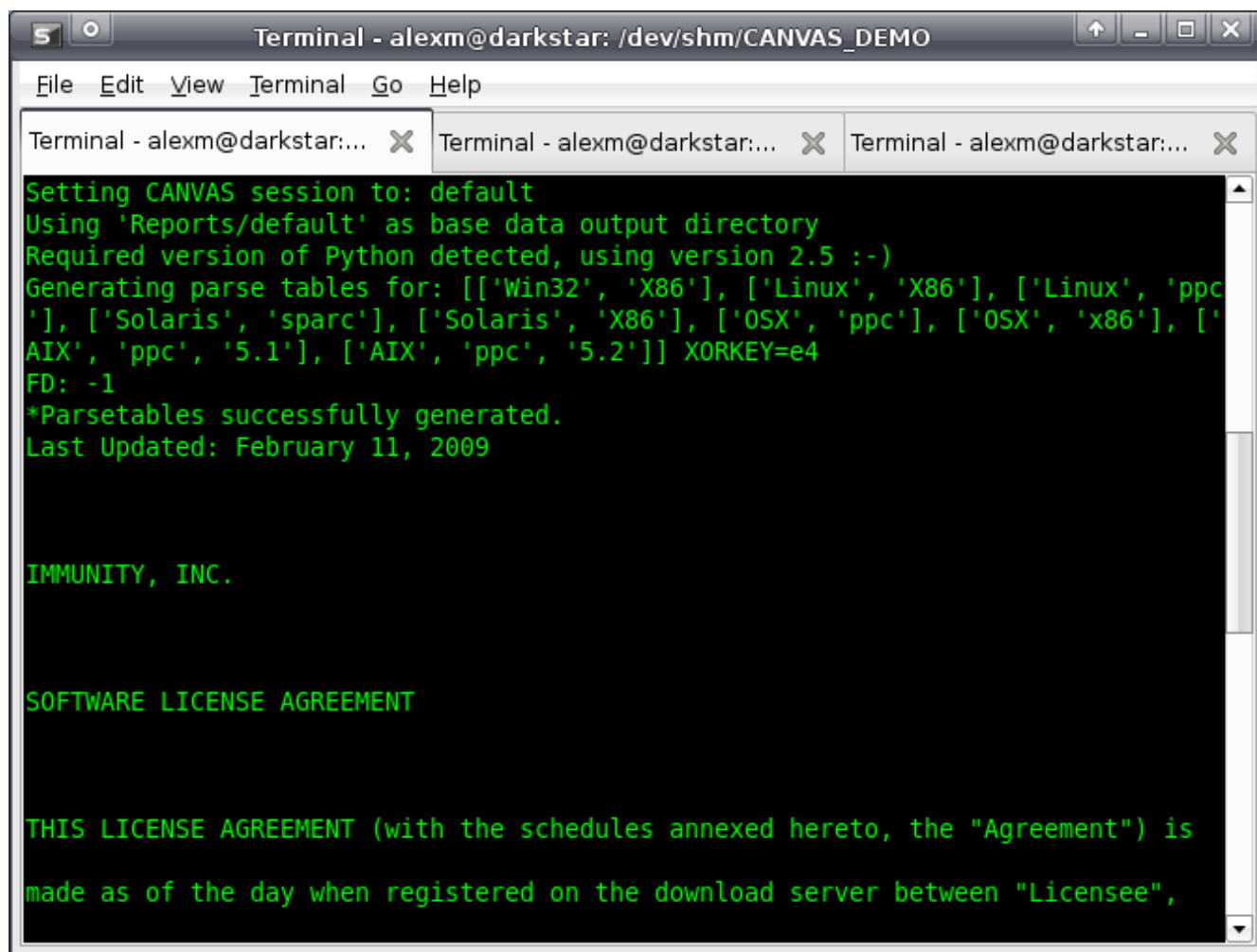
If you've ever sat in on a demo with me or chatted me up about CANVAS I try to always make the point that you should be running CANVAS on Linux. In the last few years I don't think I've ever been to one organization that doesn't have some kind of virtualization solution, be it VMWare, VirtualServer, Xen, etc. As security professionals creating VMs and having a working knowledge of multiple OSES is a required part of our skill set now. So if you're constrained to Windows by executive decree or even if Windows is your preference, roll yourself a Linux VM or download VMWare player and a Linux based appliance to run CANVAS off of.

Starting CANVAS

- 1) Browse to your CANVAS directory (generally CANVAS_YourCompanyName)
- 2) On the Linux commandline you can type: *sh runcanvas.sh* or *python runcanvas.py*
- 3) On Windows you can simply double click on *canvas.bat* through the GUI



If it's your first time running this version of CANVAS you'll see CANVAS take some steps to generate data it'll continually reference as well as the license agreement. You'll have to hit enter a few times to scroll through the entire license agreement then you'll be prompted to accept it.

A terminal window titled "Terminal - alexm@darkstar: /dev/shm/CANVAS_DEMO" with a menu bar (File, Edit, View, Terminal, Go, Help) and three tabs. The terminal output is as follows:

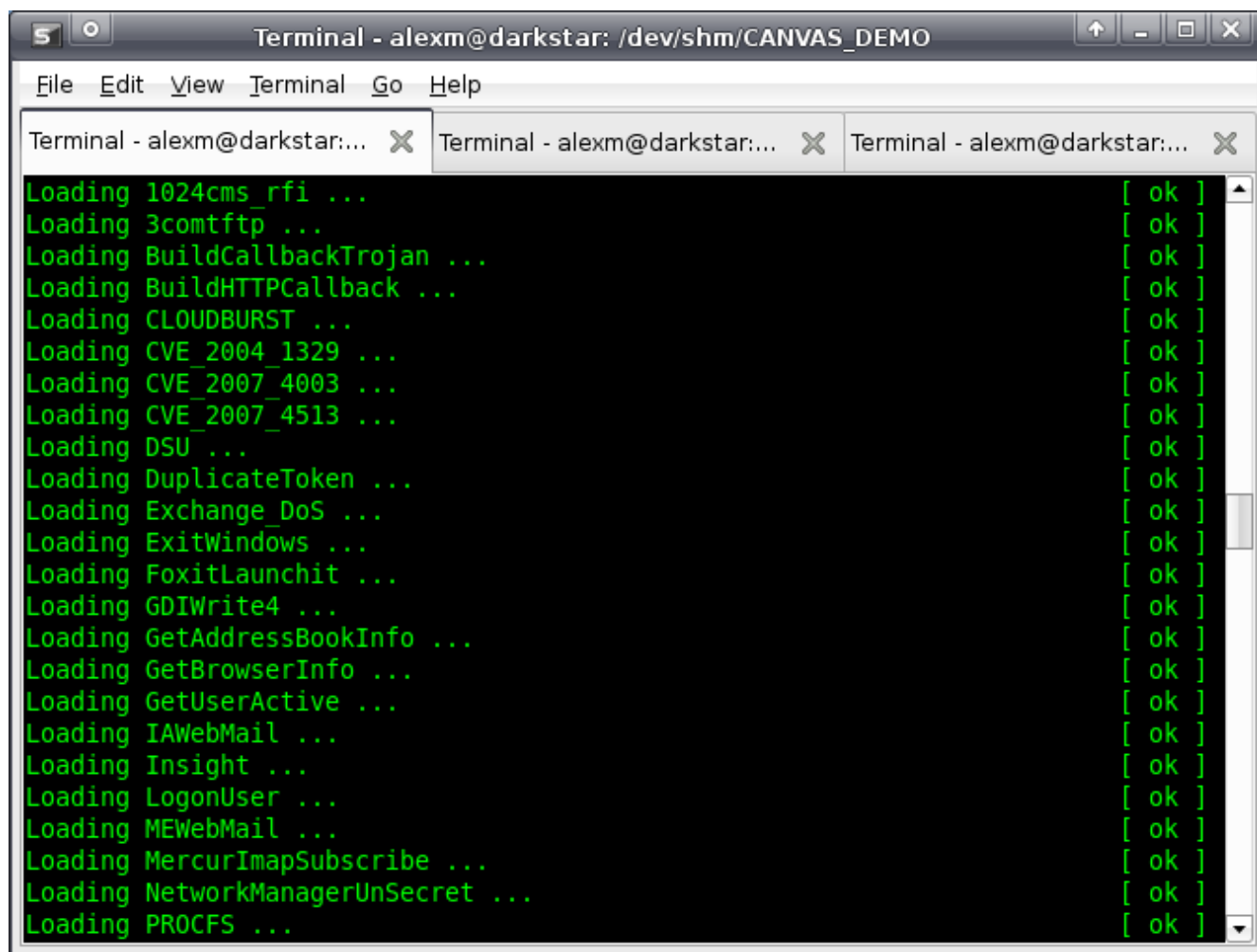
```
Setting CANVAS session to: default
Using 'Reports/default' as base data output directory
Required version of Python detected, using version 2.5 :-)
Generating parse tables for: [['Win32', 'X86'], ['Linux', 'X86'], ['Linux', 'ppc'], ['Solaris', 'sparc'], ['Solaris', 'X86'], ['OSX', 'ppc'], ['OSX', 'x86'], ['AIX', 'ppc', '5.1'], ['AIX', 'ppc', '5.2']] XORKEY=e4
FD: -1
*Parsetables successfully generated.
Last Updated: February 11, 2009

IMMUNITY, INC.

SOFTWARE LICENSE AGREEMENT

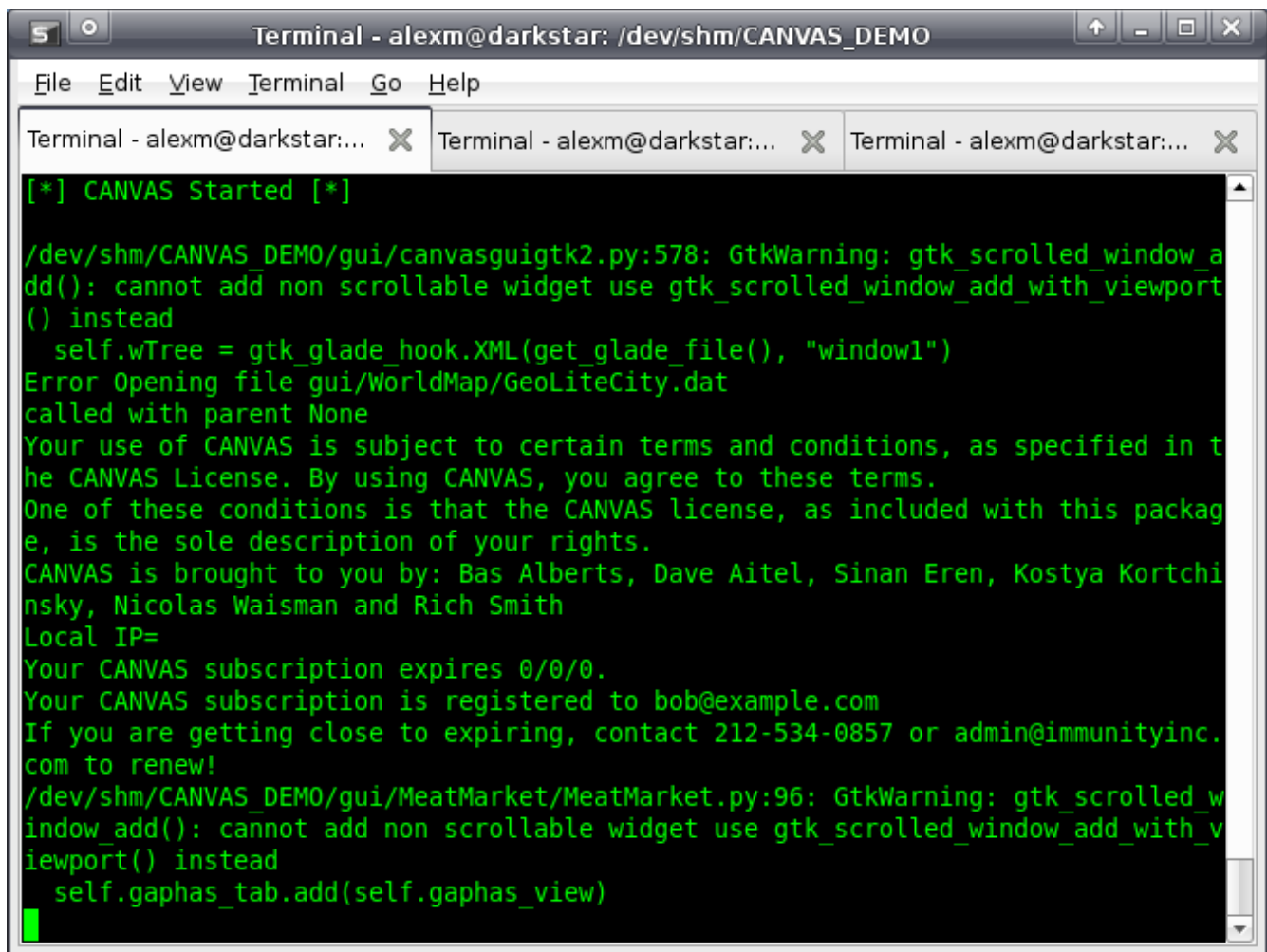
THIS LICENSE AGREEMENT (with the schedules annexed hereto, the "Agreement") is
made as of the day when registered on the download server between "Licensee",
```

