Hydrogen Cyanide (HCN) User Guide

Part I: Almond Scented Joy

Introduction:

Immunity is releasing its latest rootkit for Windows platforms in September 2009 with CANVAS version 6.50. Hydrogen Cyanide (HCN) is a complete ground up rewrite of the original Windows rootkit and has been tested on all versions of Windows from Windows 2000 through Windows 7. The new rootkit offers some new features in addition to the functionality included in the previous version. This tutorial will focus on HCN's installation and the beaconing feature.

Disclaimer:

HCN works by loading a driver to execute commands within the context of the Windows kernel. Though extensive testing of the rootkit has been done Immunity strongly cautions users wishing to deploy HCN on mission-critical production systems. Proper backups should be in hand prior to loading HCN on such a system.

Known Issues:

Hiding directory names/paths that contain spaces may have unexpected results on Windows 2000. Unloading the rootkit from these systems will solve the problem.

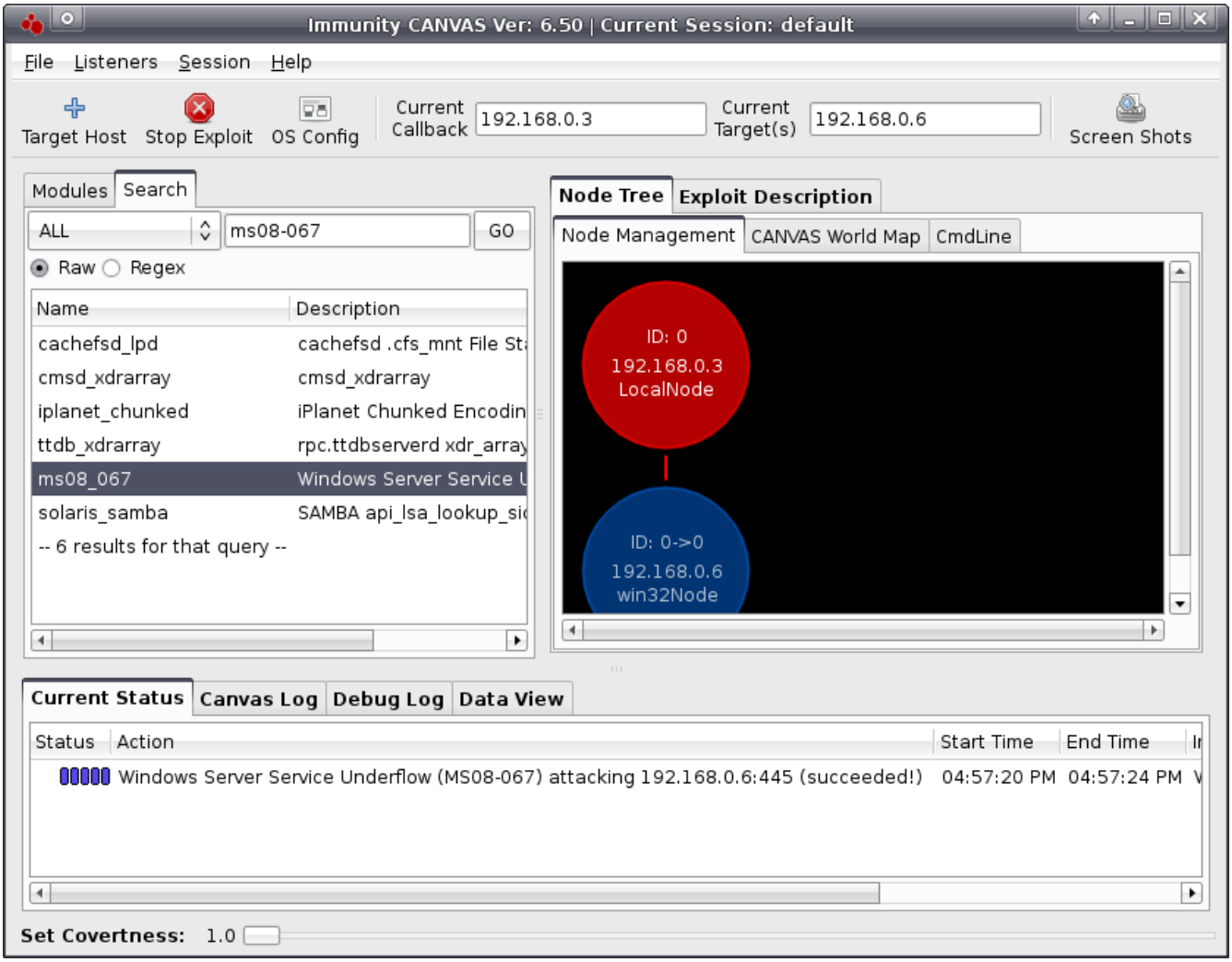
Goals:

