

the security journal



- *Technical Surveillance*
- *Genntoo*
- *Complex Security*
- *Change Management*

Volume 24 - Summer 2008 Edition



Your global information security experts

5350 Tomah Drive
Suite 3200
Colorado Springs, CO
80918

<http://www.securityhorizon.com>

Editor in Chief: Russ Rogers
Assistant Editor: Michele Fincher

The Security Journal is always looking for quality writers for this publication. If you're interested in submitting an article for an upcoming edition, please contact us at editor@securityhorizon.com.

Interested in advertising in The Security Journal? Contact us at editor@securityhorizon.com

Special thanks go out to **Ping Look** from **Look Designs** for the creation of the Security Horizon logo and masthead images for *The Security Journal*.

For more information on Look Designs:
www.republicofping.com
design@republicofping.com

For more information on Security Horizon, please visit our web site or email us at: info@securityhorizon.com.

Security Horizon, Inc. was founded in 2000 and is headquartered in Colorado Springs, Colorado. As a company, we believe that sound security services are based on education, experience, standards, ethics and unquestionable integrity.

Table of Contents

Notes from the Editor
russ rogers.....p. 3

IAM Course Schedule.....p. 4

Technical Surveillance Countermeasures, Why & What
ed fuller.....p. 5

Creating Your Own LiveDVD with Gentoo
paul criscuolo.....p. 9

Thought for the Day: Complexity is the Enemy of Security
greg miles.....p. 17

The Need for Change Management for Network Design in Small to Medium Businesses
brian hitch.....p. 20

IEM Course Schedule.....p. 25



Copyright ©2008 by Security Horizon, Inc.
All rights reserved. Reproduction without permission is prohibited.



Black Hat[®]

Briefings & Training

World Tour

Black Hat Briefings & Training USA

Caesars Palace Las Vegas • Nevada

August 2-7, 2008

Registration now open.

Black Hat Briefings & Training Japan

Keio Plaza Hotel • Tokyo

October 14-17, 2008

Call for Papers and Registration Opens July 1.

Black Hat Briefings & Training DC

Hyatt Regency Crystal City • Arlington, Virginia

Feb 16-19, 2009

Call for Papers and Registration Opens December 1.

Black Hat Briefings & Training Europe

Movenpick Amsterdam City Centre • the Netherlands

April 14-17, 2009

Call for Papers and Registration Opens January 1.

WWW.BLACKHAT.COM

Notes From the Editor



Russ Rogers
CISSP
Editor in Chief

Keep a Grip on that Hand Held



When people consider software vulnerabilities, they most often think of the operating system or the applications that they have loaded on their workstations or servers. A lot of time and energy has been invested in these types of software to help ensure their security, be it in the form of patches, software updates, or a change in vendor. But the world isn't really so simple anymore.

None will argue that technology continues to advance at an alarming rate. Advances in hardware technology now make it easier to carry our connectivity with us wherever we travel. Whether you're in China, Brazil, or the United States, you have the capability to access the web, your personal and professional email, and your contacts. But hardware seldom works by itself. Small

software applications live on each of these devices, either as applications stored on the device, or as embedded software.

The Blackberry, iPhone, Treo, and numerous other devices work pretty much the same as a normal computer system. The hardware exists to process the software loaded on the device. But instead of loading the operating system into the normal storage devices, such as a microSD card, this particular software is often loaded into a special chip on the device. This is embedded software. What many people don't realize, however, is that this software is just as susceptible to vulnerabilities as normal operating systems on larger computing platforms. So, in addition to the normal means of software compromise, handheld devices are potentially vulnerable to attacks on the embedded software, as well.

If you recall the hype surrounding the iPhone when it first appeared on the market, you'll also understand just how detrimental a wave of virus or Trojan activity could potentially be for users of these devices. For example, when the 1.1.2 version of the iPhone firmware was released late last year, it came with a function blocking users from loading their own software on the device. Ironically, the hacking community saw this as a challenge and broke through this protection in a matter of hours of the firmware's release.

Even the Blackberry has been a victim to software hacks. At a recent security conference in New Zealand (<http://www.blackberrycool.com/2007/11/28/006149/>), a security researcher demonstrated how to get trojan software functioning on the Blackberry devices.

But what it comes down to, in the end, is user

Notes, cont.

accountability and responsibility. Are you careful about what you load on your device? Who do you connect to on a daily basis? Do you REALLY trust that software provider? Are you certain the application only does what it claims? And what happens if your hand held device becomes infected with malicious software? Do you “sync” the device up with your computer? Is your computer connected to a network?

In one step in the right direction, a security application developer for Apple platforms, SecureMac (<http://macscan.securemac.com/>), was recently accepted as an approved developer for all Apple platforms, which includes the iPhone. SecureMac intends to port its already popular malware scanner from the Mac platform to the iPhone. In addition, the company is already working on application that will allow greater security on these devices; and provide a mobile, handheld platform for security scanning of local networks.

But, in the end, the responsibility for the security of each device falls on the shoulders of the end user. Are you passing along guidance for the proper handling or security of these devices to your network users? With these devices playing a bigger role in the business world, it's critically important to keep a wary eye on the freedom they provide and protect ourselves from the potential ramifications.

Until next time,

-Russ



8/2 - 8/3, 2008	Las Vegas, NV
8/4 - 8/5, 2008	Las Vegas, NV Black Hat
8/23 - 8/24, 2008	Omaha, NE
9/9 - 9/10, 2008	San Antonio, TX
9/16 - 9/17, 2008	Tempe, AZ
9/23 - 9/24, 2008	Burbank, CA

The IAM is a two-day course for experienced Information Systems Security analysts who conduct, or are interested in conducting, INFOSEC assessments of information systems using NSA's methodology. It is a high-level, non-intrusive process for identifying and correcting security weaknesses in information systems and networks.

