# Global Information Assurance Certification Paper

**File Integrity Assessment Using FreeVeracity**
Jason Amsden
February 4, 2001

## Introduction

The purpose of this paper is to describe the use of the file integrity tool FreeVeracity as a part of an organization's potential strategy for "defense in depth". FreeVeracity is one of the few, if not only, free client-server distributed file integrity tools on the market today.

File integrity tools use various algorithms to detect additions, deletions and changes to file systems. The best time to install a file integrity tool is right after the initial installation of software, before the computer is brought online on any network. Using this aforementioned method, as long as the software media came from a secured source, the file integrity tool is assured of recording a pristine, untampered version of the operating environment for future comparison.

With all the products and technologies available today in the security industry, it is helpful to understand the context of where file integrity tools fit into the overall security space. Figure 1 illustrates the key layers of security an organization should be concerned with when creating a secure computing environment. Each defensive layer has concern with the elements of availability, confidentiality and integrity of data that require protection/control in an organization's security olution. FreeVeracity is a ool that provides integrity wareness and event anagement at the host efense layer. Integrity hecking in and of its self oes not create a secure ystem *(see referenced hrack article for details)*. reeVeracity should be sed in conjunction with ther tools and techniques s part of a comprehensive ecurity strategy to provide or the confidentiality, availability and integrity of computer environments at all defense levels.

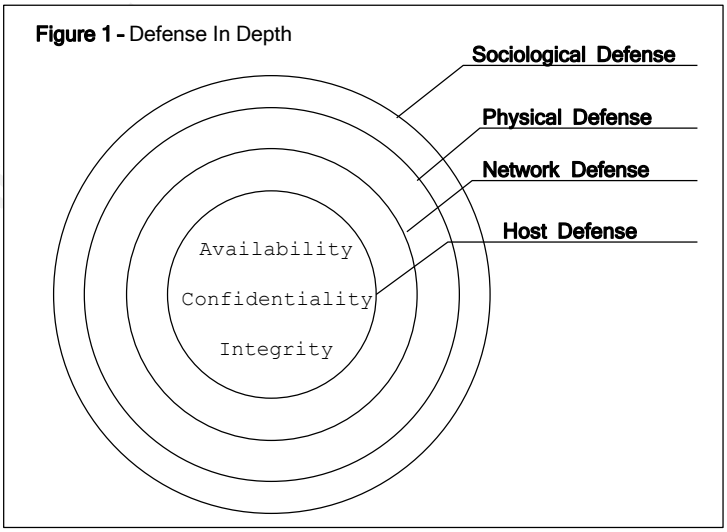

**Figure 1 -** Defense In Depth

## FreeVeracity Background

FreeVeracity can be obtained at www.freeveracity.com for operating system platforms that are free (such as Linux). For operating systems that are not free (such as Microsoft Windows) FreeVeracity's commercial arm (Rocksoft Pty Ltd), www.veracity.com, sells a licensed version of the product for non-free operating

system platforms. This paper will focus on FreeVeracity running on RedHat Linux, though the technology is applicable to other operating system platforms.

FreeVeracity can function on a stand-alone computer, but FreeVeracity is designed to run in a client-server model utilizing TCP port 1062 (assigned by the Internet Assigned Numbers Authority). Running FreeVeracity in a client-server configuration allows for centralized management of many machines as well as improved security by keeping key integrity information files on a remote (more secure) computer. Figure 2 illustrates a typical use of this technology. Users on the Internet have access to a server farm located in the DMZ via HTTP *(see Figure 2)*. A router access control list (ACL) or firewall could be used to limit Internet users to this service. If, however, an exploit is found in the http server or a script/executable accessible via http (tcp port 80) and a DMZ server is compromised, the individual would not be able to easily avoid detection if the exploit or unauthorized actions added/deleted/modified a FreeVeracity monitored file system. Another layer of security, a stateful firewall, separates the exploited server and the FreeVeracity Snapshot Collector. No pre-established inbound connections are allowed to the Secure Cell, so the unauthorized user is unable to "cover their tracks" on the exploited server. The next time the Snapshot Collector pulls a snapshot comparison of the exploited server any file change activities will be noticed and emailed/paged to the system administrator.