After the license agreement you'll see all the modules being loaded into memory, each time you start this version of CANVAS from now on this is primarily what you'll be looking at.



```
Terminal - alexm@darkstar: /dev/shm/CANVAS_DEMO
File Edit View Terminal Go Help
Terminal - alexm@darkstar:... x Terminal - alexm@darkstar:... x Terminal - alexm@darkstar:... x
Loading 1024cms_rfi ... [ ok ]
Loading 3comtftp ... [ ok ]
Loading BuildCallbackTrojan ... [ ok ]
Loading BuildHTTPCallback ... [ ok ]
Loading CLOUDBURST ... [ ok ]
Loading CVE_2004_1329 ... [ ok ]
Loading CVE_2007_4003 ... [ ok ]
Loading CVE_2007_4513 ... [ ok ]
Loading DSU ... [ ok ]
Loading DuplicateToken ... [ ok ]
Loading Exchange_DoS ... [ ok ]
Loading ExitWindows ... [ ok ]
Loading FoxitLaunchit ... [ ok ]
Loading GDIWrite4 ... [ ok ]
Loading GetAddressBookInfo ... [ ok ]
Loading GetBrowserInfo ... [ ok ]
Loading GetUserActive ... [ ok ]
Loading IAWebMail ... [ ok ]
Loading Insight ... [ ok ]
Loading LogonUser ... [ ok ]
Loading MWebMail ... [ ok ]
Loading MercurImapSubscribe ... [ ok ]
Loading NetworkManagerUnSecret ... [ ok ]
Loading PROCFS ... [ ok ]
```

Finally, depending on your OS and version you may see some GTK warnings pop up just before the GUI spawns. This is normal and they are safe to ignore.