FREE 1 Year license to the SAINT Vulnerability Scanner just for attending any of our courses!

**Training so good, we teach
the competition!**

To register for one of these courses or to get additional information, please contact us at:

(719) 488-4500

info@securityhorizon.com

<http://www.securityhorizon.com>



...Why & What

Technical Surveillance Countermeasures (TSCM) is simply the identification or detection and locating of eavesdropping and surveillance devices that have been installed or are being used in an environment. These devices or “bugs” can be active or passive. In this article I will attempt to identify the reason for TSCM, or threats and types of devices or “bugs” that could be used.

The general rule of thumb is if you have information that somebody else desires or could profit from, then you are a potential victim of unknown or unauthorized eavesdropping or surveillance. Generally it has been the public belief that the targets of such activities are government, military, big corporations and investment groups. While these are, in fact, very common targets, they are not the only ones. The list should be expanded to include any individual with power or money that is coveted by another. Also, companies and/or individuals that are involved with any litigation or sensitive negotiations are prone to having eavesdropping or surveillance.

Eavesdropping can be found in use during criminal investigations by government agencies and civil suits. People have been known to bug their children and spouses, while Insurance companies have been known to bug claimants they believe may be committing insurance fraud. That does not mean those activities were always legal and I would always tell you to consult a lawyer on legal issues when in doubt. But the point is that the targets of eavesdropping or surveillance are considerably larger than government or corporate espionage.

The fact is that there is a significant underground industry in the U.S. consisting of the manufacture, sale, installation, and monitoring of such devices. There is information floating around, like the U.S. State Department/DCI March, 1997, report that claims that this underground industry is worth over two (2) billion dollars annually in the U.S. alone. The range of these activities goes from simple listening in on conversations to the installation and monitoring of sophisticated devices. As previously stated these devices can be active or passive, and all of them will fall into one of five primary categories of

bugs; Acoustic, Ultrasonic and Subsonic, Optical, Radiated Frequency (RF), and Hybrid.



Acoustic Bugging:

Acoustic bugging is by far the simplest. This can consist of such things as using a water glass to listen to the conversation on the other side of a wall or door or placing a plastic tube such as from a stethoscope into a small opening to listen in on a conversation. Do you remember listening to a conversation through the heating and ventilation ducts in a house? That is acoustic bugging. All acoustic bugging is done by listening with the naked human ear and without the means of electronic devices.

UltraSonic and Subsonic Bugging:

Ultrasonic and Subsonic bugging is a bit more sophisticated than Acoustic bugging. This involves the conversion of the audio signal to an audio frequency outside the human hearing spectrum. Conversion of the audio is usually done by the digital expansion or compression of the audio waves that are then transmitted

to a receiver. These types of devices are somewhat effective, and can be hard to detect. The drawback to using these is that they are not cheap and the range is limited based the frequency used, the density of any object within the line-of-sight between transmitter and receiver, and the power of the transmitter. Subsonic frequencies have greater range and can pass through low density objects, but are more prone to distortion. Ultrasonic frequencies have a much shorter range and are easily blocked by objects of any density.

Optical Bugging:

Optical bugging falls into two types; visual and audio. Visual surveillance consists of the commonly understood devices such as cameras, binoculars, and telescopes. Visual bugging is limited much like Ultrasonic and Subsonic bugging to line-of-sight, but can be significantly enhanced through digital techniques. The audio types are the devices that convert audio or other data into light beams. These provide for vary narrow beams to be focused on the target and reduce ambient or extraneous noise providing very clear signals. These are harder to detect but again are limited to line-of-sight. Think about the laser microphones that are displayed in spy movies, those are a good example of optical bugging.

RF Bugging:

Radiated Frequency (RF) bugging is by far the most well known type of device used. You could consider this type to be a merger between the three previous categories but it is considered separate due to the common

use of digital, not analog, frequencies. The frequency ranges vary based on the device from Very Low Frequencies (VLF) 3 KHz to 3 MHz on the low end to 900 MHz to 1.5 THz on the high end. It should be noted that the high end is the Microwave range. These bugs range from the “spy shop” devices that can be bought for under \$25.00 to sophisticated devices that can cost \$100.00 or significantly more. Most of these devices are considered to be disposable as they are very easy to detect. The issue with these devices is that while they may be easy to detect they do not rely on line-of-sight at the low or mid ranges and therefore the receiver is extremely hard to locate. Based on the power source and frequency used these devices can have a range between a few feet to several miles. The most common type of device is the hidden microphones commonly displayed in the spy movies.

Hybrid bugging is the combining of any of the previously discuss categories.

Wiretapping:

The last area to discuss is the most preferred method of obtaining information through bugging, and that is Wiretapping. Wiretapping is extremely difficult to detect as there is usually no transmission of energy to detect if installed correctly. The degree of expertise required to do wiretapping is highly skilled and not an amateur endeavor. Wiretapping is considered to consist of four distinct categories; Hard, Soft, Record, or Transmit.

Hard wiretapping is done by physically



connecting to the communication line (I.e. telephone or network), called tapping the line. This is done through physical access to the communication line. Even fiber optic communication lines are susceptible to being tapped, they just require more skill than normal copper lines. The idea is that wires are connected to the communication line and then the information is routed to another location for monitoring.

Soft wiretapping is done through the software used to control the phone system. Though most phone systems are controlled by the local telephone company there are many companies that own and operate their own Public Broadcast Exchange (PBX) for their internal phone system. A soft wiretap is the preferred method of law enforcement, government agencies, large corporations, and malicious hackers who tend to find it easy to access the software though the maintenance access. While it is easy to detect on a PBX, it may be difficult to gain cooperation from the local telephone company to check their PBX's. This type of wiretapping is the kind that you read about being done under the Patriot Act.