At the end of this tutorial you will be able to --

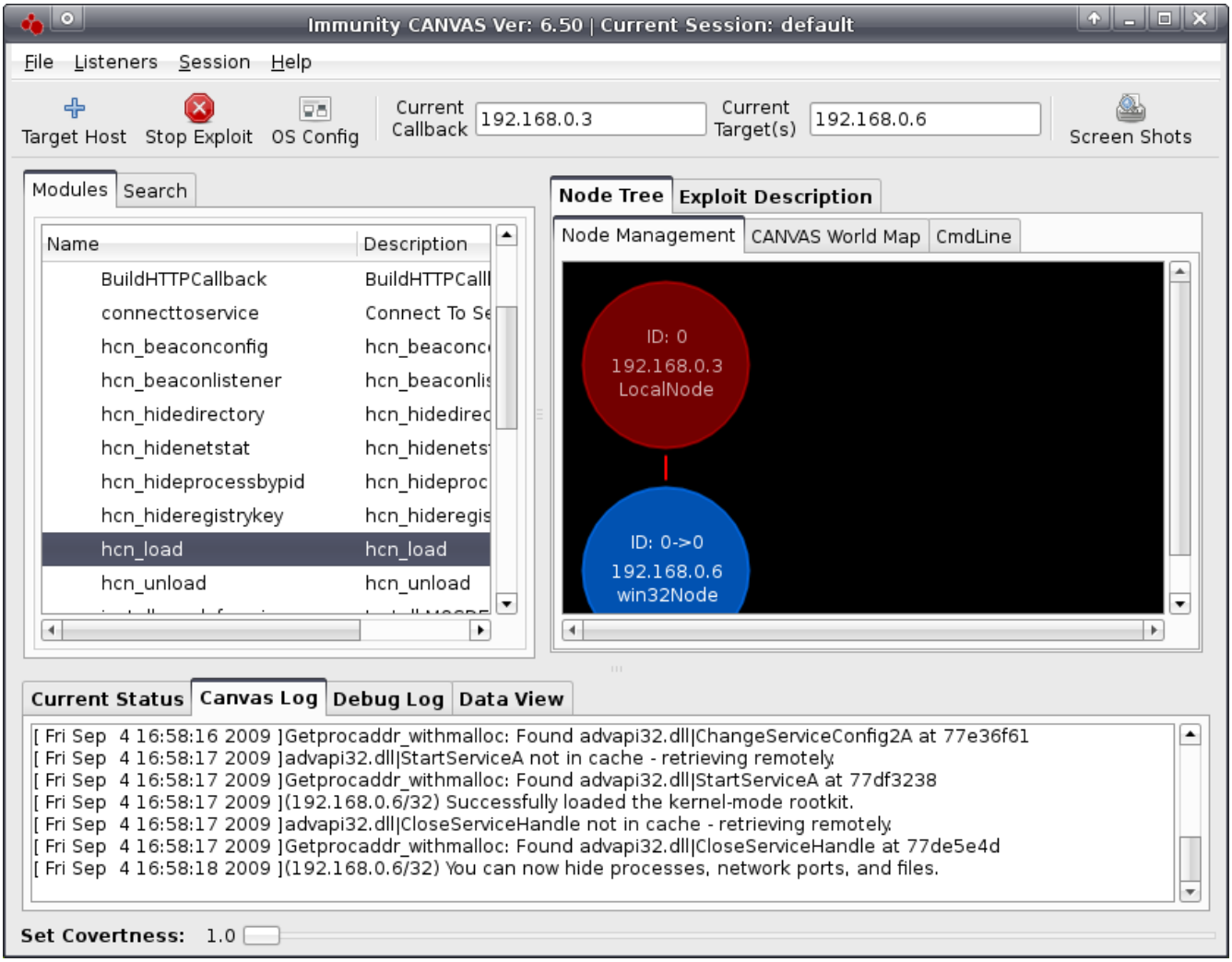
- Load and unload the HCN rootkit
- Set up the beaconing feature

