

Know Your Enemy: Honeywall CDROM Eeyore

Bootable GenII Honeynet Gateway

[Honeynet Project](#)

<http://www.honey.net.org>

Last Modified: 07 May, 2004

The [Honeywall CDROM Eeyore](#) is a bootable CD that contains all of the tools necessary to create and run a second generation honeywall. The CDROM is based on a trimmed down version of a Linux and is designed to be used as an appliance: it contains only the tools necessary to operate the Honeywall. A customization environment allows advanced users to add features to the CDROM which make it possible to use the tools on the CDROM in situations outside the honeynet. The first version of the CDROM, *Eeyore*, is a beta release.

This paper will serve as an introduction to the concepts and design of the Honeywall CDROM Eeyore. Later papers will cover more advanced topics, including data analysis, customization, and distributed capabilities. We consider this paper a "living document;" it will change along with CDROM itself to include the most current information. It is assumed that the reader has read and understood the basics of honeynets as described in [KYE: Honeynets](#) and [KYE: GenII Honeynets](#). In addition, the CDROM itself contains more detailed technical documentation on the system's user interface, internal layout, kernel configuration, and customization features.

Goals

The Honeywall CDROM Eeyore reduces many of the challenges of deploying honeynets while creating a platform for more advanced deployments. The CDROM is a highly minimized Linux LiveCD with very specific functionality. It is designed to function more like an appliance than a standalone operating system. The gateway created by the CDROM incorporates only the tools necessary for that system to function. The customization features (discussed later) allow new CDROM images to be created with additional or modified features.

The Honeywall CDROM Eeyore is designed to be:

- **Easy to Deploy and Maintain**

Historically, a properly configured honeynet required a great deal of time to install, configure, and test. Honeynet gateways must incorporate a number of different tools to provide data control, data capture, and automated alerting. Each of these tools needs to be downloaded, compiled, installed, and their unique configuration file modified to the honeynet environment. Once done, all of this needs to be tested to ensure that all of the tools functioned smoothly together. The Honeywall CDROM Eeyore comes with all of the tools needed to run a honeywall

pre-built and ready to go. The system can be configured and maintained using a simple user interface. For even faster deployments, the CDROM can load an existing configuration variables from a file on floppy disk. Once the system is set up it can be managed remotely by SSH running on an administrative interface.

- **Customizable**

There is no such thing as a "typical" honeynet deployment. A honeynet can contain different kinds of honeypots, in different numbers, set up many different ways. The CDROM brings together many of the most common tools used in honeynets, but may lack functionality for specific honeynet environments. Most of the tools that run the Honeywall reside on the CDROM itself, making it very difficult to add new features once a CDROM is burned. A customization environment has been developed to allow users to add and modify features in the ISO image *prior* to burning it to a CDROM. Using the customization tools, new ISO's can be created quickly and easily.

- **A Network Security Platform**

Honeywall tools such as the Snort IDS, snort-inline IPS, and iptables bridging firewall can be used on production networks to defend end systems. The flexibility and customization of the CDROM make it possible to use it for things besides a classic honeynet. Several different uses of the CDROM in production environments are discussed later in this paper.

Structure of the Honeywall CDROM Eeyore

The Honeywall CDROM Eeyore is a Linux LiveCD based on the [FIRE forensics CD](#) by William Salusky. We selected this base as it is built on a very slimmed down version of Linux, has a focus on forensics (data capture and analysis) and several members were already familiar with it. We customized the CDROM, including only the tools and their associated libraries needed to perform the Honeywall functionality. The CDROM has a small footprint of just over fifty megabytes. The [CDROM license](#) is based on a combination of GPL and BSD.

Bootable Linux CDs are typically used either for system installation, such as the [Debian](#) installer or to run an OS without modifying the host's hard drive, such as [Knoppix](#). In the first case, the contents of the CD are copied onto the host's hard drive. Once the copying has taken place, the CD can be removed and the system booted directly from the hardware. In the second case, all files remain on the host CD and a temporary filesystem is created in the host machine's RAM. The temporary filesystem is necessary to allow programs that need to create files in order to run properly (such as vi's temporary files or lock files used by some daemons). Since the files can't be written back to the write-only CDROM, the operating system loaded by the CDROM uses the pseudo-filesystem in the host's RAM. New or modified files appear to exist in the filesystem on the CDROM but are really stored in RAM. When the host is rebooted these changes are lost.