Record wiretapping is very simply tying in a recording device to the communication line. Electronically, this is hard to detect but is visually very easy to detect. The biggest drawback to using the record wiretap is the storage device that is connected. Access must be maintainable to the device to change out and recover the recordings. The best part of finding a record wiretap is that you can then set a trap to catch the "spy".

Transmit wiretapping uses the hard wire connection to the communication line but instead of copying the data or signal to another location or storage device, transmit

wiretapping utilizes RF frequencies to transmit. This is probably the most common type of wiretapping and probably the easiest to detect as with any bugging device that utilizes RF transmissions.

Within this paper I have identified that there is a real and plausible threat to any company or person that has power or money to risk. The degree of threat will vary based on the situation and circumstances that include litigation activities, mergers and acquisitions, and other sensitive negotiation like investment trading. Malicious activities include a range from the disgruntled employee to Corporate Espionage. The ability and skill required to do eavesdropping and surveillance ranges from the simple to the complex. All of the bugging can be classified into four categories and should be considered a risk along with the four categories of wiretapping.



Ed Fuller is the Chief Operating Officer and Senior VP of Security Horizon. He has over 30 years of experience in Operations, Communications, Computer Information Systems and Security. He is the primary lead for INFOSEC Assessments, Appraisals and Training for Security Horizon. He may be reached at ed@securityhorizon.com.



SOLARWINDS

Network Management Simplified



**DESIGNED BY NETWORK ENGINEERS
FOR NETWORK ENGINEERS.**

You won't find any shelf-ware at SolarWinds. Our affordable, easy-to-use network management products install in minutes & deliver robust, scalable functionality.

See why more than 45,000 customers trust SolarWinds products to manage their networks.

DOWNLOAD FREE 30-DAY EVALUATION VERSIONS TODAY AT SOLARWINDS.COM



49 Desktop Applications for Finding and Fixing Network Issues

Engineer's Toolset delivers the powerful network management, monitoring, and troubleshooting tools you need to easily and effectively manage your network right from your desktop — all in one extremely affordable package.

from \$1390



Scalable Performance Management for Complex Networks

Providing extensive visibility into the health of network devices, servers, and applications, Orion ensures that you have the real-time information you need to keep your systems running at peak performance all the time.

from \$2475



Change & Configuration Management Made Easy & Affordable

Cirrus is an affordable, easy-to-use network change and configuration management solution that automates device configuration and change management across multi-vendor network infrastructures.

from \$1245



Learn More Today:

**www.solarwinds.com
866.530.8100**



SOLARWINDS

System administrators have used Live Distros (short for distribution) for some time to help with a difficult system that will not boot properly.

More recently, a number of security minded distros have come about and are aimed at giving any system the ability to do security tests. As a security consultant, this is a great tool to get access to a system, or to provide another testing system without having to build an entire computer from scratch. The problem with these security tools is that they are updated on someone else's timetable and usually that is far too slow for an up-to-date environment. This article will describe how to create your own Live Distro using tools available on the Gentoo operating system

Before we get started, there is a tool developed specifically for the Gentoo community to build Gentoo releases, called Catalyst. It provides a number of Python scripts to assist in the building and updating of the standard stages with very little interaction from the user. By providing an easier method to build these systems, it removes a lot of the ability to customize. The Catalyst tool might be a better option for some. Check this process for building a live distro at the Gentoo Wiki and make your own decision.

The method described in this article is more detailed but provides two distinct advantages. The first is that we will use grub to boot the live distro, not ISOLINUX. This will allow for customizable boot screens, changing kernel parameters at boot time, and discovering devices. The second advantage is that the entire build source is saved between builds. This means we can treat this environment in the same manner as a normal system. We can remove or update programs without

having to start from scratch. This will reduce the build time between iterations.

In order to get started, we will need a system running Gentoo and make sure you have enough free disk space. The amount you need really depends on the number and type of programs you build into your live distro. A safe number is about 5 Gigabytes. You should also be familiar with the Gentoo process of building a system from the different stages. We will use stage 3 for simplicity but you can use any stage depending on your needs.

The first step is to ensure that we have proper tools on our system to build and read a live distro. The two programs needed are squashfs-tools and cdrtools. The former allows us to mount, read, and create the particular file system we will be using on the live distro. The latter is a set of programs that allow for the reading and recording of CD/DVDs, including the process to make an iso image. The build environment for our distro is simply a directory. I called mine liveDVD. Everything we do will be in that directory and we need to have the privileges of superuser through out this process.

Next, we need to get the source for our build process. We are going to assume that we need a system for Intel based computers. If you wish to build one for a different architecture, like Solaris or PowerPC, make the appropriate decision. I recommend the i686 branch because it is the most generic and has the highest likelihood of booting any system. If you want a truly optimized system, use stage 1 or 2 and build each program from scratch.

Honestly though, since we are running off a medium that has far less performance than the latest hard drives, this optimization will not

result in a faster system. Once the files are downloaded from your favorite mirror, extract the contents to `/root/liveDVD/source`.

```
# tar -jxpf stage3-i686-2007.0.tar.bz2 -C /root/liveDVD/source
```

We are now ready to start building our live distro operating system. We need to chroot into top directory of our source tree that we just downloaded. Chrooting provides a safe environment to build our new system, without affecting the normal environment our system boots into. This allows us to set different compile options and use unique environment variables. Before chroot'ing, create mount points from our normal environment to the distro for better performance and usability. Below are the mount points we need and the commands to chroot into our build environment.

```
# mount --bind /proc /root/liveDVD/source/proc
# mount --bind /dev /root/liveDVD/source/dev
# mount --bind /sys /root/liveDVD/source/sys
# mkdir -p /root/liveDVD/source/usr/portage/distfiles
# mount --bind /usr/portage/distfiles /root/liveDVD/source/usr/portage/distfiles
# cp /etc/resolv.conf etc/resolv.conf
# chroot /root/liveDVD/source/ /bin/bash -login
# env-update
# source /etc/profile
```

We are now safely in our build environment that is using the normal environment's `proc`, `dev`, etc. We need access to the internet for the latest sources, so we copied the

`/etc/resolv.conf` file from the base system. We also mounted the normal environments portage distribution directory, where all the sources are saved when building a new system. This provides two advantages. One, we will not have to download many of the common programs because our normal system already has them. Two, it keeps the size of the distro down because we will not need these source tar balls when running off the CD. We need to update the portage tree at this point, so run `emerge --sync` at the command prompt.