Loading and unloading the HCN rootkit:

Let's consider a common scenario, you've successfully compromised a Windows based system (in this case we're using MS08-067) and now want to cement future access to it. Your CANVAS GUI looks something like this



Your next step will be to select the new win32Node by clicking on it, then running the hcn_load module. This module can be found under the *Trojans* section and takes no arguments, so simply double click and select 'Ok' to install the rootkit. In the below picture we can see the rootkit has been successfully loaded by looking at the messages in the CANVAS Log tab.

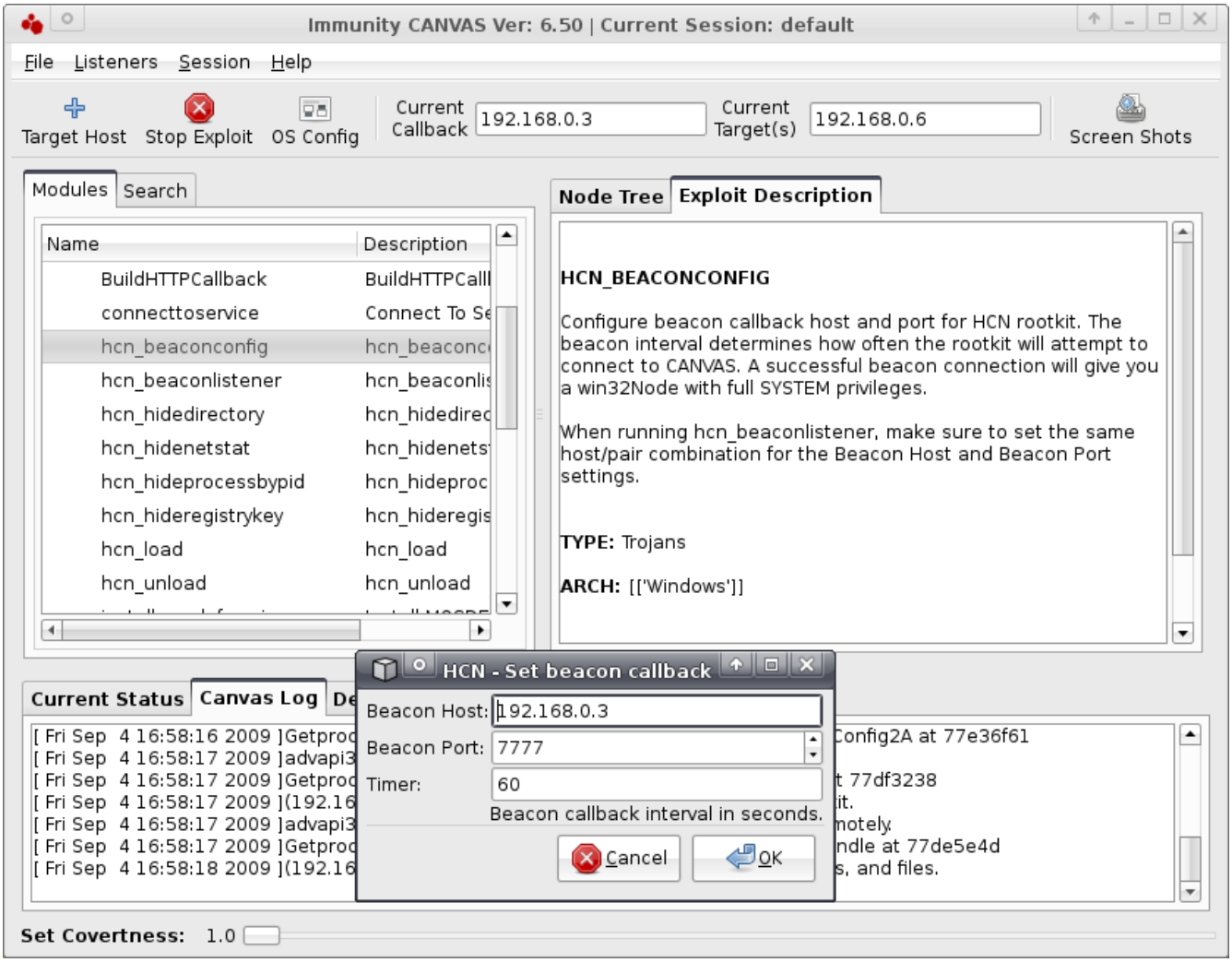