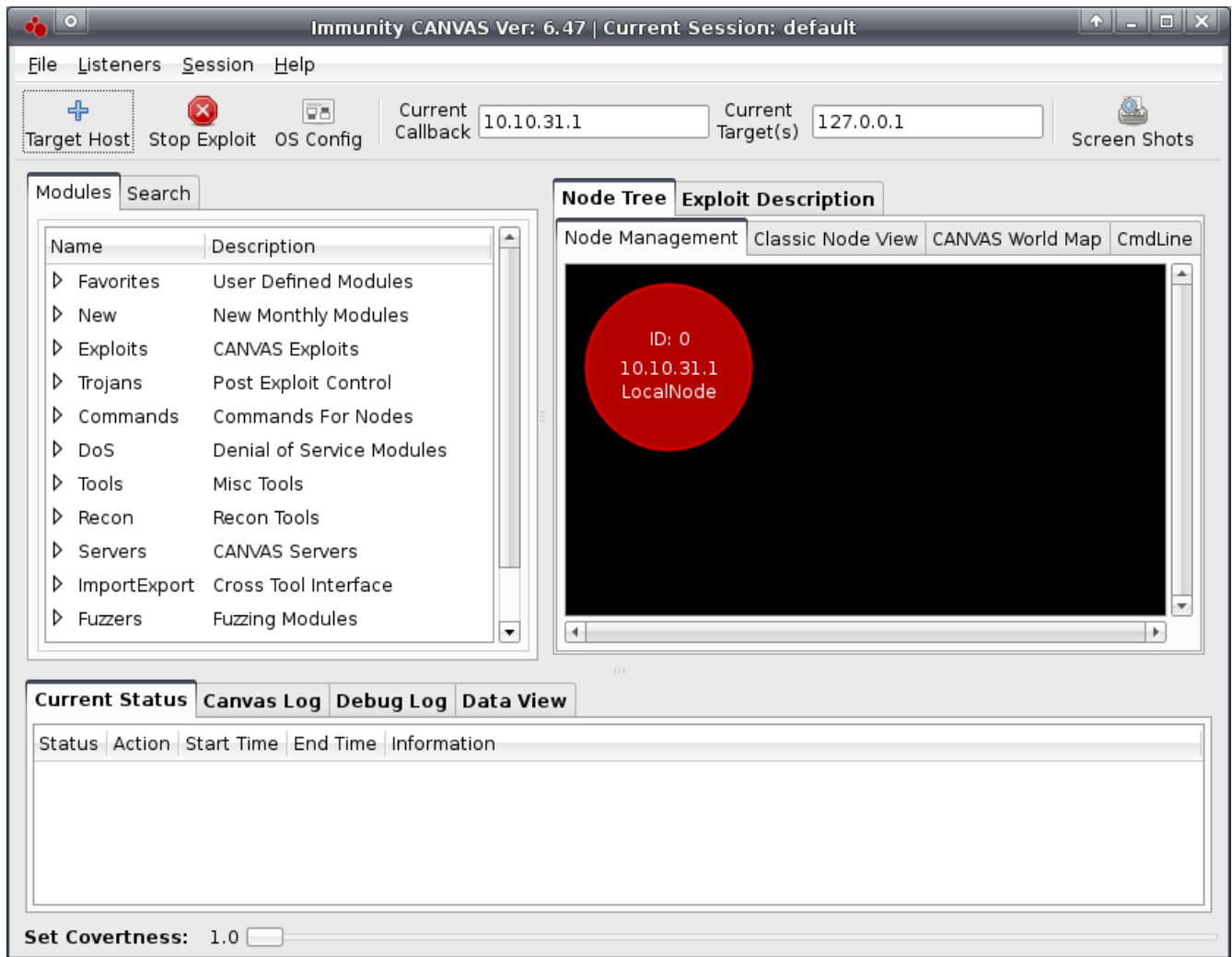
A terminal window titled "Terminal - alexm@darkstar: /dev/shm/CANVAS_DEMO" with a menu bar (File, Edit, View, Terminal, Go, Help) and three tabs. The terminal output is as follows:

```
[*] CANVAS Started [*]

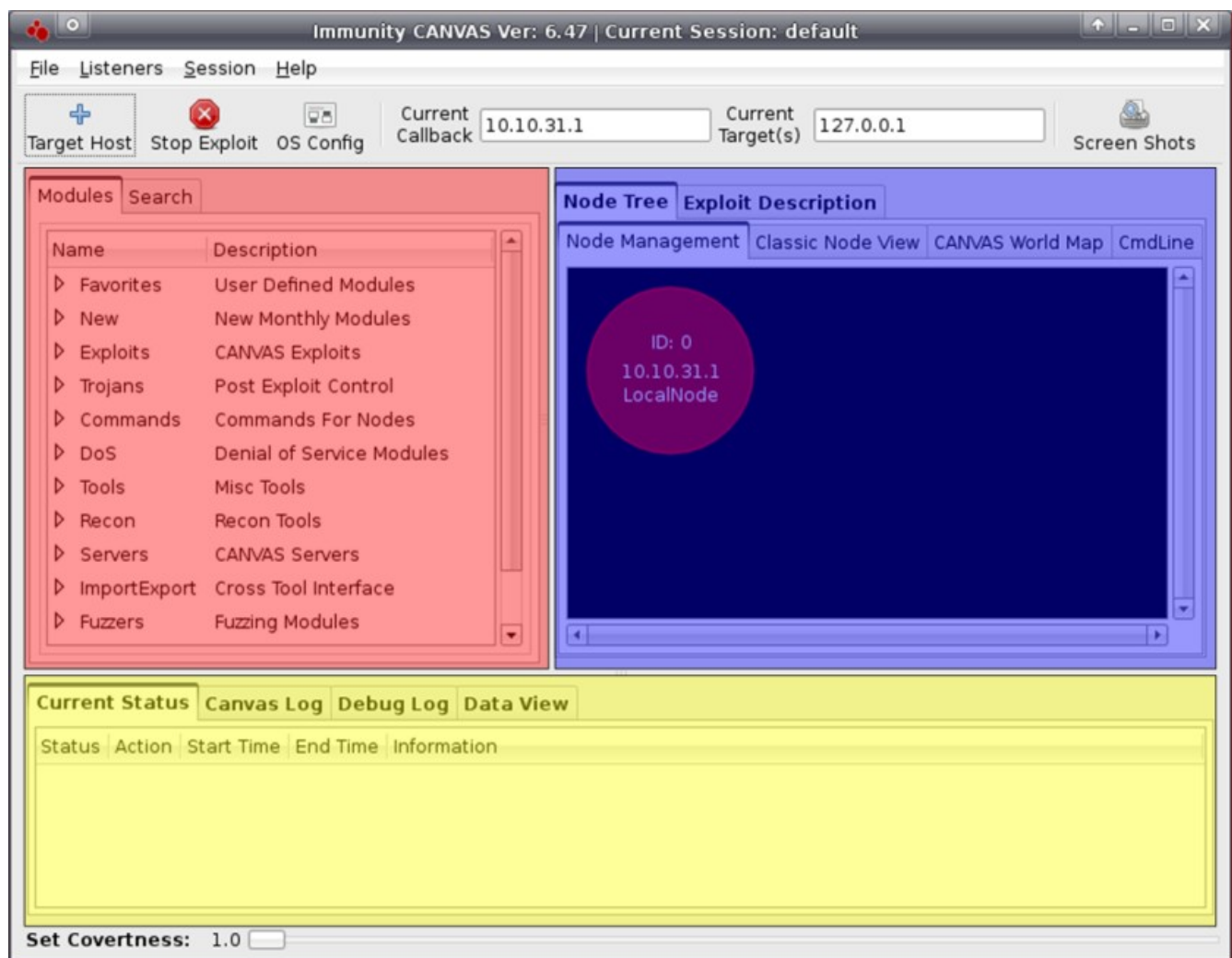
/dev/shm/CANVAS_DEMO/gui/canvasguigtk2.py:578: GtkWarning: gtk_scrolled_window_add(): cannot add non scrollable widget use gtk_scrolled_window_add_with_viewport() instead
  self.wTree = gtk_glade_hook.XML(get_glade_file(), "window1")
Error Opening file gui/WorldMap/GeoLifeCity.dat
called with parent None
Your use of CANVAS is subject to certain terms and conditions, as specified in the CANVAS License. By using CANVAS, you agree to these terms.
One of these conditions is that the CANVAS license, as included with this package, is the sole description of your rights.
CANVAS is brought to you by: Bas Alberts, Dave Aitel, Sinan Eren, Kostya Kortchinsky, Nicolas Waisman and Rich Smith
Local IP=
Your CANVAS subscription expires 0/0/0.
Your CANVAS subscription is registered to bob@example.com
If you are getting close to expiring, contact 212-534-0857 or admin@immunityinc.com to renew!
/dev/shm/CANVAS_DEMO/gui/MeatMarket/MeatMarket.py:96: GtkWarning: gtk_scrolled_window_add(): cannot add non scrollable widget use gtk_scrolled_window_add_with_viewport() instead
  self.gaphas_tab.add(self.gaphas_view)
```

Understanding the GUI Organization

This is the GUI as it appears in the current versions of CANVAS if you're working from an older copy (pre early 2009) the GUI will be similar but it will have some important differences, of course we encourage you to renew your support contract ;D



For now lets take a quick look at how the GUI is organized in broad strokes.

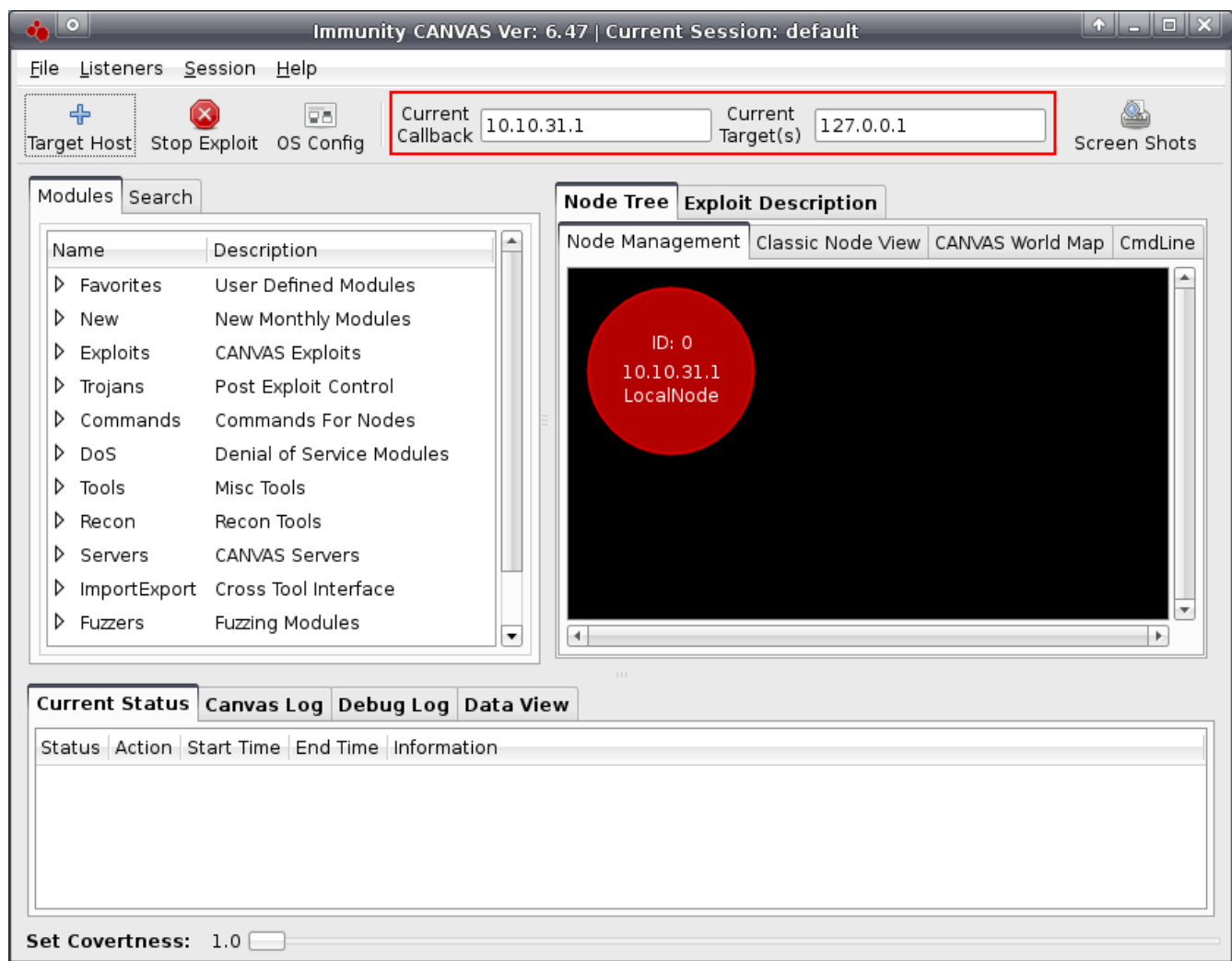


You can think of the highlighted sections as corresponding to these general ideas:

Red = Things CANVAS can do (i.e. modules)

Blue = What CANVAS knows about the world

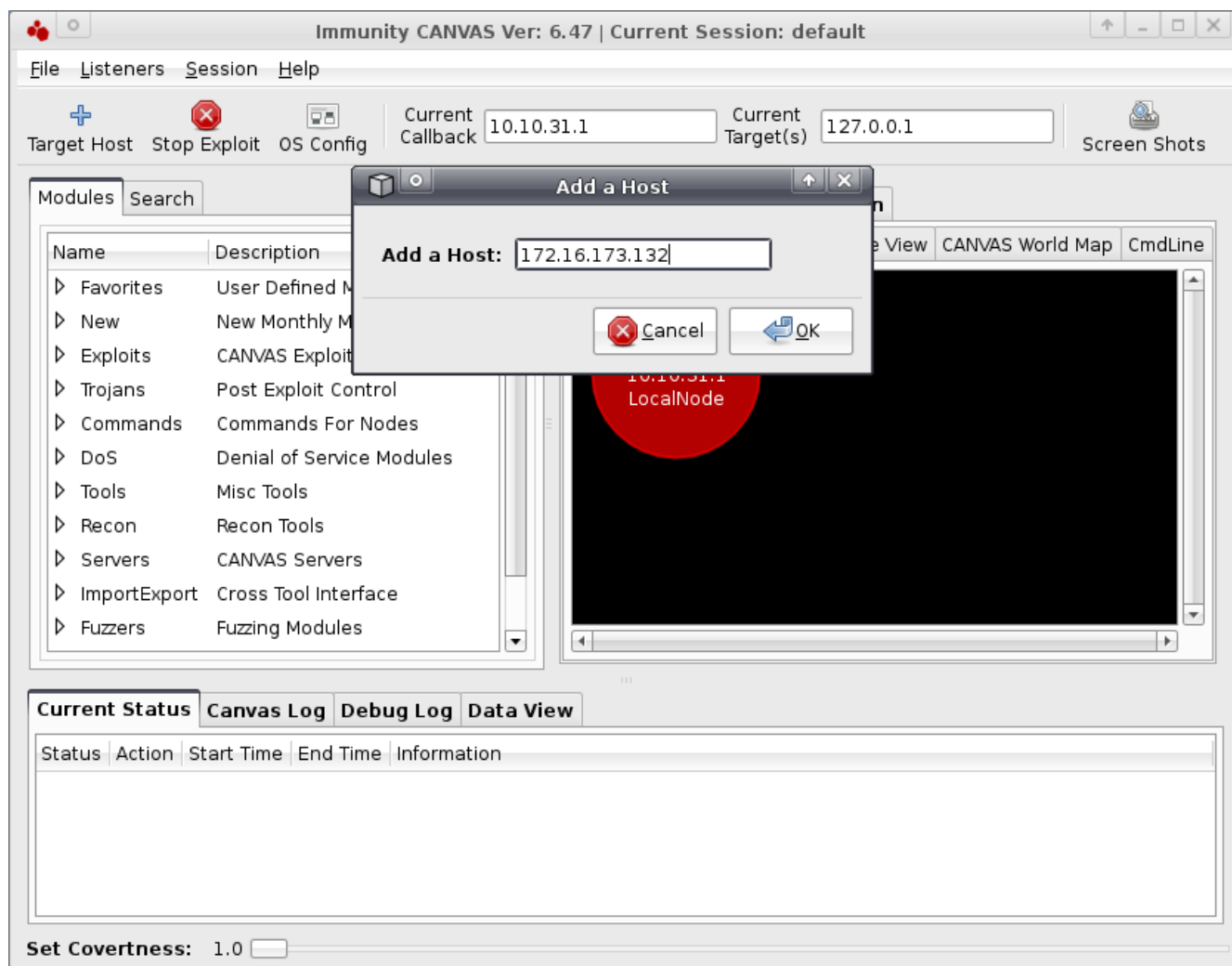
Yellow = What CANVAS is doing at any given time



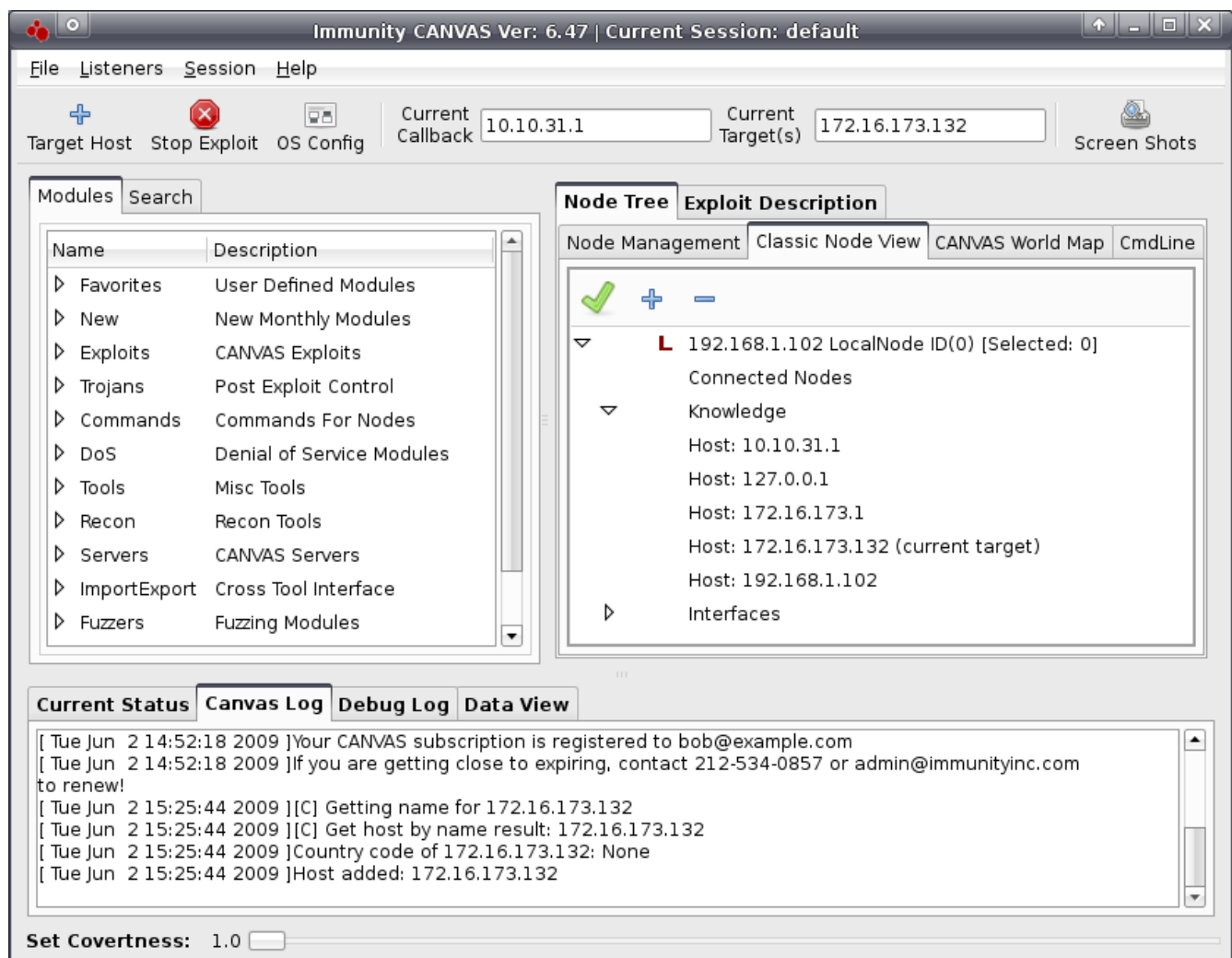
One of the biggest issues first time CANVAS users have is thinking the callback and target fields are manually editable (here outlined in red) you have to actually set these values through the GUI, it's a bit non-intuitive at first but once you've been using CANVAS for a little while it's easy to pick up.