Lets now configure some of the base files for how this system will boot and tell other programs how to properly install. Lets start with `/etc/fstab` and don't forget to include an empty line at the end of the file.

```
/dev/loop0          /
squashfs            ro,defaults
0 0
none                /proc
proc                defaults
0 0
none                /dev/shm
tmpfs               defaults
0 0

# empty line for LiveDVD
```

Modify the line starting with `TIMEZONE` in `/etc/conf.d/clock` and set the correct time zone, and then copy the correct time zone file to `/etc`. It will save error messages during boot time. Adjust for your time zone of course.

```
TIMEZONE="MST7MDT"
```

```
# ln -s /usr/share/zoneinfo/MST7MDT /etc/localtime
```

Edit `/etc/make.conf` so that the proper

packages are included at build time. Below is an example that removes a lot of the extra packages that provide flash and usability; modify to suit your needs.

```
CFLAGS="-O2 -march=i686 -pipe"
CXXFLAGS="${CFLAGS}"
CHOST="i686-pc-linux-gnu"

USE="-gnome -kde -docs -java -acl
X livedd ssl"

FEATURES=-unmerge-orphans
```

Modify `/etc/locale.gen` and uncomment the correct locale and character set. This will reduce build time for many packages and keep the distro size down. While you are at it, edit `/etc/hostname`, `/etc/rc.conf`, and any other files, like `/etc/conf.d/net` you need for your environment.

We are ready to start customizing our distro. The first step is to unmask some key packages and get a working kernel. The first three commands unmask the required packages for live distros. They are masked because they will produce unexpected results in a normal system at boot time.

```
# echo app-misc/livedd-tools >>
/etc/portage/package.unmask
# echo sys-apps/hwsetup >> /etc/
portage/package.unmask
# echo "sys-fs/dmraid ~x86" >> /
etc/portage/package.keywords
# emerge -a sys-kernel/genkernel
sys-kernel/gentoo-sources app-
portage/gentoolkit
# genkernel --menuconfig all
```

The emerge command will download a few programs. The first is genkernel, which makes creating new kernels as painless as possible. Gentoo-sources is the latest kernel

source, and gentoolkit is a series of tools specific to the Gentoo operating system. The patching program is what we will utilize later.

The last statement will start the kernel build process for us. An interface will appear, allowing you to customize the kernel. Below are the required options specific to a live distro.

```
File systems --->
    CD-ROM/DVD Filesystems --->
        <*> ISO 9660 CDROM file
system support
    Pseudo filesystems --->
        [*] Virtual memory file
system support (former shm fs)
    Miscellaneous filesystems ---
>
        <*> SquashFS 3.2 -
Squashed file system support
Device Drivers --->
    Block devices --->
        <*> Loopback device
support
        <*> RAM disk support
(16) Default number of RAM
disks
(8192) Default RAM disk size
(Kbytes)
(1024) Default RAM disk block
size (bytes)
    ATA/ATAPI/MFM/RLL support --
->
        <*> Include IDE/
ATAPI CDROM support
```

I recommend enabling as many of the file systems extensions that make sense so you will be able to mount the widest range of local hard drives and disabling Ethernet over 1394 firewall. Once you have the options set appropriately, exit and save your configurations and go get a cup of coffee. The kernel build process will take from 5



minutes to over a half hour depending on the speed of your system.

Once the kernel has completed, we need to remove the udev package due to a conflict that will occur when we try to emerge the following software. Then we can install the rest of the required packages, including the three packages listed above the kernel configuration step that we unmasked above. UDEV will be reinstalled as a dependency as we continue with the steps in this article, so fear not. It is also a good idea to also install a memory testing utility.

```
# emerge -C sys-fs/udev
# emerge -a app-misc/livedcd-tools
sys-apps/hwsetup sys-fs/dmraid
sys-apps/memtest86+
```

Now update the base operating system. You can do this in one of two ways. The first is to conduct an `emerge -uD world` and rebuild nearly every program installed with the stage extraction. I have seen this break a number of things, and since this is not a read/write operating system the security risks are low. I recommend you execute `glsa-check -l affected` and review the list of issues that come up. This is one of the programs that

`gentoolkit` installs. Then execute an `emerge -u [package]` for the programs you feel are important to update. Also, build all the tools you want to include on your live distro. Some recommendations are your favorite editor (emacs vs. vi), a command line web browser like lynx, dhcpd to get a dynamic ip address at boot time, ftp, telnet, syslog, and mingetty for automatically login in as root at boot time.

