Global Information Assurance Certification Paper

GIAC Security Essentials Practical

Thursday, November 09, 2000

Security and BeOS: The Media OS

A State of the Art Security Report on the Pre-Release of The BeOS Networking Environment (BONE)

INTRODUCTION AND THE BIG PICTURE

What makes "this" study of the BeOS unique?

What makes this study unique is two important characteristics. First, Be, Inc., has, for the purposes of this study, authorized me to procure a copy of their new pre-release of the BONE networking environment and install it over the V5.0 Pro vintage of their BeOS. Second, I've actually implemented everything in this report unless explicitly stated. So I write with the authority of reporting on a system, which has not been made public yet, and one, which I have hands on experience working with, analyzing and interrogating. Wherever possible I'll include URLs to third party software I tools used to augment this study.

The uniqueness of the BeOS architecture.

What makes the BeOS unique is that it is a media operating system. It has the user interface of the Apple, and the Internals of UNIX, without being Apple or UNIX. In geek speak, it is a object oriented, multithreaded, multitasking OS, with a 64Bit, fully journaled database-like filesystem. It is additionally, (almost) POSIX compliant, with a BASH shell, and runs on an Intel or Apple platform, using a unique RAM memory management scheme. However, it is NOT a multi-user system.

V5.0 Pro and Security.

The conversation of security with the V5.0 pro (Non-BONE) vintage is a very short one indeed. There is little or no security. It's a single user system, everything runs a root, and there are no inherent security mechanisms available. Any security must be brought in via 3rd party software or not at all. Security risks abound everywhere.

What is BONE anyway?

BONE stands for BeOS Networking Environment and it is a complete rewrite of any the previous OS networking environments existing in vintage BeOS releases. The BONE employs a modular design architecture, which permits easy removal or replacement of any of its individual parts, by users or by Be. Effectively, BONE is an API spec for a networking architecture in addition to a description of how those modules interoperate. It is said that, the implementation Be will ship can have parts replaced by users at will if they so desire, provided that they adhere to the

specification. In marketing speak, it places Be's OS on the map of legitimacy in the internetworking world.

BONE and Security.

Finally with the introduction of BONE, the notion of implementing a security scheme become more realistic and viable. This is due to the added features of the BONE environment supporting a security infrastructure. The importance of this is critical since those same features permit more powerful server software to run on the BeOS than previously, thus creating the need for greater security. It's still a single user system and that's a problem, but the BONE architecture does introduce a number of infrastructure features that security people really find useful. The next sections will address those items in detail.

ASPECTS OF THE OPERATING / NETWORKING SYSTEM AND SECURITY

The File System

- File Permissions and Attributes
  The BeOS employs a UNIX file system permissions scheme that does have in impact on system security. Although the BeOS is currently a single user system, the traditional user and group security permissions are implemented. This has an impact when using internet applications such as web servers as they can override or augment mechanisms such as Apache's '.htaccess' constructs.

- Visibility via the Tracker
  Depending on the various attributes set, a file/program may appear invisible to the "Tracker" – the BeOS version of Apples Finder mechanism. Invisible files and processes are always present a security concern, since hackers delight in remaining elusive. Performing integrity checks via the command line mitigates this concern, but requires work.

- File Level Security
  There are some things which can be done to brute force protect sensitive files on the BeOS system. They can be encrypted, using a standard encryption algorithm such as the combination of the BlowFish Translator and the Cryptic Software. They are easy to use and effective. However, it is a destructive translation process, which does not leave the source file untouched and then create a new additional encrypted one. The source file gets encrypted and decrypted in the same file, so be aware.

  - BlowFish Translator http://www.bebits.com/app/1449
  - Cryptic Software http://www.bebits.com/app/1450

The BeOS Kernel

- Single User vs Multi User
  The BeOS is currently a single user system, but there are enough aspects of a multi-user file system implemented that any application can take advantage of them. RobinHood (a Native BeOS Web Server) is a good example of this. It respects underlying file permissions and if a incoming web request asks for a piece of the file system that is does not have permissions for, a dialog box pops up requesting a username / password. It is also notable that the files at the system level are not writable with out changing their permissions. This can be done at a telnet prompt using the

standard Chown, Chmod, Chgrp conventions or graphically using the tool, SetPerms, which can be found at http://www.bebits.com/app/1293.

- Altering System Level Parameters
  There are some tracker and kernel level parameters which the security expert may be interested in changing. Rather than digging into the scripts and risking harm to the system via a scripting error, a useful GUI tool is available called, PrimalToys: http://www.bebits.com/app/1529.