Selecting Hosts / Launching Modules

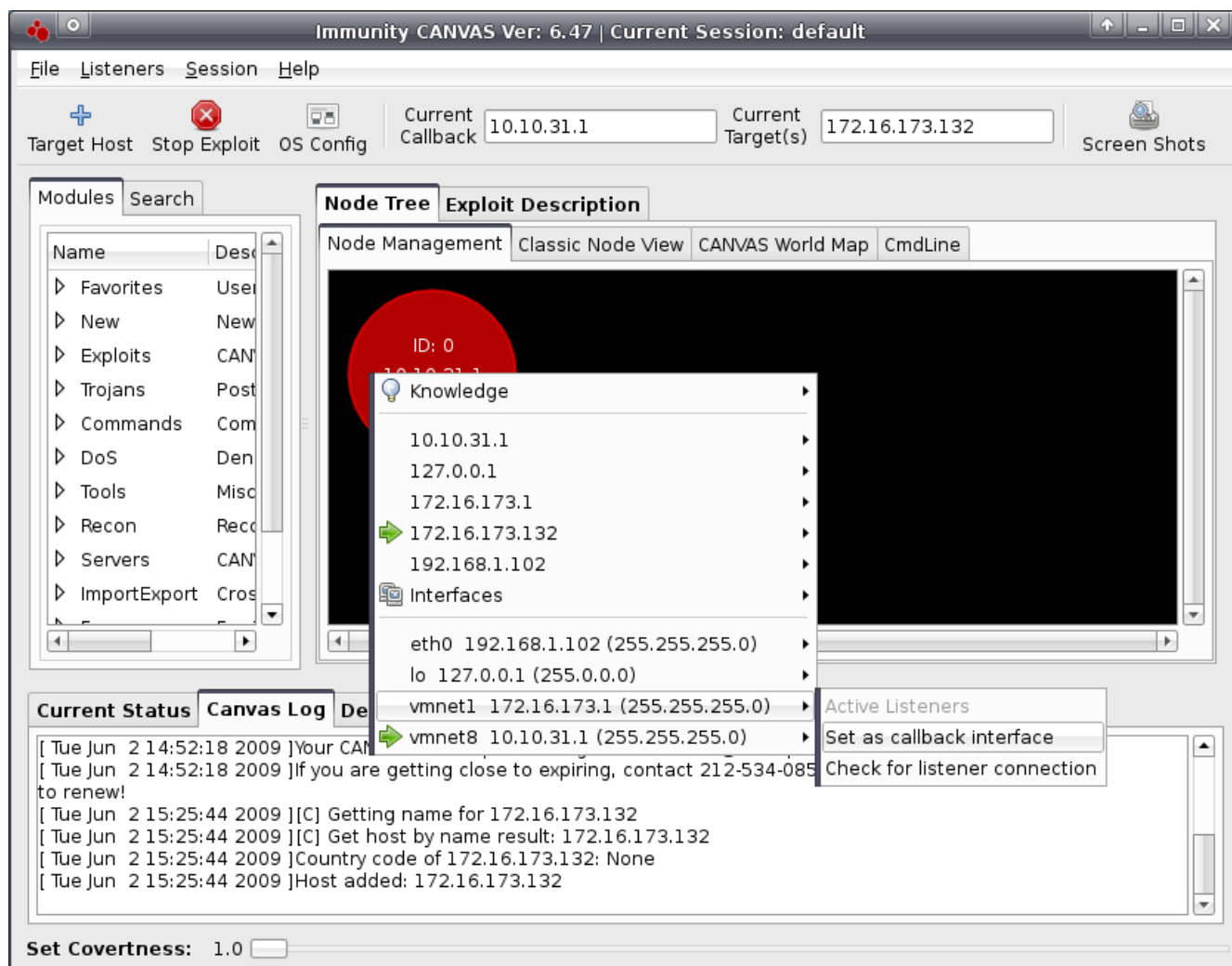
Now that we understand a bit more about how the GUI is organized, let's get to the business of using CANVAS. The most common usage case folks encounter with CANVAS is launching remote exploits against hosts, so that's what we'll focus on here.



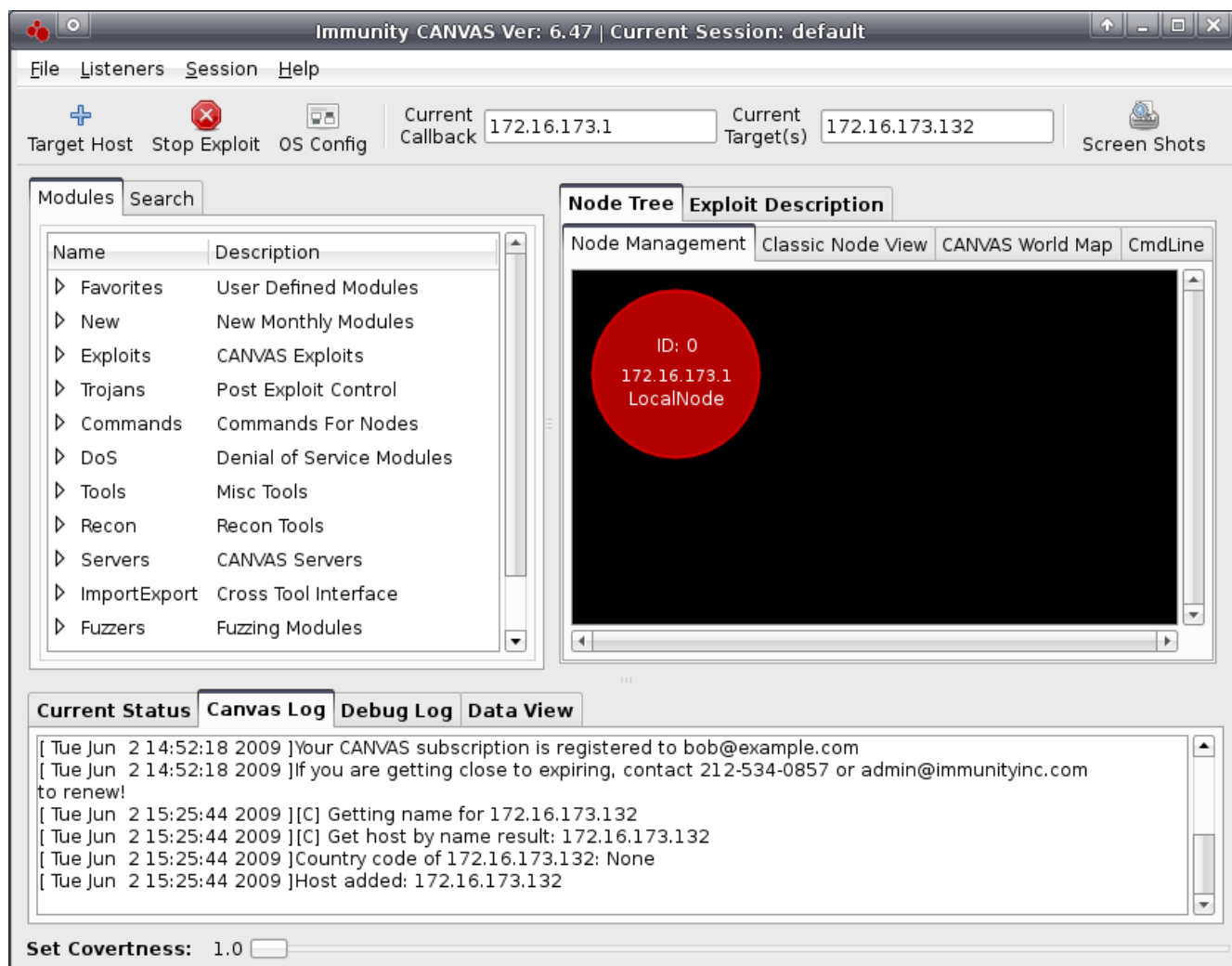
Step one is going to be adding a host, so what we'll do is click the Add Host button in the upper left hand corner of the CANVAS GUI and we're presented with the above window to which we put in an IP address though a hostname would also work so long as the computer running CANVAS is able to resolve it.



A few things have changed in the CANVAS GUI as the result of our previous action. First, the IP we just entered is now our **current target**. Second, if we click on **Classic Node View** in the blue section of the CANVAS GUI we can see the host has been added. As we start to find out about hosts and their attributes we'll see this information view populate with the information we find.