Don't forget to include the file system utilities for all the extensions that were enabled during your kernel build process. Once all the applications are installed, update the system configuration files.

```
# etc-update
```

Now we need to configure many of the boot files required to get this system up and running. Let first set any boot time services up. Here are a couple of examples:

```
# rc-update add syslogd default
# rc-update add sshd default
```

A root password is a good idea as well, even if you installed mingetty.

```
# passwd
```

Configuring grub is the last step in the chroot'ed environment. First `emerge -a grub` and then delete `/boot/grub/grub.conf` and `/boot/grub/menu.lst`. The trick with a live distro on CDs or DVDs is that iso9660 cannot handle the symlink from `grub.conf` to `menu.lst`, which is the traditional method on normal systems. After deleting those files, create `/boot/grub/menu.lst` and edit the contents to look something like the example. The key aspects of this grub configuration is `loop=/livedc.squashfs` `udev` `nodevfs` `cdroot` `dodmraid`. This will enable the booting from the CD/DVD.

```
default 0
timeout 3
#splashimage=(hd0,0)/boot/grub/
splash.xpm.gz

title=LiveDVD
kernel /boot/kernel-genkernel-
x86-2.6.25-gentoo-r6 root=/dev/
ram0 real_root=/dev/loop0 vga=791
looptype=squashfs loop=/livecd.
squashfs udev nodevfs cdroot
dodmraid
initrd /boot/initramfs-genkernel-
x86-2.6.25-gentoo-r6

title=LiveDVD No Frame Buffer
kernel /boot/kernel-genkernel-
x86-2.6.25-gentoo-r6 root=/
dev/ram0 real_root=/dev/loop0
looptype=squashfs loop=/livecd.
squashfs udev nodevfs cdroot
dodmraid
initrd /boot/initramfs-genkernel-
x86-2.6.25-gentoo-r6

title=Memtest86+
kernel /boot/memtest86plus/
memtest.bin
```

Exit out of the chroot'ed environment, by typing `exit`. Make sure you reset your environment variables by issuing `env-update` and `source /etc/profile` at the command prompt, and then un-mount all of the connections from the chroot'ed.

```
# umount /root/liveDVD/source/proc
# umount /root/liveDVD/source/dev
# umount /root/liveDVD/source/sys
# umount /root/liveDVD/source/usr/
portage/distfiles
```

The directory `/root/liveDVD/source` now

contains the base system for our distro. If for any reason we need to correct something, or build another application in to our distro, simply repeat the steps in setting up the chroot and go. This install process allows for an update of the portage tree and updating the required packages at any time in the future.

To create the live distro, we are going to copy the source directory tree and then clean out the unnecessary parts of the environment. The tree that we build the distro from will be stored in `/root/liveDVD/target`. Since this is a read only environment, many of the standard aspects of an operating system are not needed, hence the cleaning. We also need to create a unique directory that enables the boot process from grub.

```
# rsync --delete-after --delete-
excluded --archive --hard-links
--quiet /root/liveDVD/source/boot
/root/liveDVD/target
```

Rsync is faster than `cp` and allows us to only copy the files we need.

```
# mkdir -p /root/liveDVD/target/
files/source
# mkdir /root/liveDVD/target/files/
source/newroot
# touch /root/liveDVD/target/files/
livecd
# rsync --delete-after --delete-
excluded --archive --hard-links
--quiet --exclude "var/tmp/*" --
exclude "var/cache/*" --exclude
".keep*" --exclude ".cvs*" -
-exclude "*sample" --exclude
"*example" --exclude "*.h" --
exclude "*.a" --exclude "usr/
portage" --exclude "etc/portage" -
-exclude "usr/share/doc" --exclude
"var/db" --exclude "usr/src" /
```

```
root/liveDVD/source/ /root/  
liveDVD/target/files/source/
```

The directory and file we create in the second and third steps above relate to the mount point the live distro will use for the RAM drive, which will hold the operating system. The distro will not boot if these are not present. The last command will move the necessary aspects to `/root/liveDVD/target/files/source/`. This will keep the distro as small as possible, allowing for as many programs as you need. If you are interested in doing more cleaning, execute the following commands after the `rsync` is complete.

```
# cd /root/liveDVD/target/files/  
source/  
# rm -rf var/tmp/*  
# rm -rf var/lock/*  
# rm -rf var/cache/*  
# rm -rf var/db  
# rm -rf tmp/*  
# rm -f etc/mtab  
# touch etc/mtab  
# mkdir var/log  
# mkdir var/lib/  
# mkdir var/lib/dhpcp  
# rm -rf usr/portage  
# rm -rf etc/portage  
# rm -rf usr/share/doc  
# rm -rf usr/src/  
# rm root/.bash_history  
# rm -rf root/.ssh/  
# rm /root/liveDVD/target/boot/.  
keep
```

Almost there. We need to create the special file system, which will hold the boot operating environment.

```
# cd /root/liveDVD/target/files  
# mksquashfs source/ /root/  
liveDVD/target/livecd.squashfs
```

The last step is to create the ISO image. The `-V` flag is the volume name, and the `-o` is the output of the build process. Alter to your liking.

```
# cd /root/liveDVD/  
# mkisofs -V "LiveDVD" -R -b boot/  
grub/stage2_eltorito -no-emul-  
boot -boot-load-size 4 -boot-info-  
table -iso-level 4 -hide-rr-moved  
-c boot.catalog -o /root/liveDVD/  
livedvd_x86.iso -x files /root/  
liveDVD/target/
```

Congratulations. If everything worked out, you now have an iso image that is ready to be burned to a CD or DVD. K3B is an excellent CD recording application. If you want to save a bunch of media, use VMware Workstation. Create a new virtual workstation with no hard drive and as little RAM as you care to test. Set it to boot from your new iso image and start it up. You will see any errors in the process and you can go back and correct or adjust as needed, rebuild, and test.