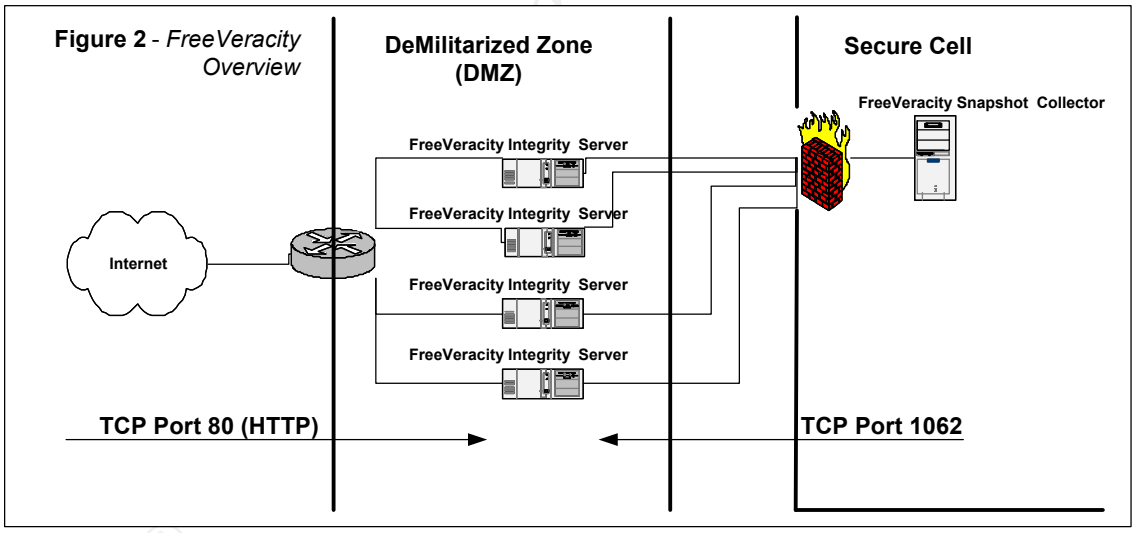


**Figure 2** - *FreeVeracity Overview*

When configured appropriately, a FreeVeracity integrity server listens on TCP port 1062 and serves a collection cryptographic hashes, known as snapshot files, based on defined portions of the integrity server file system. The directories and files included in the snapshot file are defined in a configuration file called a clipfile (see Appendix B for an example clipfile). The following sections will explain an example configuration of each key FreeVeracity component in a client-server configuration. This client-server configuration is designed to avoid one of the most common pitfalls when implementing a file integrity system; the storing of the file integrity information (snapshot) in an unsafe manner. Users intending to

implement FreeVeracity are encouraged to read the official documentation posted at www.freeveracity.com.

**Agent Configuration (a.k.a. FreeVeracity Integrity Server)**
The steps below outline what needs to be done to get a FreeVeracity Integrity Server in listening mode. While the agent is in listening mode, the FreeVeracity Collector has the ability to collect snapshot information in regards to the agent's file system.

| Task | Description |
|---|---|
| 1 | Download FreeVeracity to the agent computer.<br>http://www.freeveracity.com/download.shtml<br>Look for "**FreeVeracity V3.0 for Linux/386 V2.2.5 (.tar.gz)**" |
| 2 | Unpack the executable on the agent computer.<br>`tar -xzvf freev_v300_linux386.tar.gz` |
| 3 | Move the unpacked freeveracity folder to a permanent location (`/usr/local/bin`). |
| 4 | Make sure the permissions are set to only allow the root account read and execute permissions on the freeveracity folder and the contents within the freeveracity folder. |
| 5 | Make sure freeveracity can execute.<br>`/usr/local/bin/freeveracity/freeveracity`<br>A help message should appear after typing the command. |
| 6 | Generate a hashed password for server authentication.<br>`cd /usr/local/bin/freeveracity/`<br>`./freeveracity server hashpw [password here]`<br>Copy down the output of the password hash starting with "SHA-DES". |
| 7 | Create the Veracity configuration file in the freeveracity directory. (veracity_config.txt).<br>See Appendix A for sample configuration file contents. |
| 8 | The server is ready to be executed!<br>`cd /usr/local/bin/freeveracity/`<br>`./freeveracity server start [place hashed password here]` |
| 9 | The server is now in listening mode for the collector system to take a snapshot.<br>Optionally, Task #8 can be put in the rc.local file to start FreeVeracity automatically after the computer reboots. |