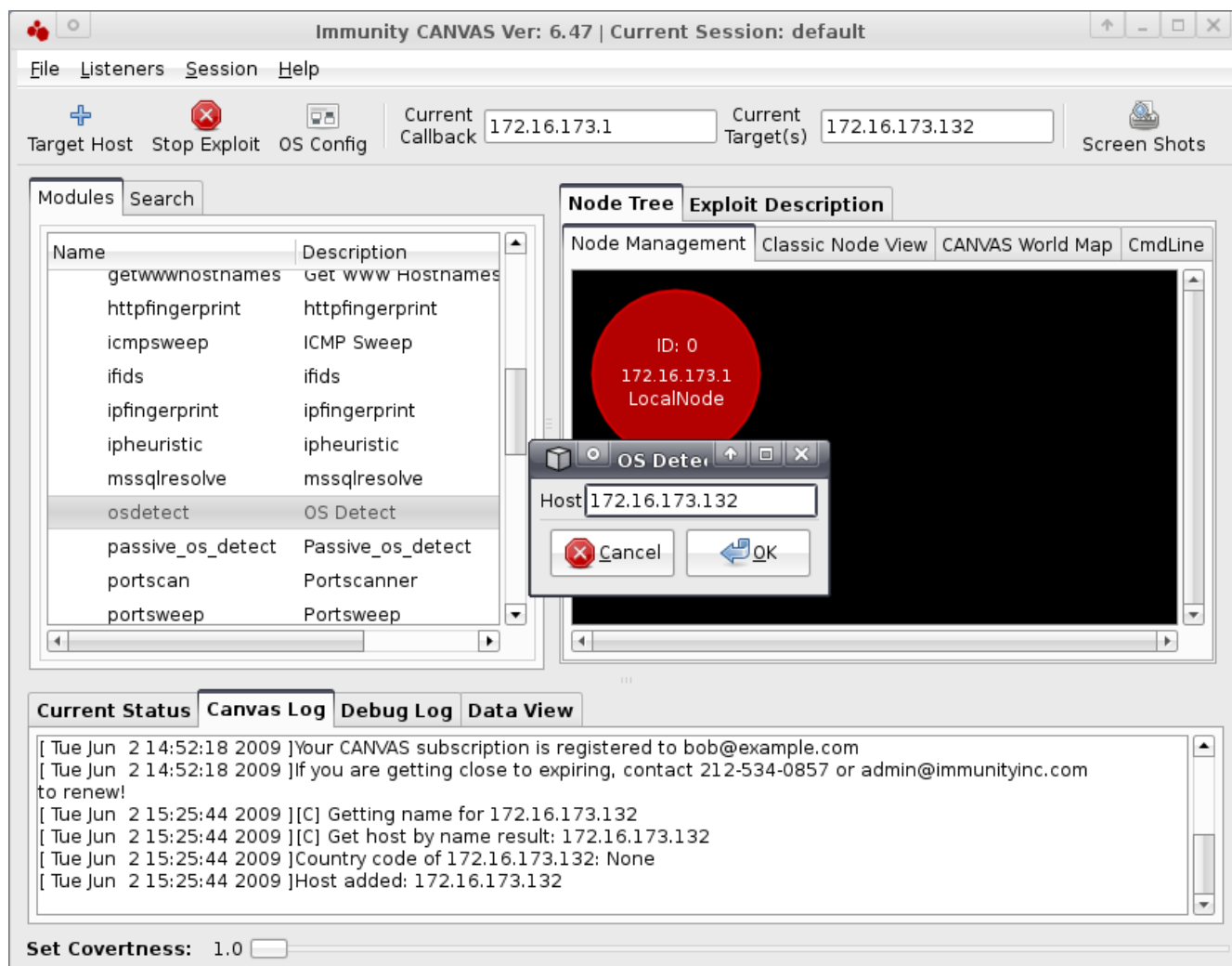


The next step is setting our **callback**. When we start running exploits CANVAS will take care of writing our shellcode automatically, setting the callback tells CANVAS that you want the exploited host to connect to the IP address provided. To do this we're going to go back to **Node Management** and right click on our local node (which is the representation of the host running CANVAS), scroll down to the interface we wish you use as our callback and select it as the callback interface.

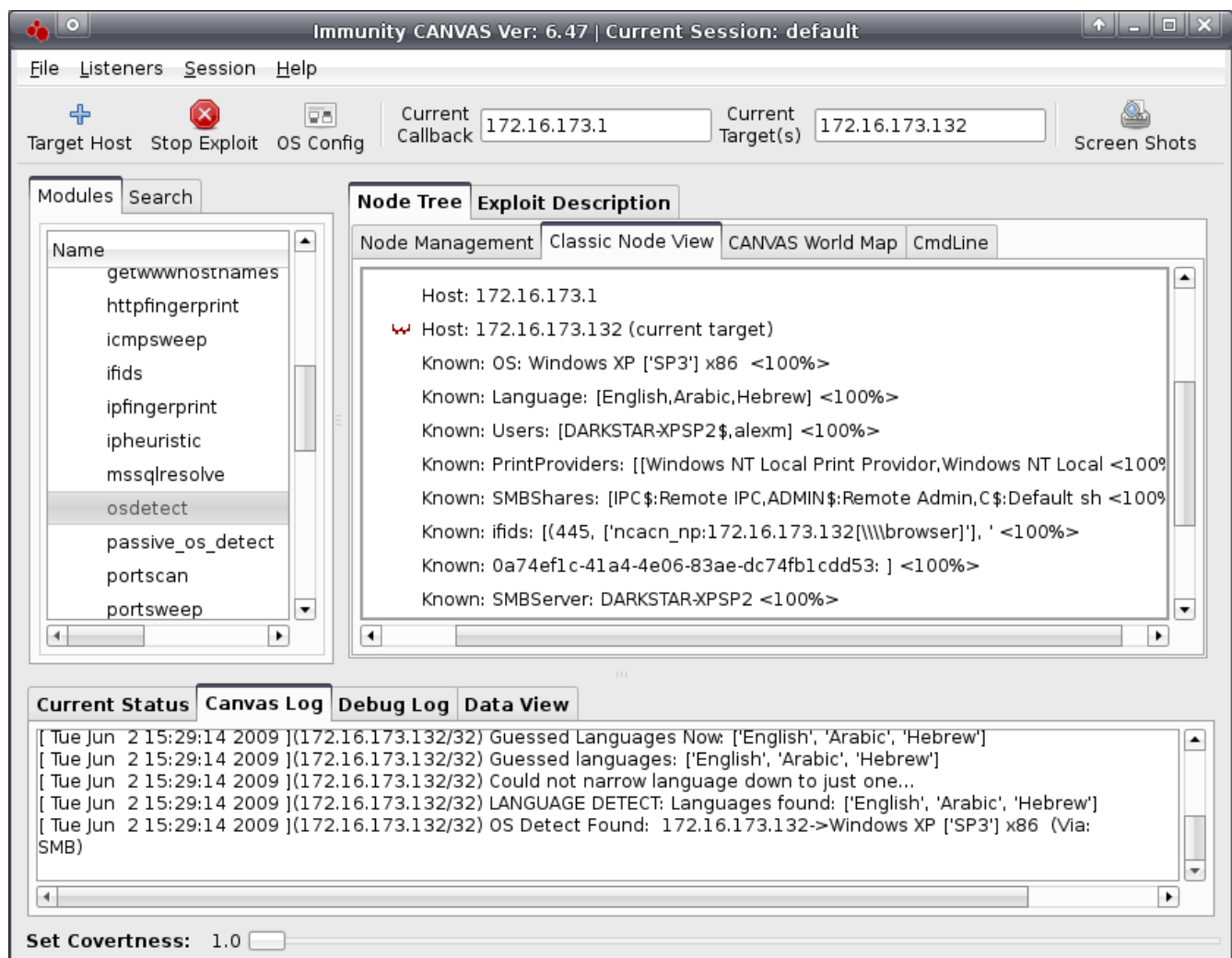


The CANVAS GUI has changed a bit again given our last action, we can see the callback address has now been set. If we click on the **CANVAS Log tab** from the yellow section of the GUI we can also see that when we added a host CANVAS recorded it. It's worth mentioning that the CANVAS Log tab simply pulls from the CANVAS.log file included in your CANVAS directory. CANVAS.log is just a simple flat text file that you can open with any text editor.

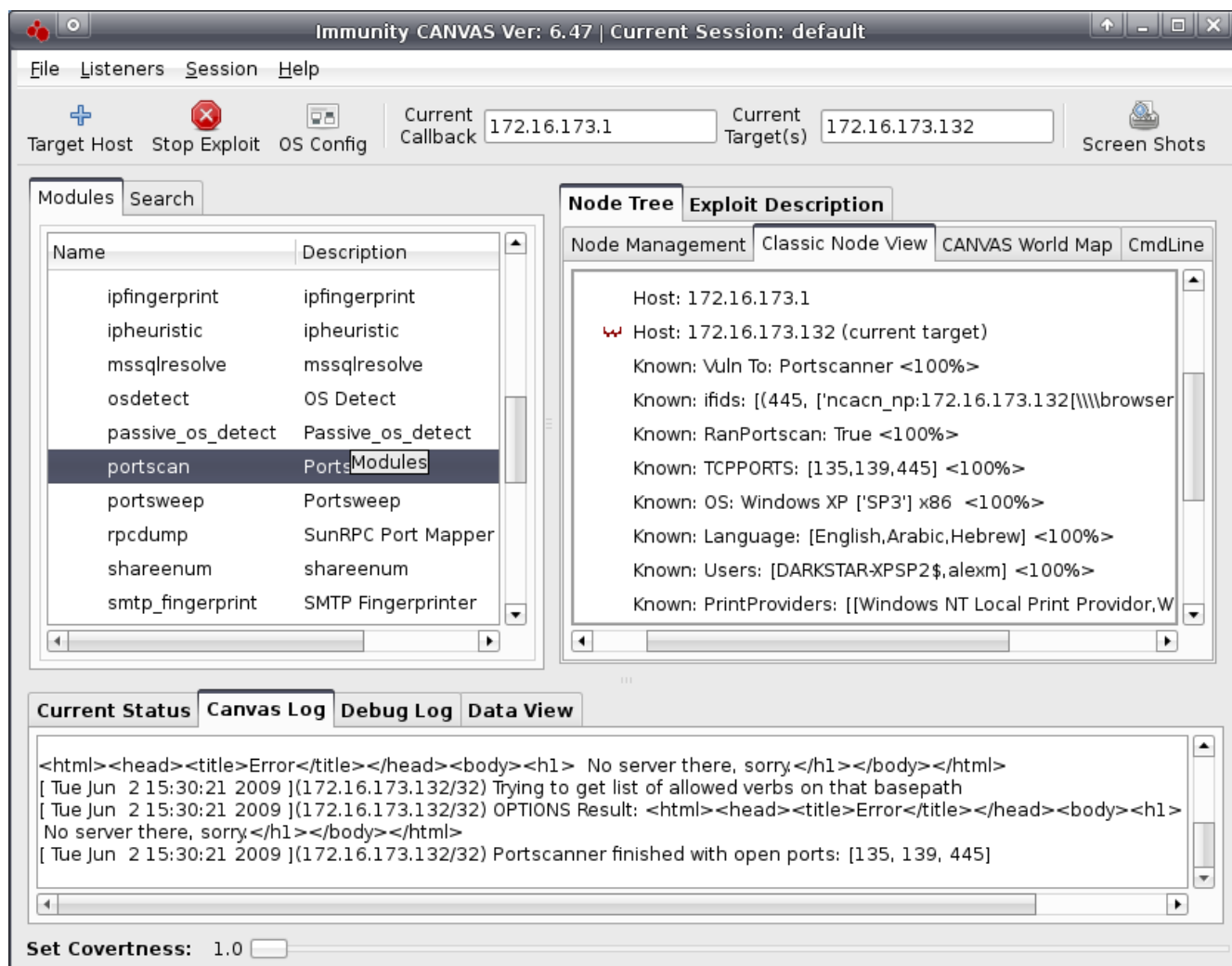
A good habit to form is that when you're using CANVAS you should always have the CANVAS Log tab open. Knowing what CANVAS is doing at any given second is preferable to having CANVAS give you summary updates.