Below is a script that puts all the steps together. If things work, which I can almost guarantee they won't, the result of will be a working live distro. The script is a little different in that we also have a directory `/root/liveDVD/files/` that contains all the configuration files that we edit by hand above already prepared. The list of configuration files needed ahead of time are:

```
/etc/fstab  
/etc/make.conf  
/etc/locale.gen  
/boot/grub/menu.lst  
/etc/conf.d/hostname  
/etc/clock
```

We also create a special file, called `patch.pl`, for updating the operating system in the

Gentoo, cont.

chroot'ed environment. Below is the contents of the file and it should also be in `/root/liveDVD/files/`.

```
#!/usr/bin/perl

@issues = `glsa-check -l
affected`;
foreach $issue (@issues) {
    if ( $issue =~ / \((.+)\)/ ) {
        $tmp = `emerge -u $1`; }
    }
}
```

Finally, here is the bash script. Due to the size constraints of this article we've uploaded the script to a .txt file that you can download from the Peak Security website:

<http://www.peaksec.com/LiveDVD-script.txt>

References:

http://gentoo-wiki.com/HOWTO_build_a_LiveCD_with_Catalyst_for_newbies

http://gentoo-wiki.com/HOWTO_build_a_LiveCD_from_scratch
<http://www.unixwiz.net/techtips/chroot-practices.html>



Paul Criscuolo (CISSP) has been involved in the Computer Security Industry for over 15 years, working as the Intrusion Detection and Incident Response Team lead at Los Alamos National Laboratory, and most recently as a security consultant and Penetration tester. He is co-author of the recent *Nessus Network Auditing – Second Edition* book from Syngress. Paul can be reached at pcriscuolo@gmail.com.

Security
Horizon

Your global information security experts

Advertisers!
Reach *THOUSANDS*
of readers every
quarter at an
outrageously low
price!!

The *Security Journal* is never offline, making your ad constantly available-- ALL of our issues are still downloaded quarterly



Contact us at:
editor@securityhorizon.com
or
719-488-4500

SAINT®



Integrated Vulnerability Assessment & Penetration Testing

Examine. Expose. Exploit.

Examine your network with the SAINT® vulnerability scanner, and expose the areas where an attacker could breach your network. Then, take the next step and exploit the vulnerability with SAINTexploit™ penetration testing.

- ✓ PCI compliance reporting
- ✓ Correlation of CVE, CVSS, and exploits
- ✓ IPv4 and IPv6 scans and exploits

Get your free white paper at www.saintcorporation.com/products.html

For more information call 1-800-596-2006 x0119 or sales@saintcorporation.com

Complexity is the Enemy of Security

I have the privilege of teaching security courses for the University of Advancing Technology (www.uat.edu). One of the classes I teach is a Security Essentials course that introduces the basics of information security. It also introduces 12 Security Principles (Merkow):

- Principle 1: There is no such thing as absolute security
- Principle 2: The three security goals are confidentiality, integrity, and availability
- Principle 3: Defense in Depth as strategy
- Principle 4: When left on their own, people tend to make the worst security decisions
- Principle 5: Computer security depends on two types of requirements: functional and assurance
- Principle 6: Security through obscurity is not an answer
- Principle 7: Security = Risk Management
- Principle 8: The three types of security controls are preventative, detective, and responsive
- Principle 9: Complexity is the enemy of security
- Principle 10: Fear, uncertainty, and doubt do not work in selling security
- Principle 11: People, process, and technology are all needed to adequately secure a system or facility
- Principle 12: Open disclosure of vulnerabilities is good for security!

The principle I would like to focus on is Principle 9: Complexity is the Enemy of Security. This is true from two perspectives:

- Complex systems are more difficult to secure,
- The more complex the security

implementation, the more likely users will attempt to bypass the security mechanisms.

Complex Systems

Complex systems create a real challenge when trying to implement security. Just like the thousands of parts it takes to make an automobile run, the systems must keep it's thousands of pieces of hardware and code working in harmony for the effective functioning of the system. On top of an already complex system, security features add to the complexity, and must also work in harmony with the system. Great system designs with poor security are overall poor systems. Microsoft's "blue screen of death" and Unix's "kernel panic" are examples of when these complex mechanisms are not working together in harmony.

Another thought to consider is the more complex the system, the more challenges you have in operating and maintaining the system. Complex systems require complex infrastructure to operate and maintain. At some point, you can make the system so complex, there is no way to fix it when it malfunctions or stops operating.

Testing of complex systems to assure proper operation and security is difficult because of the complexity issues. You have a greater chance of missing something important.

Complex Security

Two major challenges with complex security are:

- Security can be so complex that it negatively affects the organization's operations

- Personnel will try to find work-arounds to security functionality to ease their pain

Complex security requires greater levels of technology, personnel, training, awareness, and monitoring. If security is so complex that it cannot be effectively managed, it can lead to operational impact and potential mission failure.

When it comes to security, people are generally considered the weakest link. I don't mean that in a derogatory manner, but if security is so complex that it impacts a person's ability to do their job, they are going to look for opportunities to reduce that impact, even if it makes them less secure. People get frustrated by how many times they have to enter their passwords, or the impact of a virus scan running while they are trying to do their work. Unfortunately, the IT staff is notorious for creating work-arounds that might bypass security features. The greater level of complexity for security, the greater impact to operations and the greater potential work-arounds that negatively impact security.

KISS (Keep It Simple Silly)

The KISS principle applies when considering security and the level of complexity associated with security. There MUST be a balance between the business needs and the security needs of the organization. In order to find this balance, an organization must conduct an activity such as a Business Impact Analysis (BIA) to understand what and the value of what they are trying to protect.

When it comes time to implement security into an organization, consideration has to be given to the political, technical, and physiological state of the organization. Generally trying to force a great deal of change at one time is going to receive great resistance from the organization. This would be amplified by trying to implement complex security. Security must be implemented in a way that is right for the organization. Making security complex just to be complex is not the right answer.



Resources:

Merkow, Mark and Jim Breithaupt; Information Security Principles and Practices, ISBN: 0-13-154729-1, Pearson Education, 2006



Greg Miles is the President of Security Horizon (www.securityhorizon.com) and an Adjunct Professor with the University of Advancing Technology. (www.uat.edu) He is also a co-author of the soon to be released *Techno Security's Guide to Securing SCADA: A Comprehensive Handbook On Protecting The Critical Infrastructure* (2008, Syngress Publishing, ISBN 1597492825). You may contact him at greg@securityhorizon.com.