The BeOS POSIX System

- Directory Structure
  Any Unix Guru will quickly find the differences between the BeOS POSIX directory structure and what they are traditionally used to. For instance, everything stems from the "/boot/" directory tree, including systems directories "/boot/beos/etc", development directories "/boot/develop", and application directories "boot/beos/apps". The construction of familiar directory constructs are done by SymLinks. This takes a bit of getting used to, but once that's over, it's easier navigating.

- Users and Groups
  Since the inherent file system has plenty of multi-user infrastructure in it, changes to the owners and groups can be made to files. If the owner of the machine decides to change some of their boot up scripts, they can force the system into changing the umask parameters of any new file created to reflect that users identity. To do so, they would have to edit the file /boot/home/config/boot/UserSetupEnvironment and modify the export statements at the bottom for User and Group, (even shell). This could present security (and stability) problems if used unwisely.

- Low Level System Functionality
  The BeOS is UNIX 'like' but it isn't UNIX. In Fact is isn't even fully POSIX compliant (it's mostly). This becomes a problem when UNIX and Network attacks work effectively against the BeOS but security tools used to perform network and host based security don't compile on the system because some low level libraries, functions or headers are not present. The only hope is that they will appear in mass when the final release of BONE is dispatched.

BeOS System Startup Sequence

- System Startup Directory / Files
  Unlike the runtime levels of Linux and similar to the startup of BSD, the BeOS uses a single set of scripts to start up the OS. These scripts are found in the directory "/boot/beos/system/boot" and are as follows:
  - The BootScript --This is the Key script. If this is corrupted, the OS won't load. Some apps, such as BeLogin will have you modify this script (and in this case you must), but in general the BeOS doc's tell you to stay away from it. It's a good idea!
  - The NetScript – This script defines how the IP network loads. This is interesting to peruse to get a handle on the internals of the networking environment.

- The SetupEnvironment Script -- This is one real key file to change if you are having ".profile" and environment variable problems. And you will if you try to do anything significant.

- User Startup Directory / Files
  Additionally, the BeOS has startup files for the user as well as the system. They are found in the directory "/boot/home/config/boot" and are as follows:
  - UserBootScript – the script which gets executed as the user logs onto the system (Very Useful when a third party logon procedure is in place).
  - UserSetupEnvironment – This script sets the environment variables and makes calls any .profile files you may have. Another useful script.

The Bash Shell

- Profile and Environment Problems
  An issue I found was that the BeOS has selected amnesia of system environment variables and profile values when you telnet in remotely, even if you have an existing account of the system. This may not be as much of a security hazard, as it is a irritant and a reminder to further investigate how the BeOS processes it's environment variables (which 'is' a security concern). To get around this problematic situation, I did a "env > export-my-environment" at the command line in my /boot/home/config/boot directory (while at the console) and added "export" to the beginning of all the lines, then when telneting in remotely, I did a 'source export-my-environment', and my environment variables were now available as if I was at the console. (It's cheap, but it works!)

Important Development Environments

- Perl 5 and GNU C/C++
  The BeOS does have a couple of important development environments, namely (Perl 5) http://www.bebits.com/app/658 , and the (GNU C/C++) environments. This permits the compilation from source code various POSIX compliant applications that lend security experts (or hackers) could find useful. I was able to compile and install the Apache web server using these. This is important when considering porting third party software to the BeOS.

The BONE Networking Environment

These tools do not exist in the Non-BONE BeOS vintages, and really form the basis of true networking and security defense implementations. Although they are found in some strange places in the file system, at the writing of this document the following networking tools were in existence and useable:

- Inetd, /etc/Inetd.Conf
- TcpDump, TcpTrace, and LibPCap Facilities
- SysLogd, and /var/log/syslog
- Ping, IfConfig, and TraceRoute Facilities
- FTPd and Telnetd

- DHCP and PPP
- NSLookup: (Non-BONE): nls http://www.geocities.com/bemisfit/

The final release of BONE is promised to have all of your favorite UNIX networking utilities either already ported or will port readily. Which means if Be didn't provide it, it should be able to be ported, compiled and installed easily. An example of such is cron.