As an appliance, the Honeywall CDROM Eeyore combines both of these types of functionality. When booted, the CDROM loads most of the operating system into RAM, including all binaries and libraries located in the root file system (e.g. `/usr`, `/lib`, and so forth). These files are not copied to the hard drive and any changes to them are lost when the system reboots. There is a great deal of data you will want to remain persistent between system reboots, including system configuration files and logs. If all of this was loaded in RAM everytime, it would be lost upon reboot. The CDROM will take over your computer's harddrive to store all persistent data in a directory named `/hw`. This directory contains sub-

directories for the contents of `/home`, `/etc`, `/var` and `/conf` which contains configuration variables used by the honeywall. Symbolic links from the RAM filesystem are used to make these directories appear in the normal root directory.

This approach allows the host's hard drive to store more log data while minimizing the chances that important tools on the CDROM can accidentally be modified or removed. Based on the requirements for the RAM filesystem and the memory-intensive applications it runs, you must use a system with at least 256MB of RAM. We recommend starting with at least a 30GB harddrive, although the necessary hard drive size will vary depending on the network the system is being deployed on and frequency with which logs are erased or stored elsewhere. Currently only IDE drives are supported, there is no SCSI support. The CDROM will wipe the entire drive when initializing, so be sure there is no valuable data on it. The speed of the processor required by the Honeywall system is also dependent on the amount of traffic on the network. A minimum of a Pentium III class system (or equivalent) is suggested. At least two network interfaces are required for use as a gateway and an optional third interface can be used for remote administrative access. The CDROM currently supports only 3Com 3c59x and Intel eepro100 interfaces. You can read more about system requirements at [Honeywall CDROM Eeyore site](#)

Onboard Functionality

The first time the CDROM is booted on a host system, the operator is presented with the Honeywall's [user interface](#). The UI is based on a series of bash scripts that use `dialog` to create the interactive windows and menus. Almost all Honeywall configuration and administration functions can be done from the UI. An explanation of each menu option is displayed in the lower left hand corner of the screen when that option is highlighted. Users still have access to the full Linux command line environment and many of the standard GNU utilities.

Initial configuration of the Honeywall can be done entirely through the UI. In the Initial Setup the host machine's hard drive is formatted, persistent directories are created, network information is entered, and specific variables to handle data control, data capture, and automated alerting are set. A third interface can be configured for remote administration. Firewall rules can be set up to control what hosts/networks can connect to the administrative interface. Once the configuration has been completed the system is rebooted and the Honeywall is now active. We have included an [InitialSetup Document](#) to help you plan your installation ahead of time. An alternative to using the menu for configuration is to have a configuration file pre-built and accessible from a floppy.

Each of the variables set through the user interface is stored as an individual file in the `/hw/conf` directory on the host's hard drive. For example, the variable for whether your gateway runs as a layer two bridge or a layer three gateway is found in `/hw/conf/HwMODE`, while the email address used for alerts is found in `/hw/conf/HwALERT_EMAIL`. These variables are used by scripts on the Honeywall to configure each of the tools needed to run the system. The scripts must be reloaded for a change to one of their variables to take affect. The CDROM menu has the option to export all the variables to a single file called `honeywall.conf` in `/etc`. This file is not used by the honeywall itself, it is only a means to transport configuration files from one system to another or to create pre-built config files. The user interface has utilities for both exporting and importing the `honeywall.conf` file on a running system. The configuration can also be imported from a floppy during Initial Setup.