RESEARCH AT
COLORADO
TECHNICAL
UNIVERSITY
DEPARTMENT OF
COMPUTER
SCIENCE

Hypothesis:

H₀

The state of the industry lacks the ability to enumerate QoP metrics to quantify the security posture of an organizational assurance information program.

H₁

The enumeration QoP metric can be developed that quantifies the security posture.

H₂

The QoP metric approach can be applied across like industry sectors.

The information collected will be used in my dissertation towards a Doctorate of Computer Science and future publications. This degree is from Colorado Technical University; 4435 N. Chestnut Street; Colorado Springs, CO 80907; 800-559-9287.

Doctorial Research

Mark Gerschefske Sr
CISSP, IAM / IEM

E-mail:

QOP.Survey@gmail.com

SECURITY METRICS - ENUMERATION OF QoP FOR INFORMATION ASSURANCE OF IT SYSTEMS

Doctorial Research

SECURITY METRICS RESEARCH FOR THE IMPROVEMENT OF IA

The purpose of this research is to develop metrics that enable the measurement of Quality of Protection (QoP) for IA systems. Some work has been done in this field and NIST has published guidelines on defining what metrics are. This work is intended to build on what has already been published and provide an agreement on how metrics can be built from existing policies and procedures. To further research in this area a survey has been structured to test the hypothesis (to the left). You're requested participation is solicited in this survey to further the research in the area of metrics within IT organizations. This survey should take less than 20 minutes to complete and all information provided will be recorded anonymously. Those who choose to participate will be given the opportunity to benefit from the research by receiving a report that will summarize the survey findings. The web survey can be accessed at:

http://www.surveymonkey.com/s.aspx?sm=AK5MoHwDoPcGbuDxfRf5kg_3d_3d

You may also send me an email at QOP.Survey@gmail.com and I will send you a link to the Website. Your privacy and email address will be protected and not shared with any third parties. I would like to thank you for your consideration in participating in this survey to expand the knowledge of metrics within the security community.



...for Network Design in Small to Medium Businesses

In many organizations, the network is not as reliable as it should be. With today's businesses, the network is as important as electricity, or the telephone. The company relies on the network for access to the critical applications that are used to serve customers and track operations. They rely on it to communicate via e-mail and to research material on the Internet. Senior executives have come to rely on the network to support strategic business initiatives and new applications. In the knowledge era, the network needs to be in a constant state of readiness (Carden, 2008).

A well-designed network provides the crucial backbone that all data moves through for any organization. It also provides the security or lack of security for potential threats. There are very little moving parts, and the equipment is likely installed in a closet where no one will ever see it to appreciate it. Most every aspect of an organization's data and communications will run through this equipment (Eckel, 2006). If a business has a poor network design, it poses potential problems right from the start. The business does not have the ability to plan for changes when the company becomes larger and needs to expand the network. Having a poorly designed network poses security issues if the network is not laid out to the maximum effectiveness. If an incorrectly placed device is added into the network, a potential bottleneck or security hole will arise. This initial design phase of the infrastructure is crucial to be well organized and addresses the business needs of the organization.

An infrastructure is made up of multiple components. When random errors occur, they are more difficult to diagnose. If the network is not configured optimally, most of the business'

IT resources will be spent on lower priority tasks while other higher priority tasks are not addressed timely. In order to have a viable network that works efficiently, there must be four key elements that define what a viable network looks like. These elements are scalability, high availability, monitoring and strong security.

Common practice for most business is to have some type of network in place to help conduct the electronic facets of the business. Most businesses realize that having a network is a requirement to share resources. Networks are becoming larger and more powerful as technology changes; yet, as the years go on, the network infrastructure goes untouched. This problem is perpetuated by other ongoing projects, downtime problems, money issues or other priority business activities. These problems cause the network maintenance to be postponed. New technologies emerge faster each year and create more security holes. The new speed increase causes bottlenecks to the network and overall degrades performance. IT's infrastructure components have a finite lifetime and need to be upgraded or replaced periodically, typically every three to four years. Once a problem is identified, whether it is a security breach or bottleneck, it is usually too late to patch the problem. The infrastructure needs to be overhauled and an arduous maintenance undertaking must be performed to attain an updated status.

Scalability needs to be proposed during the design phase of the infrastructure. Scalability involves key components be added seamlessly within the network, as the technology changes. Scalability allows the organization the ability to replace and reuse.

one's infrastructure, which decreases the network budget and allows the company to not have to replace the entire network every four to five years once it becomes outdated.

Most small and mid-sized businesses neglect the importance of proper network design. If constructed correctly, the network will facilitate change management processes for scalability and health of the infrastructure throughout the life of the organization. Small to medium size businesses neglect the network design for a few reasons. With most start-up businesses, the owners are worried about the initial costs and creating a stance in the marketplace. Non-technical individuals, such as family members and friends, are usually the ones running the business and maintaining the business environment. These members are not interested or knowledgeable in implementing the proper change management structure and network design. The work presented will show that small to medium businesses have a need for a best practices document on change management.

Having this blueprint prior to implementation provides an insight into each component and how it interacts with the rest of the structure. All small to medium businesses are unique in their needs for network designs; having an infrastructure tailored to the business' goals is ideal. Business owners are not aware of all components and information needed for a network. Two things the business can do to effectively plan from the start of the organization are to have an in-house Information Technology team to design, build, and maintain the network, or hire IT consultants to design and build the network. In-house technology teams are not practical for all organizations as most businesses do

not require this level of support. Once the IT consultant has completed their design and network build, it is up to the company to maintain the network. The business must decide which method is more effective for their business; planning prior to implementation is crucial.