Some items specifically targeted are:

- BIND 8.2 tools: -- addr, dnsquery, irpd, named-bootconf, nslookup, dig, host, mkservdb, named-xfer, nsupdate, dnskeygen, named, ndc
- Configuration Tools: -- route, ifconfig, netstat etc.
- Utilities: -- telnet, ping, ftp, traceroute, tcpdump, libpcap
- Sockets
  - sockets are file descriptors
  - the sockets API is much more compliant
  - raw sockets are there
- It's relatively easy to add new protocols.
- There is a kernel networking interface.
- Much, much more….. ;-)

Console Based Security

The out of the box BeOS has 'no' console security, it's always on, and the only user is effectively root. Securing the console is paramount. Fortunately there is a useful program to do just this called "BeLogin" and it can be found at: http://www.bebits.com/app/916. What this application does is to replace the functionality which automatically invokes the deskbar and tracker with it's own application that prevents them (and thus the console interface) from being launched until a valid username / password are provided. As an added feature, it will permit multiple usernames and passwords, which is good. It will have you edit the sensitive system BootScript, but it's pretty easy, and the application really does the job. I believe it uses a basic cryptographic hash to store passwords as well, it's not done in clear text.

Host Based Security Facilities

The BeOS needs some host based security features or the capability to reliably port them over from other similar systems. The following applications either existed as part of the BONE installation or I found on the website http://www.bebits.com .

- Inetd and TcpWrappers
- Logging Services – Syslogd, "/var/log/syslogs"

Arguably, 'the' most concerned set of Internet processes is the "Inetd" and "Inetd.conf" files. In addition, 'the' most important security application is "TCP Wrappers", which works hand in hand with the Inetd system. TCP Wrappers can provide decent protection against the most common "Script Kiddie" attacks. TCP Wrappers permits valuable security to be implemented on a few levels:

- It doesn't automatically permit the connection, instead it tests the connection for validity first, if it passes, then it's permitted to connect.

- It permits the opportunity to secure the system by turning off selective network processes which are unnecessary or vulnerable (in the inetd.conf file), a big plus!
- It permits the existence of a 'hosts.allow' and 'hosts.deny' file, two keys to host based security, where we can establish policies at the domain or IP level of who is permitted to have remote access to the machine and who is not.
- It allows the FTP daemon to be fortified via an 'FTPUsers' file, effectively disallowing anonymous and guest logins (a big security problem).
- It permits the fortification of the Telnetd daemon via the –h and –U options, which effectively disallows the daemon from giving out any 'extra' information about the operating system, and requires that incoming IP address be resolvable before it will permit connection (helping to protect against IP spoofing).
- It allows logging of the FTP daemon (with the –l option) and Telnet daemons to /var/log/syslog – this is a big key to intrusion detection and host base integrity checking.

The two lines from my Inetd.Conf file which help fortify Telnetd and FTPd are as follows:

```
ftp       stream  tcp     nowait  root    /bin/ftpd       ftpd -l -l
telnet    stream  tcp     nowait  root    /bin/telnetd    telnetd -h -U
```

NOTE: It is difficult and dangerous to try to get Inetd to run with logging and TCP Wrappers by altering the System BootScript. A much safer and better choice is to alter the UserBootScript in the /boot/home/config/boot directory and add the following lines to the bottom of the script. These lines will kill the existing Inetd Daemon and run it again using logging and TCP Wrappers.

```
#
# Kill the existing inetd daemon and restart it using tcp wrappers
# (Yes, the PID is found in /var/tmp) go figure!?
#
kill -9 $(cat /var/tmp/inetd.pid)
/etc/net.d/inetd -l -w -W /etc/inetd.conf &
#
#
# Start Apache
#
/boot/home/apache/bin/apachectl start
#
```

Other Host Based security mechanism I observed were the existence of a third party Proxy Server and a third party Natd daemon. They were constructed for the BeOS, and installed easily, however they were for the R4.5 vintages and I couldn't get them to work properly under the BONE environment. I also detected an absence of any firewall implementations in the BeOS archives. It is essential that the development community fortify this area of security for the BeOS platform for it to become a viable server in the future.

- Proxy Servers http://www.bebits.com/app/1006
- Natd http://www.rickb.com/software/bin/BeEarth1.5x86.zip
- Firewalls / Packet Filters – None available for BeOS to date.

Network Based Security Facilities

- TcpDump, TcpTrace, and LibPCap Facilities

The most significant low level capabilities I observed were the tcpdump, tcptrace, and libpcap infrastructure. They are fundamental to all of the network based intrusion detection, and flexible response systems I have seen in the security field. They are also key for the construction of a suitable packet filtering / firewall implementation. The current implementation with BONE does work as expected. This is REALLY good!