Unloading the rootkit is just as simple! Double click on the *hcn_unload* module, ensure the host with HCN loaded is selected, then simply click 'Ok'. Once completed the host will have to be rebooted.

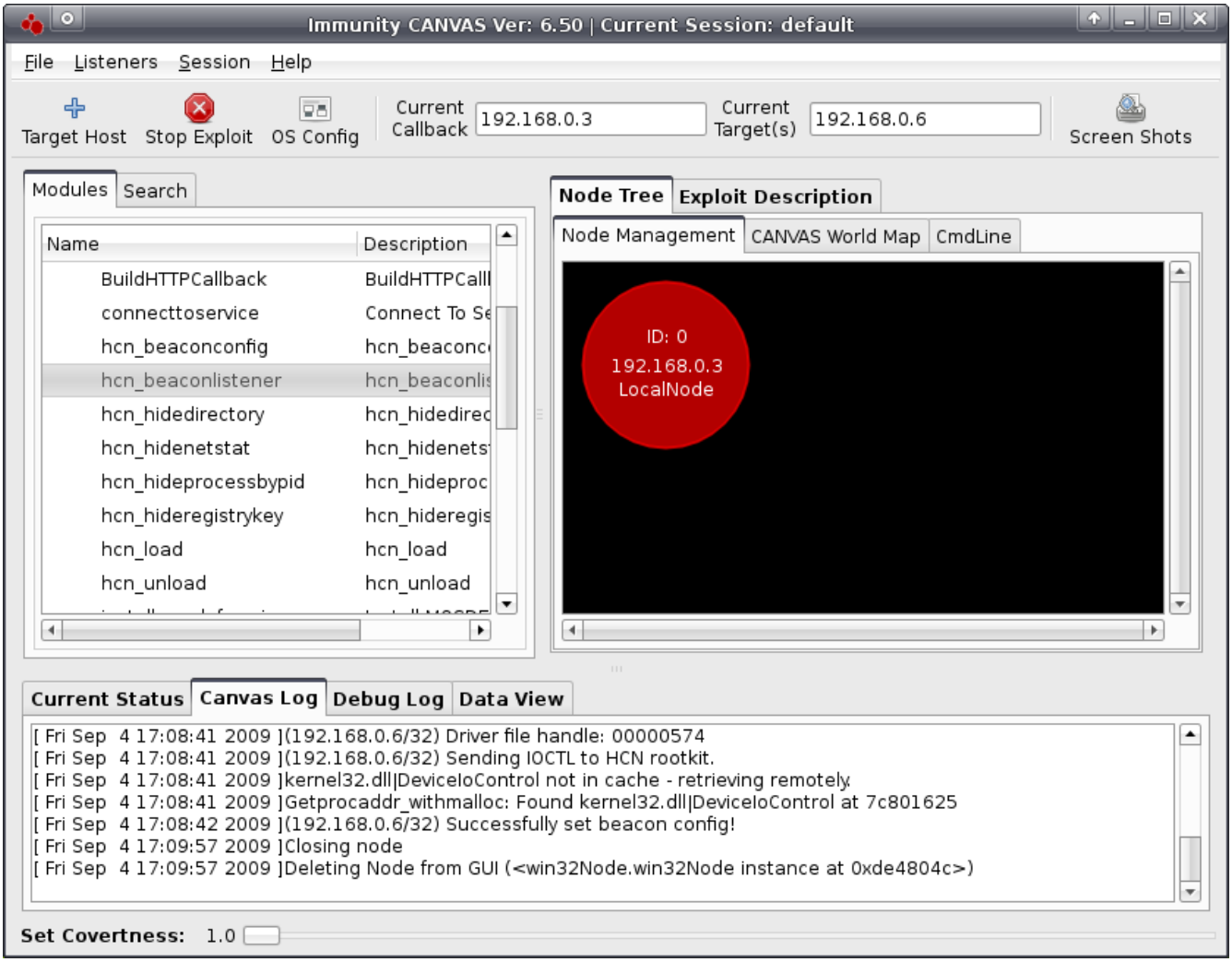
One of the common reasons to install a rootkit is to ensure continuing access. By staying in memory we can be much more stealthy and avoid some defensive security technologies however this comes at the price of being unable to guarantee future access. If you exist solely in memory and the host is rebooted, the process you exist in is restarted or a litany of other common issues you'll loose connectivity to our compromised host.