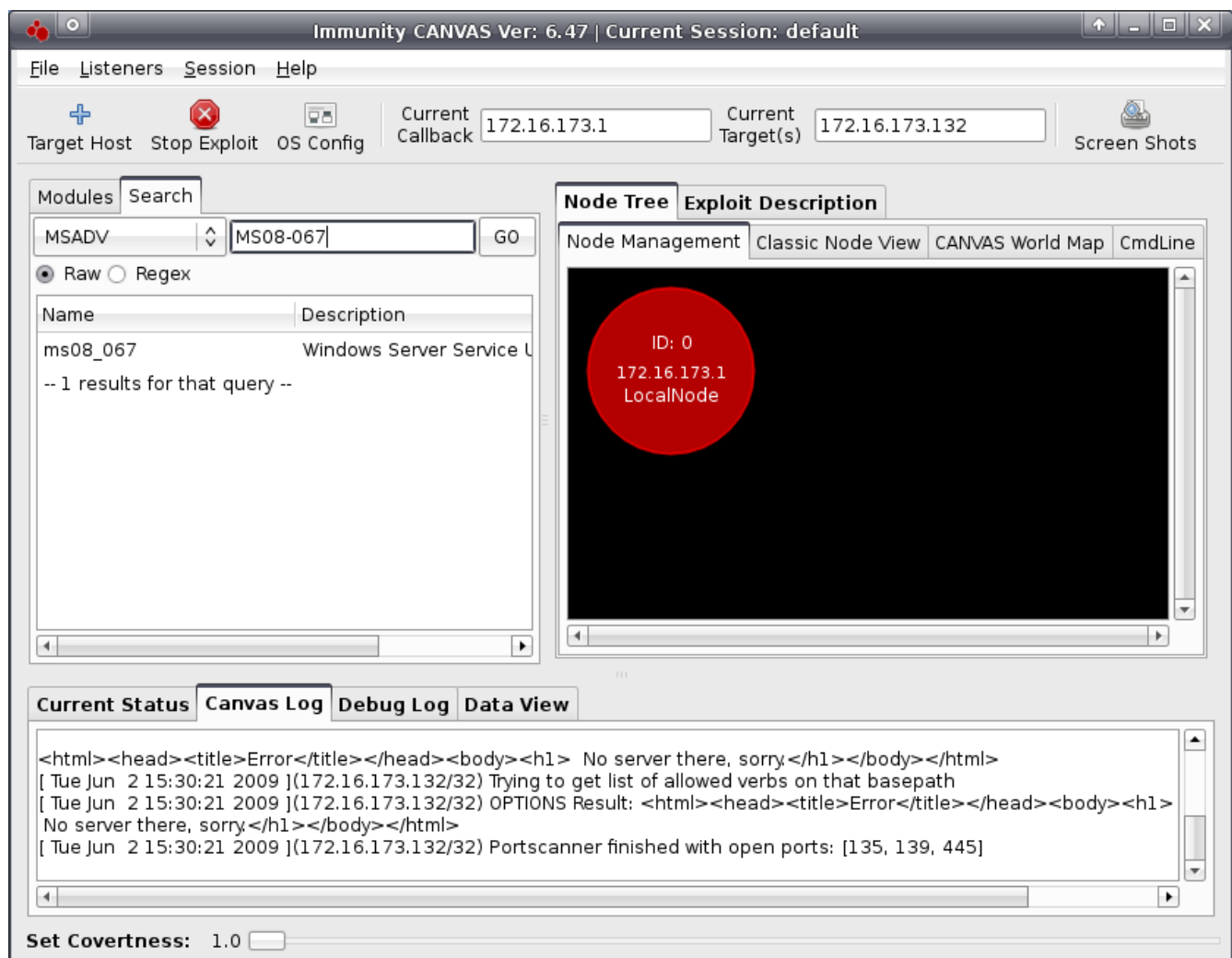
Now that we've got a host added to CANVAS it's time to find out a few things about it. **OS Detect** is a good place to start for this since we're on the same network segment as our target so within the red section of the CANVAS GUI we'll expand the Recon modules section and double click on OSDetect. You'll see the target IP we've selected is referenced, all that's left is to click ok.



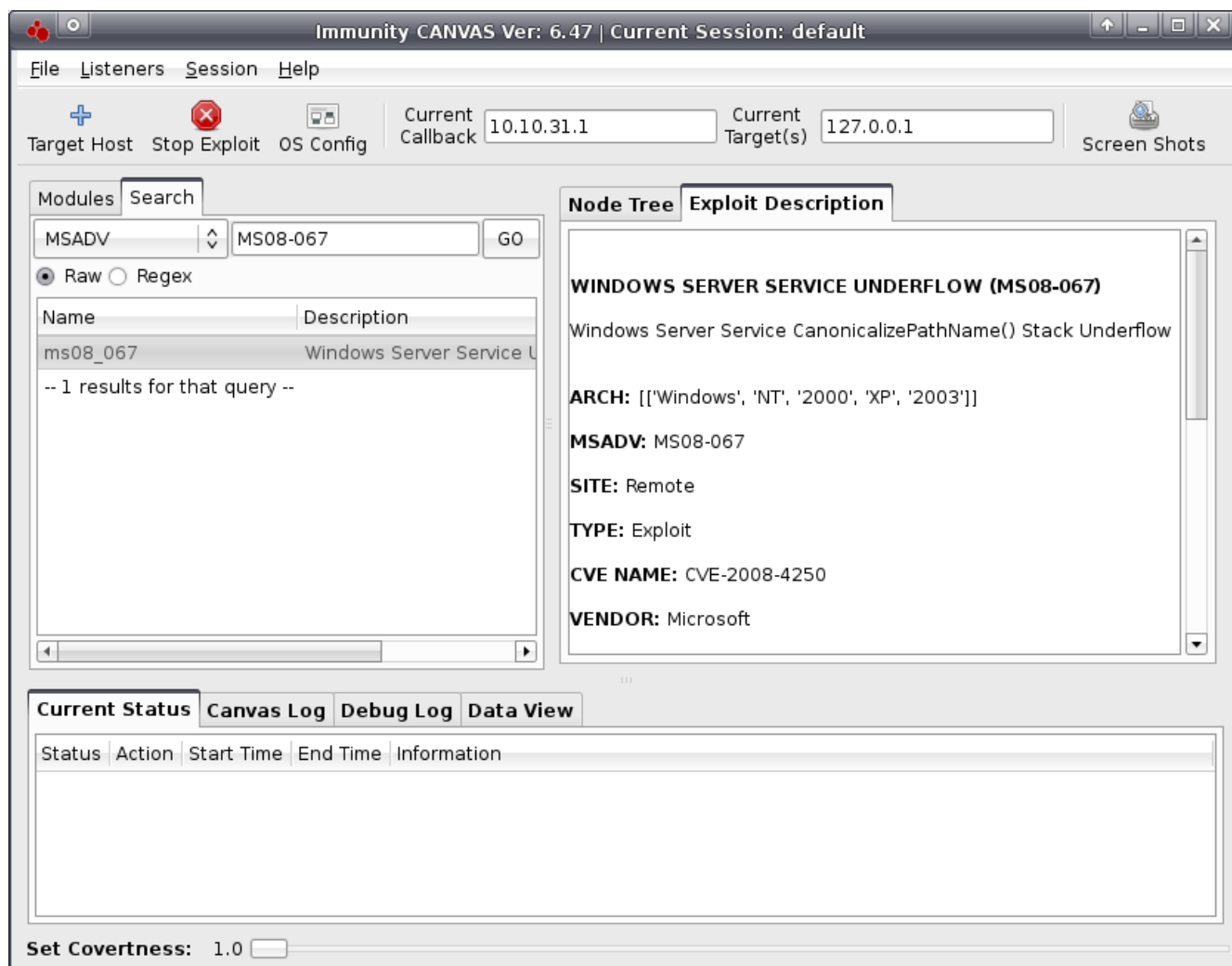
Looking at the **Classic Node View** we can see that a whole bunch of information has been added to the host! Next to the IP address of the host we added we can see a 'W' icon, which here is short for Windows. We can also see that we were able to find a lot of good information about the host as well.