**Collector Configuration (a.k.a. FreeVeracity Client)**
The Collector configuration builds on the agent configuration in many ways. This part of the configuration introduces Snaplets. Snaplets are a set of perl scripts and libraries to help in the collection of snapshots from multiple computers. The tasks in the table below outline the setup process of the Collector computer.

| Task | Description |
|---|---|
| 1 | Download FreeVeracity to the agent computer.<br>http://www.freeveracity.com/download.shtml<br>Look for "**FreeVeracity Snaplets V2 (.tar.gz)**" |
| 2 | Unpack the scripts on the agent computer.<br>`tar -xzvf freev_snaplets_v2.tar.gz` |
| 3 | Move the unpacked vmonitor folder to a permanent location (`/usr/local/bin/freeveracity/vmonitor`). |
| 4 | Make sure the permissions are set to only allow the root account read and execute permissions on the vmonitor folder and the contents within the vmonitor folder. |
| 5 | Follow task steps 1-5 of the agent configuration on the collector computer to install the FreeVeracity executable. |
| 6 | Go to the vmonitor directory and read and follow the instructions contained in the snaplet.pl file. The compulsory parameters must be entered in this file (mainly path information, email address to send reports and a "keyring" password to hash collector passwords). |
| 7 | Execute snaplet.pl to enter the Snaplet shell.<br>`/usr/local/bin/freeveracity/vmonitor/snaplet.pl` |

| | |
|---|---|
| 8 | While in the Snaplet shell, type ? to see a list of Snaplet commands. Type CREATE [name of Snaplet] to get started and below are the expected results to create a Snaplet:<br>Enabled>Yes<br>Computer>[ip or host name of computer]<br>Serverid>root<br>Serverpw>[the non-hashed password used for the agent configuration]<br>Security>SHA-DES<br>Directory>/<br>Clipfile>/usr/local/bin/freeveracity/clip.vcf (see Appendix B for detail on clip.vcf file)<br>Options>None<br>Type>Baseline<br>Interval>1d<br>EmailA>[email address of person to receive report]<br>EmailB>[email address of person to receive report]<br>EmailC>[email address of person to receive report]<br>KeepBase>5<br>KeepHist>7<br>KeepReport>10<br>CHECK [name of Snaplet]<br>BASELINE [name of Snaplet]<br>CRON START<br>Quit<br>These variables are totally up to the administrator's discretion. Disk space should be considered when determining the number of baseline, history and report files to maintain on the system. Report types can be specified while entering the email information. |

After the Collector is installed, a Snaplet can be built for each server (repeating Collector task 8. A Baseline report compares snapshots taken against a base snapshot file. File changes will continue to be reported until the baseline snapshot file is updated. A Delta report does a "rolling" snapshot comparison, so relative file changes are only reported once. A combination of both delta and baseline reporting per host can provide early warning detection (frequent delta checks) and strong integrity assurance (baseline reports) on the monitored host.

### Conclusions

FreeVeracity is a powerful file integrity checker. System administrators will appreciate the centralized management capabilities of the client-server architecture and the tool's ease of use. Financial officers will appreciate the price of the software. Properly configured, FreeVeracity offers a strong guard for host data integrity.