HCN comes with a unique beaconing feature. You can instruct the compromised host to attempt to connect to a predefined address/port at a set interval and receive a shell. This way if new firewall rules are constructed during your pen-test (this is more common than many wish to believe) there's still a reasonable chance of maintaining connectivity. One draw back is that currently this technique can be beaten by the creation of egress rules.

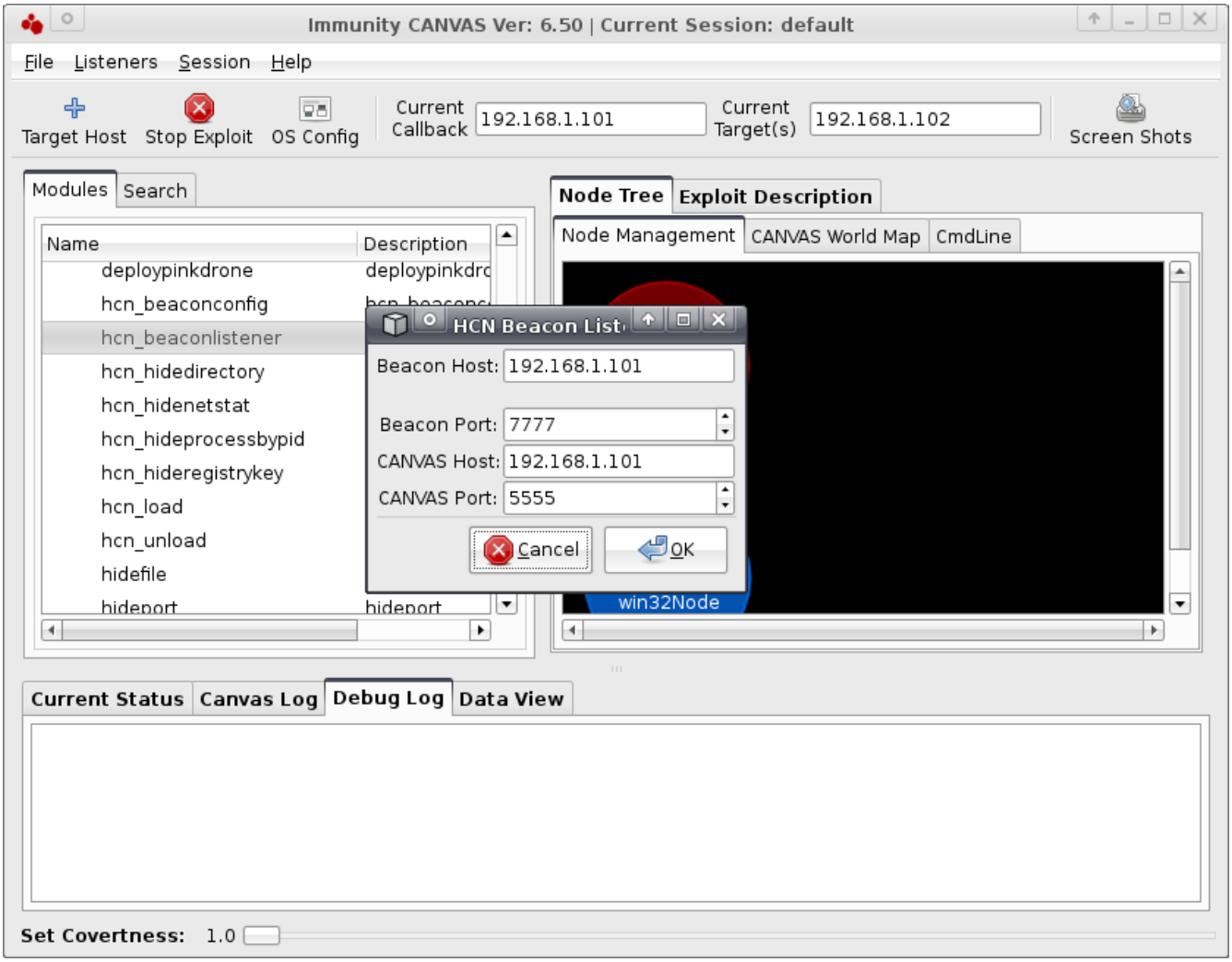
To set up the beaconing, ensure the compromised host with the rootkit installed is selected then double click the *hcn_beaconconfig* module. The beacon host field accepts the IP address of the host you wish to receive the rootkit connection from the compromised host, in this example I supply the IP address of the host I'm currently running CANVAS from. The port field is the TCP port you wish to be connected to on. The timer field dictates how often the compromised host will attempt to connect to you in seconds.