Data Control

The Honeywall CDROM Eeyore includes two primary methods for data control: rate-limiting of outbound connections based on iptables and the snort-inline intrusion prevention system. The rate-limiting firewall, which is based on the [rc.firewall script](#) can be configured to set upper limits on the amount of data that can be sent from each honeypot per unit time. The limits are based on outbound connections in the case of TCP and the number of packets for UDP, ICMP, and Other (a catch-all category to detect if attackers use a different protocol, like IPV6 inside IPV4, to send data off the honeynet). The time for each limit can be set in units of seconds, minutes, hours, or days. Outbound packets that are allowed through the firewall are then passed to snort-inline. Snort-inline is an intrusion prevention system based on the Snort intrusion detection system and is maintained by Rob McMillen of the Honeynet Project. It is capable of detecting malicious traffic using modified Snort rules and taking action against that traffic. Snort-inline can be configured on the Honeywall to use three different default rule sets that can either drop, disable, or reject known attacks. The firewall script and snort-inline rules are located in the `/etc` directory. They can be easily modified to suit the particular honeynet needs and the changes will be written to the hard drive. Both take detailed logs that can be used for analysis and alerting.

Data Capture

A second requirement is capturing all of the attacker's activity while minimizing the chance of the attacker detecting it. Activity on the honeynet is logged by several different tools. First, the firewall logs all inbound and outbound connections to `/var/log/messages`. This is good for giving you a overview of what is going on. Second, by default a Snort process captures all network activity and full packet payloads on the internal network interface (by default eth1). This includes all Sebek activity, which is sent using UDP packets. Third, an additional Snort process listens on the internal interface and generates full and fast alerts. In addition, Snort-inline also generates logs when it detects and takes action against outbound activity. All Snort and snort-inline activity is logged to `/var/log/snort/$DAY`, where `$DAY` is the numerical value of that day. We have standardized on YearMonthDay, so data captured on July 13th, 2005 would be located in `/var/log/snort/20050713`. All of this is configured automatically by the CDROM.

The only thing that needs to be configured by hand is Sebek logging. Specifically, what port and IP Sebek clients are logging to, and if you want all Sebek packets logged by the firewall (Sebek can get quite chatty, filling up your firewall logs). Most likely you will want to log Sebek activity to firewall logs at first, and disable that logging after you have confirmed Sebek is working. Once the Honeywall has captured Sebek activity in the Snort pcap files, you have to manually extract the Sebek packets using the Sebek `sebeksniff` and `sbdump.pl` utilities on the CDROM. A more advanced data analysis tool is under development and will be included in future versions of the CDROM. Sebek client software will have to be manually installed on all the honeypots behind the gateway. For more on Sebek refer to the paper [KYE: Sebek](#).

Automated Alerting

An additional requirement is the ability to generate alerts when a honeypot has been compromised. Alerting is currently implemented using cleartext emails. Swatch, the Simple WATCHer, monitors `/var/log/messages` for all inbound and outbound activity. If an outbound connection is detected, an

alert is generated and email sent. The email addresses that alerts are sent to can be configured through the UI. If the Honeywall is run in layer two bridge mode (the default) then a third, management interface (eth2) must be used to send the alerts. When sending email alerts from a management interface, you must configure the Honeywall firewall to allow TCP/25 outbound from that interface. By default, Swatch will raise email alerts based on the following events:

- Outbound connections from the honeynet
- Outbound rate limits reached for a honeypot

The problem with this approach is its limited capabilities; it assumes attackers will only initiate outbound connections. More intelligent alerting could also be based on increased inbound connections, more than 10 packets in an established connection, or perhaps an inbound connection that has more than 1K bytes. We have also identified an issue in IPTables. In the current release it appears to fail to track state properly. This means the iptables outbound connection rules may be triggered by some TCP inbound connections. This results in iptables logging an outbound connection and a false alert being generated by Swatch. This has been logged as a bug and we are looking into it.

Host-Based Security

As with any software, security is always an issue. Several steps have been taken to secure the gateway itself from compromise. First, a restrictive firewall is set on the gateway that by default denies any traffic inbound or outbound to the gateway's management interface. Second, we have attempted to run processes in least privilege mode and chroot jails wherever possible. Third, the Honeywall CDROM Eeyore currently ships with three different Linux kernels.

1. 2.4.X PaX (default)
2. 2.4.X grsecurity HIGH 2.X
3. 2.4.X generic

The modified kernels provide more sophisticated access control and stack protection for processes running on the CDROM. We are currently working on building ProPolice into all system binaries and libraries. Last, we use `monit` to monitor all other processes, and restart them if they fail. Keep in mind this CDROM is a beta release, please report all bugs or issues to our [bug server](#). The best way to run a secure Honeywall CDROM Eeyore is to have 24 by 7 monitoring of the system.