### Appendix A – Server Configuration File

```
##############################################################################
# Veracity Server Configuration File
# ----------------------------------
# Name       : single_user
# Date       : 10-Aug-2000
# URL        : http://www.freeveracity.org/
# URL        : http://www.rocksoft.com/
# Purpose    : When the FreeVeracity software is invoked as a server, it reads
#              in a server configuration file such as this one, which provides
#              it with the parameter settings that define its behaviour.
# NoWarranty : To the extent permitted by law there is NO WARRANTY.
```

```
# Copyright  : Copyright (c) Trustus Pty Ltd 2000. All rights reserved.
#              However, you are permitted to copy, modify and distribute
#              this configuration file free of charge provided that these
#              notices are retained and all modifications are recorded
#              in the modification log below.
# Summary
# -------
# This configuration file instructs FreeVeracity to define a single virtual
# server for the invoking user. The server's name is the user's
# username, and its visible tree root is the user's home directory. The
# virtual server uses the highly-security SHA-DES security method, and
# you must provide the double hash of the password for the server as an
# argument when you invoke FreeVeracity as a server using this file. You do
# not need to be root to use this configuration file.
# Modification Log
# ----------------
# 10-Aug-2000 Embedded in Veracity as the "single_user" predef. config.
#
################################################################################
=====(BeginServerConfigV1)=====
   -----(BeginMasterServer)-----
   Arguments  HASHEDPASSWORD/I
   ListenIP   *
   ServerIP   *
   ClientIP   *
   Port       1062
   MaxProc    10
   MaxMemK    50000
   ServerDir  $D
   LogFile    $D.veracity_master.log
   PIDFile    $D.veracity.pid
   User       root
   Group      $B
   TimeoutMillisecs 30000
   CheckIn    No
   NeedsSuper No
   AdminEmail $A@$C
   -----(EndMasterServer)-----
   -----(BeginDefaultSettings)-----
   Enabled           Yes
   ServerIP          *
   ClientIP          *
   AllowData         No
   AllowFollow       No
   AllowFollowCycle  No
   TimeoutMillisecs  30000
   MinDelayMillisecs 0
   MaxDelayMillisecs 2000
   Domain            None
   DomainAlias       None
   Compression       *
   -----(EndDefaultSettings)-----
   # Define a single virtual server for the user.
   -----(BeginVirtualServer)-----
   ID          root
   Desc        The $V Server for $A ($D) on $C
   Security    SHA-DES:SHA-[MAKE SURE YOUR HASHED PASSWORD STRING IS HERE]
   VisibleTree /
   LogFile     None
   MasterLog   Yes
   User        root
   Group       $B
   AdminEmail  $A@$C
   -----(EndVirtualServer)-----
=====(EndServerConfig)=====
################################################################################
```

## Appendix B – Collector Clipfile

```
=====(BeginClipListV1)=====
"/etc/*" -f-> +B.md5
"/etc#*" -d-> +create +delete
"/bin#*" -f-> +create +delete +B.md5
"/usr/sbin#*" -f-> +create +delete +B.md5
"/boot#*" -f-> +create +delete +B.md5
"/lib#*" -f-> +create +delete +B.md5
"/usr/bin#*" -f-> +create +delete +B.md5
"/sbin#*" -f-> +create +delete +B.md5
"/usr/local/bin#*" -f-> +create +delete +B.md5
"/usr/local/sbin#*" -f-> +create +delete +B.md5
"/*" -d-> -all
"/&Overacty.v%f~" -f-> -all
=====(EndClipList)=====
```

## References


Nothcutt, Stephen, "Information Assurance Foundations", v1.41, SANS GIAC LevelOne Course Material

"What is the role of a file integrity checker like Tripwire in intrusion detection? ", http://www.sans.org/newlook/resources/IDFAQ/integrity_checker.htm (2/10/2001)

Rauch, Jeremy "Basic File Integrity Checking" 8/14/2000 http://www.securityfocus.com/frames/?focus=linux&content=/focus/linux/articles/fileinteg.html (2/10/2001)

halflife, "Bypassing Integrity Checking Systems", Volume 7, Issue 51 September 01, 1997, article 09 of 1, http://phrack.infonexus.com/search.phtml?view&article=p51-9 (2/10/2001)

"FreeVeracity Readme", http://www.freeveracity.com/readme.shtml (2/10/2001)

"FreeVeracity Reference Manual", Version 3.0c (10 August 2000) for FreeVeracity V3.0, http://www.freeveracity.com/reference/index.html (2/11/2001)

"Using MD5 to verify the integrity of file contents", March 10, 2000 http://www.cert.org/security-improvement/implementations/i002.01.html (2/10/2001)