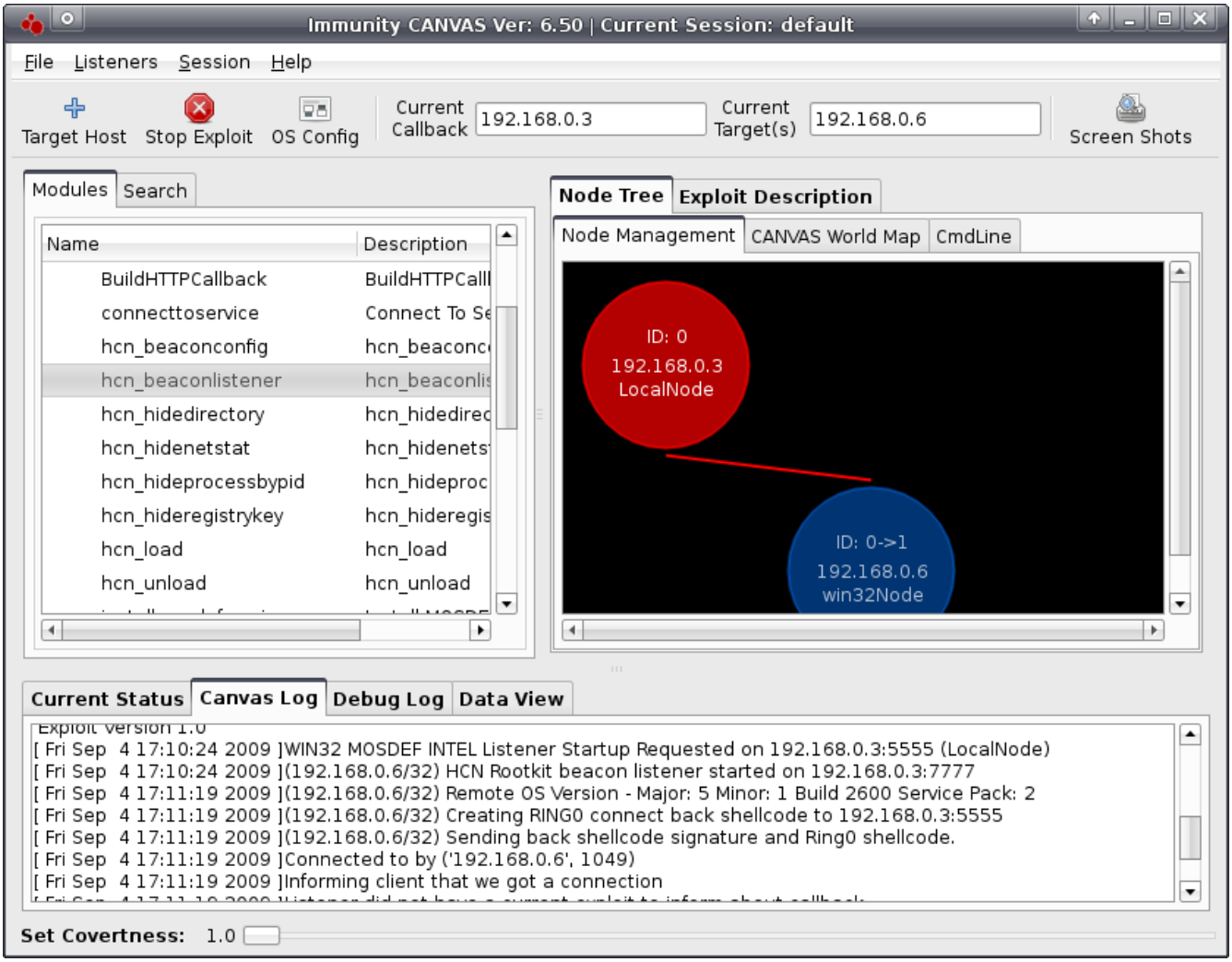
As an example, I've closed my compromised win32 node, now CANVAS can preform no actions on that host (I.e. Post exploit commands can not be run because I am no longer connected to the node). This simulates a common situation where you close CANVAS and need to come back to it later. But since we already configured the beaconing on this host, I can start my beacon listener and await a connection from my already compromised host!



Now I want a connection to my already compromised host. To get one I have to tell CANVAS to expect a specific type of connection, to do this I double click on the *hcn_beaconlistener* module. It takes the following options:



The beacon host and beacon port options, should be the same as what you set in the hcn_beaconconfig module. When the rootkit connects to the beacon port, CANVAS will respond by sending specialized ring0 shellcode, that will establish the real node connection back to CANVAS. The CANVAS host and CANVAS port determine where the node connection will happen. This allows you to have your beacon listener on one network, but have the actual node connection be on another network. Now all that's left is to wait for the timer on the compromised host to complete and for the host to attempt a connection. Once the connection is received. and the shellcode is executed, I get a standard win32 MOSDEF shell for the effort!



Now I've got my win32 node back and I can take actions as I see fit!