- Snort and Shadow

Snort and Shadow are both intrusion detection systems, snort is also a flexible response system, with lot's of capabilities. I tried extensively to get both to compile on the BeOS with BONE and only could get someplace with SNORT (with the help of Snort's Author, **Martin Roesch**, of course). We haven't gotten short to fully deploy, but the core 'C' module is compiling at the time of this writing. This is good at this stage and if we can get it to work properly it would be a VERY significant event for the BeOS community in terms of host and network based security. And it appears from all angles, that the BeOS will require this kind of protection in the future.

- SSH Client: http://www.rickb.com/software/bin/ssh.zip
- SSH Daemon: -- None to Date.

Another key issue in network security is the ability to do encrypted remote management of hosts. Currently the BeOS does have a SSH client which works as expected. However, a ported SSH daemon (server) would not compile due to not having a certain number of low level UNIX network socket functions implemented. Let's hope the BONE folks include those in the final release. This is a critical feature as well as it prevents passwords from being sniffed across the network while remotely managing the server.

BeOS Internet Servers

- FTP and Telnet Servers – The keys to remote management. They are single users only, and can be fortified adequately as discussed previously.

- Web Servers
  - RobinHood
    This is a native BeOS web server which is 100% written for the BeOS from the ground up. It's fast, lightweight, reliable and complies with the majority of web server specifications. It uses the native file system (BFS) permissions to implement password protected areas as well as it's own 'Relm' based security, which is a bonus. It's a decent application and in the absence of Apache should be used.

  - The Apache Web Server (thanks **David Reid**)
    Only the latest Apache server would compile and deploy V 1.3.15. There are some peculiarities in the way it was handling images, but in general it worked fine. The compiling was a bit difficult due to the fact that the make module doesn't fully understand the BeOS directory structure yet, but with the judicious application of symlinks to network libraries and header files, it compiles just fine. And it's Apache! ;-)

  - http://dev.apache.org/from-cvs/apache-1.3/
    - V 1.3.15 for the BeOS Platform

SSL Encryption

It was indicated that Apache would compile and run both Mod_SSL and Open_SSL but I wasn't able to compile them. I suspect they will just fine in time but for now I do not have proof that they actually do on the BONE environment.

- Open_SSL http://www.bebits.com/app/1020
- Mod_SSL (from Standard Web Resources)

An important advent is the porting of PHP to the BeOS, which ushers in the ability to construct database applications, and dynamic scripting with the Apache server. This is important because it indicates the emergence of sophisticated web developing environments appearing on the BeOS landscape and that spells the need for powerful security applications and processes to protect them. CGI in general poses many well documented security problems.

- PHP for BeOS http://www.bebits.com/app/566

Misc Important Third Party Software Applications

Security concerns for an operating system include basic maintenance utilities like the following:

- Cron: Scheduler http://www.bebits.com/app/1085
- Backup: SigFriedBackup http://www.bebits.com/app/88
- System Process Management: TaskManager http://www.bebits.com/app/432
- Monitoring Ram and HD Space: SpaceMonitor http://www.bebits.com/app/497

Having the ability to backup the system is critical and having the ability to schedule them at discrete time intervals is necessary as well. The two applications, SigFriedBackup, and Scheduler perform those tasks nicely, however they have difficulty being invoked from the UserBootScript as the instructions indicate (in BONE). This problem has not been overcome just yet, so reboots are a concern. The scheduler and Space Monitor applications instruct you to alter the UserBootScript in the /boot/home/config/boot directory and add the following lines to the bottom of the script. This doe not work in BONE upon a reboot, but oddly enough, it will work on the command line. (FYI, the problem could very well be me.)

```
# Start the Scheduler
#
# BEGIN Install-x-vnd.BT.Scheduler
/boot/home/Downloads/Scheduler/Scheduler -background &
# END Install-x-vnd.BT.Scheduler
#
#
# Invoke Space Monitor
#
/boot/home/Downloads/SpaceMonitor-1.3.1/SpaceMonitor &
#
```

The last two items in that list were a Process Manager, which is absolutely essential for killing run away processes and a Resource Monitor which displays Ram and HD space

increments. Both programs are easy to install and a big help. I highly encourage you to use them..

SECURITY HOT SPOTS

The following is a quick list of security hot spots which exist in the BeOS currently. These either have to be adequately addressed or avoided all together. This is by no means a comprehensive list.

- Single User System -- this is a problem but it's not the end of the world. If a hacker gets a shell account on any other box, it's a foregone conclusion that they'll get root eventually. I think it's more important for BONE to put host based firewalling or intrusion detection / reaction mechanisms in place in the mean time.

- V5.0 Vintages w/out Bone Infrastructure and OS Hardening – This OS is vulnerable, plain and simple. The more features it has, the worse the security situation will be. Put it behind a firewall!