Upgrading and Customization

The challenge with a bootable CDROM is that it is static. You get whatever comes with the CDROM. At times, you will have to change your Honeywall, either upgrade it or customize it for your environment. We have built the functionality for both into the CDROM. First is upgrading. You will have to upgrade your Honeywall as new functionality is added and vulnerabilities are addressed. We have attempted to make the upgrade as simple as possible. Upgrades should consist of nothing more than downloading the latest .iso image, burning the CDROM, then booting the new CDROM. All persistent files on the harddrive (such as configuration files and logs) should be left untouched, while the CDROM has all the latest binaries, libraries, and kernels. This makes the upgrade simple, as the latest and greatest files are all on the CDROM and then loaded into RAM on boot. However, at times, we may have to upgrade configuration files or startup scripts that reside on the harddrive, primarily in `/etc`, such as

Snort rule sets or a configuration file. This is a little more challenging. When this happens, upon boot-up the new CDROM checks all `/etc` files and does a MD5 checksum of all files against a known database of the latest files. If a file does not match, then it is first backed-up and then the (presumed) newer file is copied over. If you have any files in `/etc` you have modified and don't want automatically backed up, then add those files to `/etc/noupgrade`. You will be prompted before any upgrade, and all upgraded files are logged to `/var/log/upgrade`. While not perfect, this method works (did we mention this CDROM is a beta). [Note: There is a potential conflict with this upgrade method and customization. A more robust upgrade method exists on the customization site, which is being considered for integration into a future release.]

A second, and more powerful, capability is customization. Currently the CDROM allows you to configure your CDROM to your environment. This primarily means setting the variables, such as IP addresses, hostname, enabling Snort-Inline, etc. However, what if you want to go above and beyond that, such as adding SSH keys, modifying the boot process, adding a new kernel or binaries, changing a directory structure? Customization allows you to do just that. You download the generic Honeywall CDROM Eeyore .iso image, modify the image (including adding or removing files) then burn the CDROM. This allows you to create the exact image you want, customized for your environment. One of the ultimate goals for this is to enable organizations to pre-build their own .iso images for distributed environments. For example, perhaps you want to deploy a hundred honeynets in your organization. With the customization you could have one person create 100 customized CDROM's, all preconfigured and ready to go. These 100 .iso images could then be downloaded by their respective parties, burned to a CDROM, the booted ready to go. Customization is very powerful, letting you build a CDROM that fits your needs. You can learn more about customization at [Dave Dittrich's customization site](#).

Deployment Scenario

We designed the CDROM with a specific type of deployment in mind. The current release of the Honeywall CDROM Eeyore is designed to implement a standalone honeynet. This means much of the initial configuration, monitoring, and maintaining is done locally on the system. You can remotely administer the system with SSH, but the capabilities of the current UI can be somewhat limiting. The architecture of the Honeywall CDROM Eeyore is the same as outlined in the paper [KYE: GenII Honeynets](#). Specifically, the deployment seen in the [honeynet architecture](#). There you see the Honeywall gateway deployed as a layer two bridge separating a production network from the honeypot network. The default interface layout on the Honeywall is:

- eth0: The external interface facing the production network.
- eth1: The internal interface, facing the honeypots. This is the interface both Snort processes listen on and where Sebek packets are captured and logged.
- eth2: This interface is optional and used for remote administration. In a bridge deployment, this is the only interface that has an IP stack.

In addition to being used as a Honeywall, many of the tools used on the Honeywall CDROM Eeyore, such as the IPS, IDS, and firewall can be used to secure and defend production networks. The rapid deployment and configuration along with the customization features make the Honeywall suitable for non-honeynet use in the field. Examples include:

- **Network Defense:**

As the data control center of a honeynet, the Honeywall is designed to protect hosts on the outside of the honeynet from being attacked by compromised honeypots. The tools for data control can be used in the field to protect production systems from the outside world. The script used to create the Honeywall's rate-limiting firewall can be modified to take full advantage of iptables' huge set of rules and options. Snort-inline, the CDROM's IPS, can be used by itself or with the firewall to drop known inbound attacks. The IPS rules are based off a slightly modified form of the Snort IDS ruleset. New Snort signatures can be integrated quickly and easily to allow the Honeywall to block the latest attacks being used in the wild.

