# Freenet Test Bed Design

**University of Hawaii at Manoa**

**Project Members**
Dr. Yingyei Dong
Todd Baumeister
Dwayne Yuen

**Author**
Todd Baumeister

**Date**
1/29/2011

# 1. Freenet Test Bed Architecture

The purpose of this document is to outline the architecture of a Freenet test bed. Freenet is an anonymous peer to peer network that is currently in use by the public, and there are concerns when experimenting with certain attacks on this network. This test bed will provide the same functionality as the current public Freenet network, but it will also allow researchers to run experiments on a Freenet in a controlled environment. Since the test bed is designed to run autonomously from the publicly used Freenet network, attacks that would compromise the content or availability of Freenet can be studied here. A few of the many benefits of using this Freenet test bed are list below and will be discussed in detail throughout this document.

- Centralized experiment execution
- Isolated Freenet network
- Complete control over all Freenet nodes
- Centralized data collection and management
- Offline data analysis
- Easily scale number of Freenet nodes available

This test bed was design to meet all of the requirements mentioned above and provide a centralized point of control for experiment execution and maintenance. An overview of the architecture can be found in figure 1. Section 2 describes the configuration requirements of a firewall and router. Section 3 describes the servers used for run the Freenet nodes, and section 4 is about the data analysis component.

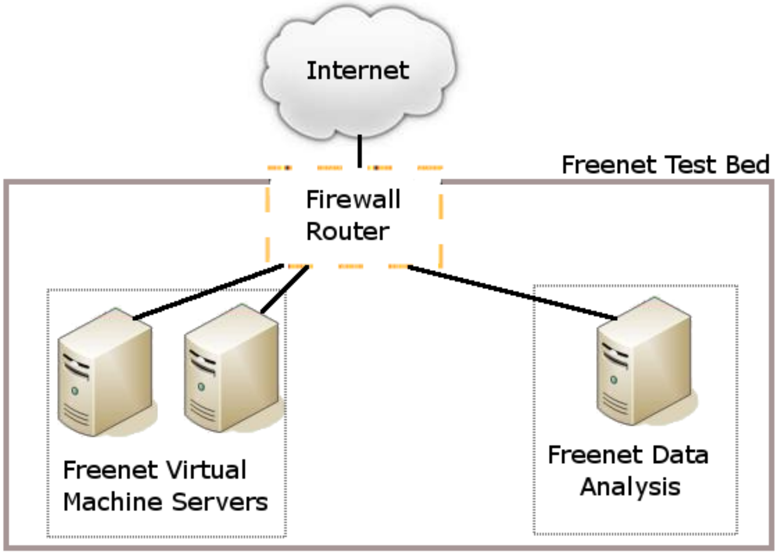

Figure 1: Freenet Test Bed Architecture

## 2. Firewall/Router

- Prevents all network traffic from entering or leaving
- Allows VPN access to connect multiple test beds together

The main purpose of the router and firewall in this Freenet test bed is to provide a barrier between our Freenet network and the public domain. This prevents any experiments carried out on the test bed from affecting the public Freenet network. The router can also be configured to allow VPN connections to be made. This will allow multiple Freenet test beds at remote locations to be interconnected. Doing this allows the number of Freenet nodes participating to be easily scaled. The router is also configured to run NAT, so the IP address space used by the test bed Freenet nodes can be controlled by the researcher.

## 3. Freenet Virtual Machine Servers

The Freenet virtual machine servers are used to host multiple Virtual Machines (VM). Each of these VMs will represent a Freenet node in the test bed network. An overview of the VM server configuration can be found in figure 2. Freenet Server 01 and 02 are two physical servers that are running the operating system VMware ESX. It is estimated that each of the servers will be able to operate 15 VMs. The current configuration has two servers hosting VMs to run Freenet nodes, but more servers can be added to increase the number of Freenet nodes. To centralize management, a script may be generated that can be used to start up and shut down all of the VMs on a VM server. A summary of the VM server configuration can be found in figure 3.

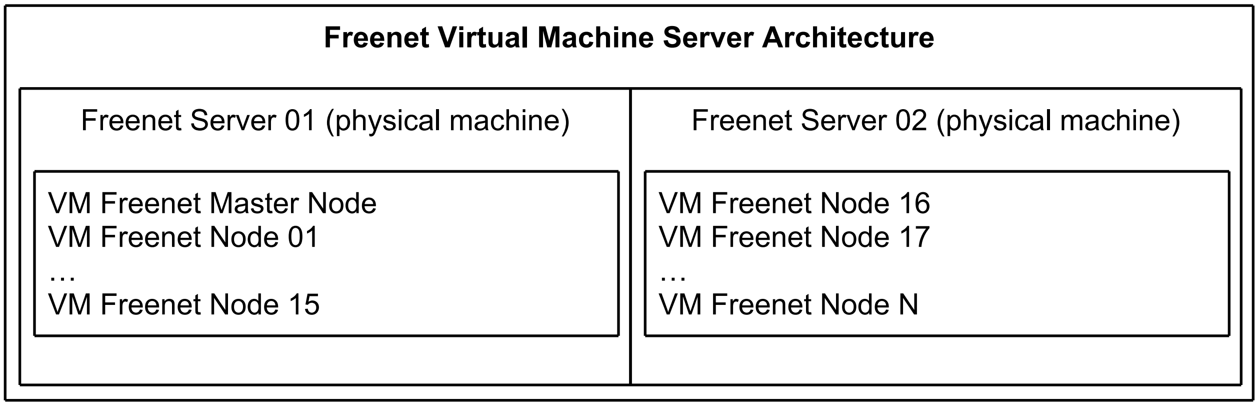
Freenet Virtual Machine Server Architecture
Freenet Server 01 (physical machine)
VM Freenet Master Node
VM Freenet Node 01
…
VM Freenet Node 15
Freenet Server 02 (physical machine)
VM Freenet Node 16
VM Freenet Node 17
…
VM Freenet Node N