- Samba – This is a favorite among hackers, they hone in on Samba protocols almost a quickly as they hone in on NFS installations. Unless you have proper security in place, the box will eventually get 'rooted'.

- NFS – Don't use it (from a paranoid security point of view), it's got a plenty of security concerns.

- Mail – This is traditionally a vulnerable area, but I'm not sure the extent of which it is with the BeOS.

- Native Telnet and FTP Services – These can be hardened with TCP Wrappers, Hosts.allow, Hosts.deny, and FTPUsers files. Use them or get hacked.

    - **Problem**: The out of the box BONE installation permits anonymous FTP's, fix this!
    - **More Problems**: The out of the box installation of NON-BONE BeOS does not have inetd or TCP Wrappers, so NEVER permit the Telnetd or FTPd servers to run unattended. You have been warned.

- Un-hardened console access – Out of the box BeOS (any vintage) has NO console security features. You must get BeLogin and install it properly for adequate console security.

- Web Server Buffer Over Flow Vulnerabilities – these are the most reported network intrusions reported for the BeOS on BugTrac. Use Apache and incorporate the recommendations they give you to secure your Apache installation. It's a good start!

- Standard Battery of Web Server CGI Problems – see above conversation. Also, be sure your scripts are written well. Check the Cert Security Advisories for that particular (or like) scripts vulnerabilities. And check for anomalies in your systems behavior. Strange things are BAD!

- Firewalls – non existing yet. The development community needs to create one in a BIG

way for this OS. It needs one!

- Instrusion Detection Systems – almost but not yet. Pray that Martin Roesch has the time and can get snort working on the BeOS BONE platform, if so it'll be a big advance for security on the BeOS platform.

- BeOS based Viruses --- None to date, but only a matter of time. Microsoft is still a big target.

- Name Service – Future releases of BONE will have name services capabilities. The security implications of the named daemon are large, and will demand measures be taken to ensure the plethora of possible attacks are dealt with.

REFERENCES AND CREDITS

Credits:

* A special thanks goes out to the good folks at Be, Inc., particularly Howard Berkey, for making an exception to allow me to participate in their BONE beta testers program.

* A special thanks goes out Kurt von Finck, of Gobe Software for guidance and for taking the time to make the Be, Inc., introductions for me.

* A special thanks goes out to David Reid for getting me a version of Apache, which actually compile and ran on the BeOS platform. And for the pointers on how to do so.

* A special thanks goes out to Martin Roesch, author of Snort, for helping me get a version of Snort a bit further to running on the BeOS Platform.

* A final thanks goes out to the good folks on the Be mailing lists who responded to my request for information: Peter Schultz, David Aquilina, Guildencrantz, Titus, Scot Hacker, Ryan Christiansen, and David Muszynski.

References:

Pelaprat, Etienne, BE Newsletter, Interview: Howard Berkey,
http://www.beoscentral.com/feature/feature.php?page_number=1&feature_id=11,
May 18, 2000 @ 12:58

Berkely, Howard, BE ENGINEERING INSIGHTS:
The BeOS Networking Environment, Volume IV, Issue 5;
http://www-classic.be.com/aboutbe/benewsletter/volume_IV/Issue05.html#Insight,
February 2, 2000,

Hacker, Scott, :Robin Hood: Password-protecting web pages", BeOS Tip Server: Applications,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip574

Hacker, Scott, : Working with Windows and Samba networks", BeOS Tip Server: Networking,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip367

Hacker, Scott, : Maintain web sites remotely", BeOS Tip Server: Networking,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip393

Hacker, Scott, : CGI basics”,
BeOS Tip Server: Scripting,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip457

Hacker, Scott, : RobinHood: Understanding the DirectoryHandler”, BeOS Tip Server: Networking,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip457

Schomer, Michael, : Writing to attributes from perl scripts”, BeOS Tip Server: Scripting,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip445

Takayuki, ITO, : WON passwords aren't secure”, BeOS Tip Server: Warnings,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip408

Hart, John, : Printing with cifsmount”, BeOS Tip Server: Networking,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip371

Schinckel, Matthew, : Custom control over attributes”, BeOS Tip Server: Tracker & Deskbar,
http://www.Betips.net/cgi-bin/chunga.pl?ID=tip070

Burgess, Richard; NAT (Network Address Translation) for BeOS version 1.5.1,
http://www.rickb.com/software/nat.htm

Ylonen, Tatu, SSH - Secure Shell Client (remote login program) for BeOS,
http://www.rickb.com/software/ssh.html.