- **Real-Time Forensics:**

The data capture and data control functionality of the Honeywall is designed to provide administrators as much information as possible about the network traffic entering and leaving the honeynet. In the event of a possibly compromised production server, the Honeywall CDROM Eeyore can be quickly deployed between the host and the outside world to monitor activity on the host without having to take it offline for analysis. With the Honeywall in place, all traffic to the server can be monitored without the hacker realizing that she is being watched. The Snort IDS and Argus flow analyzer can be used to provide a high-level view of network traffic, and Snort in full capture mode can be used to store and view all packets for analysis. The Honeywall's IPS and firewall can be used to prevent other inbound attacks and to block outbound attacks sent from the production system.

- **Traffic Monitoring:**

The passive network components of the Honeywall can be used to monitor traffic on production systems. The Snort IDS, Argus monitor, and iptables firewall (in log-only mode) can gather statistics on network activity and monitor for suspicious traffic. The CDROM's inline capability makes it ideal for monitoring networks or hosts where the traffic cannot be monitored from a tap or spanning port. The Swatch monitor can be configured to alert administrators for any type of activity on the network that is logged by the Honeywall, such as Snort exploit response alerts.

Words of Warning

One of our concerns is that people who do not understand the technology or the issues involved may now be able to deploy this technology. If you have not yet taken the time to learn and understand what kind of trouble you can get in by running a honeynet, you are not ready to set up a honeynet. You may risk downstream liability from damage, may risk an attacker (who may know more about your network than you do!) discovering your honeynet and exploiting this knowledge, or you may violate people's privacy rights and risk prosecution or civil action under state or federal electronic privacy statutes.

The Honeywall CDROM Eeyore does not reduce the skills required to deploy and maintain a honeynet, it only reduces the time to configure one. Honeynet operators must have a high degree of skill in networking and security to understand and respond to the events on a honeynet. Honeynet technology is not perfect. The data control techniques used have known limitations, and there are ways to get around the data control measures we have put in place. Anyone deploying this CDROM needs to be aware of the risks involved. For a better understanding of these issues, refer to [KYE: Honeynets](#).

There have been several notable successes of honeynets in the field over the past few years. In each case the honeynet involved was monitored by highly skilled individuals with a thorough understanding

of network security and honeynet technology. One example is the work the [Georgia Tech team](#) has done. It is imperative that before any production deployment is undertaken, the individuals involved feel comfortable with the intricacies of TCP/IP networking, computer and network forensics, and have the time and resources necessary to monitor and respond to events on the network. If you are still trying to catch up and figure out your firewall or IDS sensor, this CDROM is not for you.

Conclusion

The first public release of the Honeywall CDROM Eeyore, Eeyore, is meant to provide the tools necessary to quickly configure and deploy a second generation honeynet. Members of the Honeynet Project and Honeynet Research Alliance are actively working on the Honeywall CDROM Eeyore and have several enhancements planned for the next release including:

- **New User Interface:** The current interface, created using the dialog utility, is limited. A new menu system based on curses is under development.
- **Secure Kernel:** We are looking to add more advanced security to the kernel using [SELinux](#). A strict set of policies will be used to help mitigate risk of a compromised Honeywall gateway.
- **Data Analysis:** We are developing GUI based tools to make analyzing the data the Honeywall CDROM collects much easier to evaluate. We also plan to implement automated reporting tools such as daily and weekly reports of the activity the honeynet has captured.
- **Distributed Capabilities:** The current Honeywall CDROM Eeyore is designed to create a standalone system. We are looking into distributed features that will make it easier to manage systems created using the CDROM and to centralize data from multiple deployments.

Since this is a beta, we encourage bug reports or patches. A [public bug server](#) has been set up just for the Honeywall CDROM Eeyore and other tools created by the Honeynet Project. We would also like to hear your thoughts on what can be added/removed from the current release. Questions and comments should be directed to project@honeynet.org The Honeywall CDROM Eeyore is in active development. Be sure to frequently check the Honeynet Project site for the latest information and updates.

The Honeynet Project