Figure 2: Freenet Virtual Machine Servers

**Freenet VM Server Configuration**

Operating System: VMware ESX
# of Guest OS: 15
Machine Type: Physical server
Purpose: Host VMs acting as Freenet nodes

Figure 3: Freenet VM Server Configuration Summary

All of the virtual machines running on the Freenet VM servers will come from identical images except for one. The Freenet Master Node will be different from the rest of the VMs. This VM will be used for managing the other VMs in the experiments. A summary of the Freenet Master Node can be found in figure 5. Since the Freenet master node will be used for managing the other VM Freenet nodes, it will store scripts and data needed for those operations. The data stored on the Freenet Master node will be a master copy of the Freenet executable and configuration files. The master node will also have a management script on it used to control the other VMs. When the management script is run to start up Freenet on the VMs, the script will first check that the files in the Freenet folder on the VMs match the files on the master node. If any files do not match, the script will copy the file from the master node to the other node. In addition to the script being able to start Freenet on all of the VMs, it can also stop FreeNet. The master node may need to maintain multiple versions of the Freenet executable and configuration files if multiple versions of Freenet are going to run simultaneously in the test bed. It will need to be specified in the script which VMs map to which Freenet executable and configuration file version. An outline of the basic responsibilities of the management script can be found in figure 4.

**VM Freenet Master Node Management Script Responsibilities**

- Check if each Freenet node is running with the latest set of executable and configuration files
- Store master version of Freenet executable and configuration files
- Map a VM Freenet Node to a specific set of Freenet executable and configuration files
- Start up all of the Freenet nodes in the test bed
- Stop all of the Freenet node in the test bed

Figure 4: VM Freenet Master Node Management Script Responsibilities

All of the VM Freenet nodes will be running from identical images.The summary of their configuration can be found in figure 5. Each of these VMs will be used to run a single instance of FreeNet.The start up, shut down, and updating of Freenet on these VMs will be managed through the VM Freenet Master Node. Each of these nodes will be running a version of Freenet with custom logging enabled, and the log files produced will be stored on each VM's local disk. The log files will remain there until the Freenet Data Analysis Machine can collect the log files and properly store them. More details on the logging procedure can be found in section 4.

**VM Freenet Node Configuration**

Operating System: Ubuntu 10.10
RAM: 512Mb
Disk Space: 8Gb
IP: Static
Machine Type: Virtual Machine
Purpose: Freenet Node
Logging: Stores log files locally until retrieved by data analysis machine

Figure 5: VM Freenet Node Configuration

## 4. Freenet Data Analysis

The Freenet Data Analysis is a physical machine for centralized data collection, storage, and processing. All of the VMs running Freenet nodes run independent of each other, and they all store there own log files. The data analysis machine will be responsible for collecting all of the log files periodically and storing them permanently. When a log file is collected it will be removed from the VM running the Freenet node. The logging method on the Freenet node automatically archives versions of the log files after a set amount of time, so log files can be collected while an experiment is being run on the test bed.

The data analysis machine will be able to periodically collect log files from all of the different VMs. It will also be able to collect the log files at any given time when commanded. When log files are being collected, they need to be stored and identified with the particular experiment that is being run at that time. When log files are collected from the VMs, the log files are permanently stored on the data analysis machine and will be removed from the Freenet node VM. There will need to be different types of logging in the Freenet test bed, and each of those types of logging will have a separate data analysis process running on the Freenet Data Analysis Machine. This is done because the type of information collected from each of the logging types may vary greatly. When the data analysis machine is idle, it may be desirable to configure the machine to start processing the logged information. The processed information can then be stored along with the raw data, so it can be analyzed at a later time. An outline of the Freenet Data Analysis Machine's responsibilities can be found in figure 6. Depending on the amount of data being collected and the time to process it. The number of physical machines the make the data analysis component may need to be increased.

**Freenet Data Analysis Responsibilities**

- Collect log files from Freenet nodes
  - on an interval of X time
  - on command
- Separate and identify data from different experiment runs
- Permanently store Freenet node data
- Process data during idle time
- Multiple data analysis process running simultaneously for different data logging types

Figure 6: Freenet Data Analysis Responsibilites