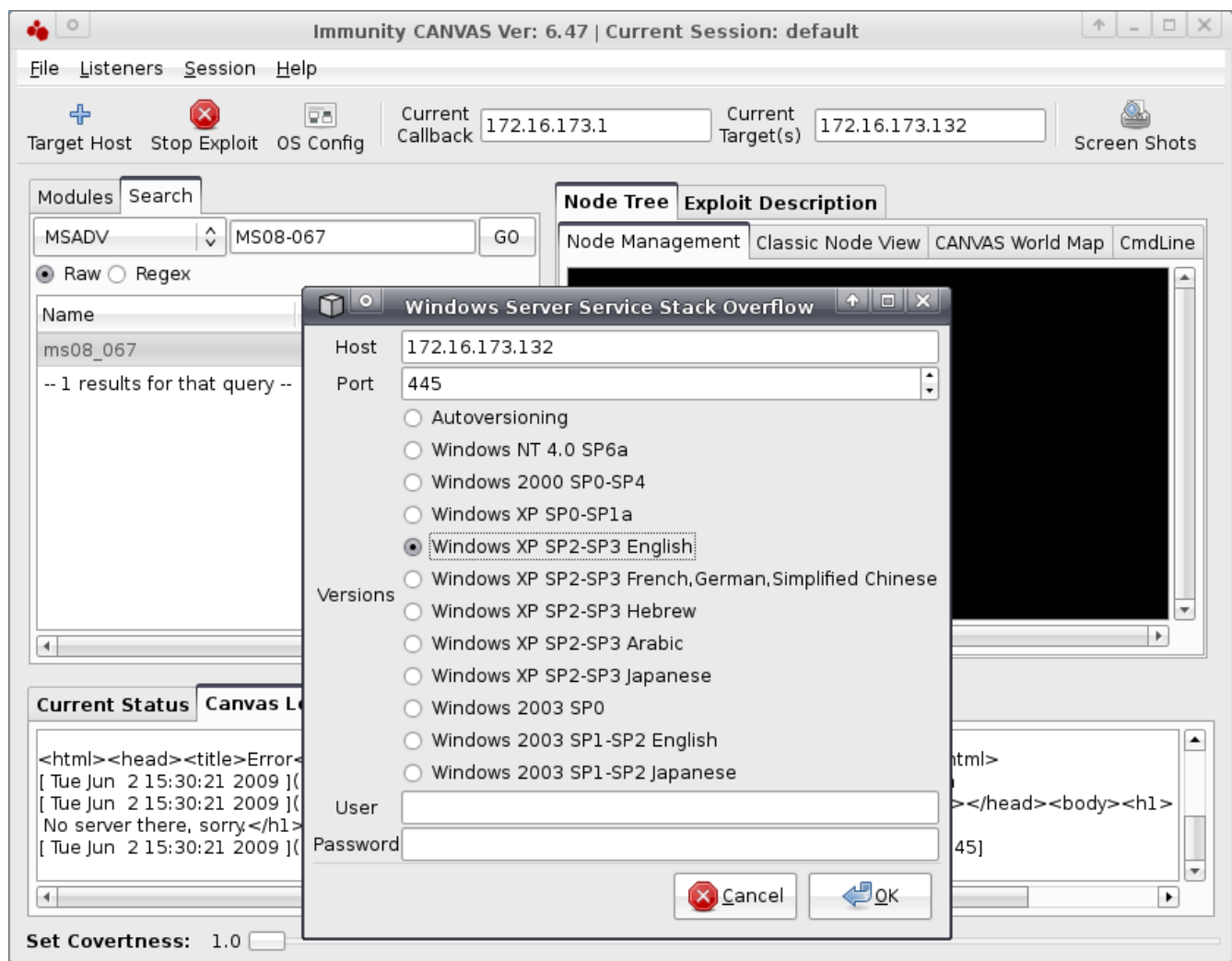
Next step is running a port scan, so we'll double click on the **portscan** module much the same way we did with OSDetect and launch it. Above we can see that we found the usual Windows ports open.



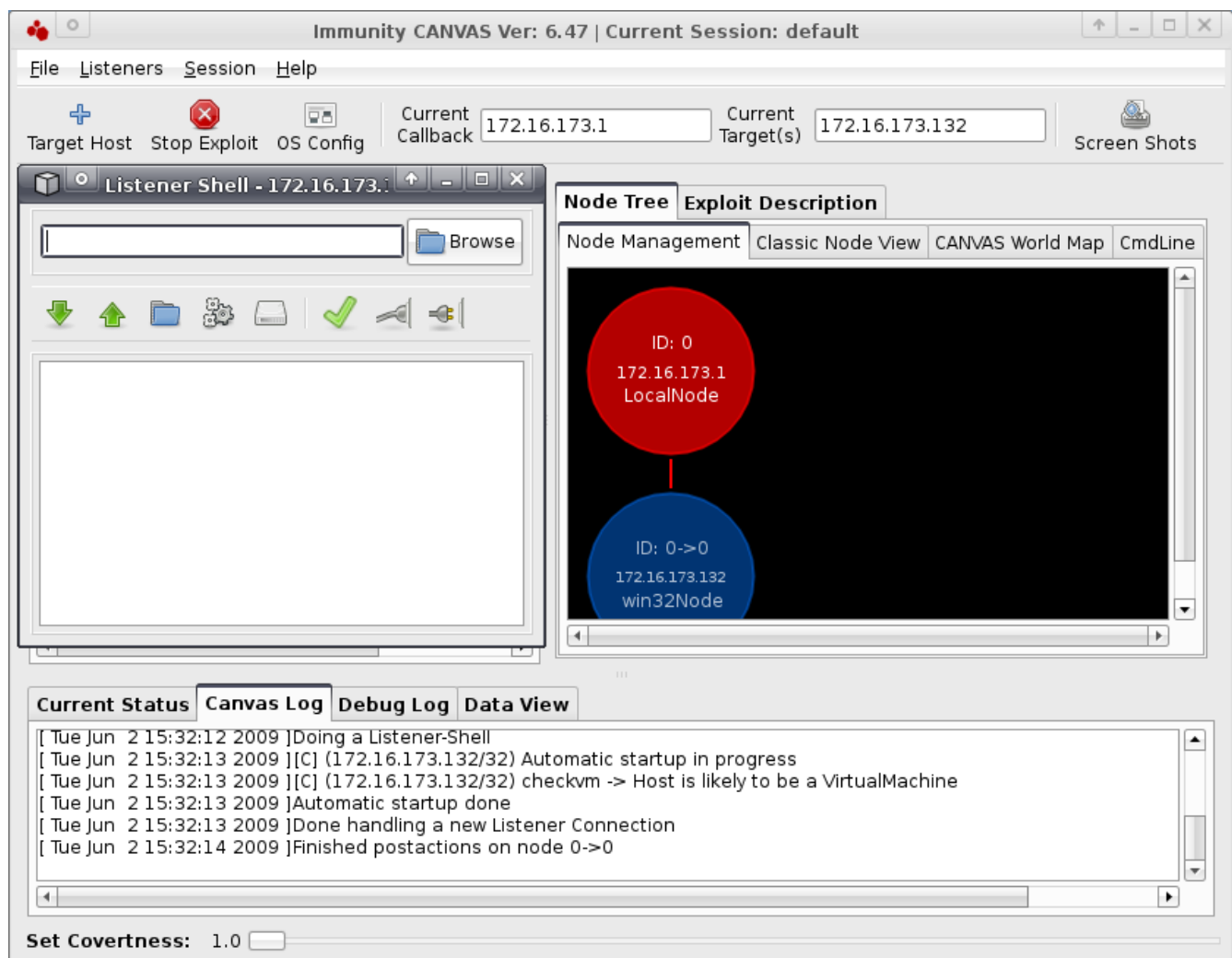
Now that we know this host is running Windows and it has the standard SMB ports open, we can look at running a few exploits. The latest exploit against the Microsoft SMB implementation is of course MS08-067, so we'll click on the **search** tab, use the drop down list to select **MSADV** and search for MS08-067. We could also have browsed for this manually by clicking on the modules tab and expanding Exploits > Remote > Windows > All Windows and double clicking on MS08_067.



One useful feature of the GUI is looking at the **Exploit Description** to find out more information. Once we've got a module selected (simply by clicking on it) we can click the **Exploit Description** tab and we get a lot of useful stuff to know about the exploit we're running.



Now it's time to launch the exploit! Simply double clicking on the exploit will bring up the options menu where we can select which version of the exploit we want to use. Here 'versioning' is a bit of a misnomer because they're all the same exploit we just have the appropriate memory addresses preprogrammed into it for varying localizations of different versions of Windows. Selecting **Autoversioning** would tell CANVAS that we want it to find out the version of the OS for us and select the appropriate version of the exploit, essentially automating the OSDetect and porstcan steps we took earlier. Since we already know this host is running the English version of XP, we'll choose that and just click ok!



Sure enough we are rewarded with a shell! And that's where this tutorial will end for this week as it's been pretty long already.

Conclusions

In this tutorial we did a bit of introductory work with CANVAS in hopes to get you familiar with how CANVAS operates. Next week we'll look at interacting with the host we just compromised in a variety of ways.

Resources for Further Thought

There aren't many printed materials that are relevant to this tutorial so I'll just link to what I've been reading recently.

- As always Immunity will [teach you](#) all about CANVAS and other security topics
- *Anansi Boys* by Neil Gaiman - not as good as *American Gods* but a very enjoyable read
- *Neverwhere* by Neil Gaiman - novelization of the BBC TV Series, not bad but not his strongest work either
- This week's tutorial is brought to you by [Andrew W.K.](#) cuz we like to party