The infrastructure used depends on the business needs. They can vary in size and



complexity. There are many hardware options within the network. During the lifecycle of the network, components become outdated and the network needs to be upgraded or completely changed. This can be costly to any organization with a limited budget. One solution to avoid the reoccurring costs is by

implementing modular hardware. Instead of replacing an entire core piece you can upgrade components, which cuts down cost, and reduces implementation time. An example of this type of product would be the Cisco 6500 series switch. Each of the modules on this type of switch can support multiple types of components to meet the needs of the organization. This modular system can grow as customer requirements expand and technology evolves, allowing businesses to upgrade and reconfigure systems by adding new modules, replacing existing modules, and adding and redeploying systems. (Cisco 2006).

A best practice for small and medium size companies could be Microsoft's best practices to network design. It entails a one server per service approach to design. This approach recommends is that each service one has to support, have one server or device on the network. In practice, having one server per service can get expensive. The benefits of using this best practice outweigh the negatives

outcomes. This best practice ensures a few good concepts in a network design. First this prevents a network from completely going down with a major server problem (Microsoft 2006). If a server does go down or the need for a major change occurs then only one service is affected.

Having one server per service helps to eliminate bottlenecks. With the network traffic distributed out to multiple servers, bandwidth to one focal point and problems that occur are circumvented. An example would be if a small game design company needed to add their game design software and the games that they were testing, and accounting files were all on one server, a bottleneck may occur. The server would not be able to handle all the requests that were needed for the business to continue working. The multiple servers would segment the traffic to one node and reduce bottlenecks. The testing employees could test the games while the coding team could work and all the business transactions could continue simultaneously without failure of the network.

The one server per service solution can be a benefit at the networking level. Even though on the there are no actual servers, the rule can be applied effectively. One possibility is the use of VLAN's. VLAN's can be used to segment the network into different broadcast domains which, in-turn, reduce traffic and creates another level of security to the network. Each part of your business can have its own VLAN and with the use of a router, can communicate to each other only when needed. Another aspect to the one server per service is having redundant networking devices like switches and routers. This prevents one switch going down and taking down the entire network. If a hardware malfunction is detected the failover switch or router immediately handles the traffic and

allows the business to stay functional. If a business has critical services that must run without interruption, then this approach is a must.

With having numerous networking devices on the infrastructure, a way a dividing the load among all the devices should be realized and addressed. Cisco has come up with a model that addresses this problem. This model has a series of layers divided into hierarchies depending on the assets value to the organization. Each layer creates a buffer to the other and if a network device malfunctions, the remaining part stays active. These layers are defined as the Core, Distribution and Access Layer (Thomas 2000).

The core layer is the center of the network for the entire enterprise and includes the highest-level backbones. This part of the network is built to provide reliable transport within the backbone structure and to the distribution layer. Traffic is coming in from the network and out of in this layer. Therefore it should be made using redundant elements to ensure maximum reliability (Thomas 2000). High performance and manageability are of the utmost importance at this layer. High availability devices such as an edge router or backbone switch need to be placed. Modularity is of great importance because all traffic passes through the devices at this layer.

Redundancy is a consideration at this level because of the uptime involved. Without modularity on these devices, updating and replacing are exceptionally hard. If these pieces of equipment were left not maintained, it would promote an increase in bottlenecks. Servers should be placed at this level as well. The reason for this is, depending on the server, the hardware needs to have the same type of high availability and bandwidth as the infrastructure to be an effective part of the

whole network.

The next layer of the hierarchical model is the distribution layer. This layer connects the access layers of the network to the core layers while retaining high routing efficiency. It controls access to the core layers, redistributes core layer routing protocols into optimized access layer protocols, and summarizes routes, if necessary (Thomas 2000). The size of the company determines if this layer gets used or not. Small businesses fewer than one hundred and fifty users could consider not having this layer at all. If the company is bigger than this, this layer is used for placing smaller more departmentalized switches and components. This layer is primarily for having the ability for fault tolerance among the networks devices. High bandwidth is still a consideration here at this level because it is the path to the core layer. Placing non-critical servers and higher end workstations at this level is a good idea. These pieces can be maintained more frequently because they have fewer nodes connected. Since this layer further segments the network redundancy is not as critical as the core layer. If a segment goes down then non critical applications are effected and should not result in much of a work stoppage. Modularity, even though not a crucial piece, could still be a benefit at this layer. If a network device such as a distribution switch needs to be upgraded because of bottleneck, replacing just the effected interface instead of the entire switch lowers the cost and reduces the downtime needed.

The last layer in Cisco's hierarchical model is the access layer. The access layer allows the infrastructure to transport data between the local segments and transport it to the higher layers. This layer consists of the network devices that are directly connected to the workstations including switches, hubs, patch

panels and wireless access points. These are where all the devices connect to gain access to the network. These network devices do not need to be as high bandwidth intensive as server or the backbone because the traffic is connected to the workstations. At this level high availability and modularity are not needed.

Workstations in a good network layout should not be connecting to the infrastructure devices or servers at the same speed. The same rule applies for server connecting to the network devices. They should not have the same speed as the connection of interconnecting network devices. For example, if a network needs to have servers connected at one gigabyte per second, then the workstations should have a lower speed such as one hundred megabytes per second or slower. The network connections need to have a higher speed such as five to ten gigabytes a second. Having the same speeds for each connection could cause bottlenecks to the servers they are accessing.

In a good design of the infrastructure is the monitoring of all processes that occur. Even with an excellent design, there can still be errors that happen within the network. In order to address these errors or problems, regardless of size, monitoring needs to be in place. These include network traffic analyzers that determine traffic costs on the network. Other devices, such as port analyzers, show whether the networks has errors and inconstancies which can slow performance.

Security monitors are used to show who is on the network and what users are accessing resources at any given time. For example, an SNMP sends information from devices in a network to a network management system. The network management server or appliance collects this data and report network issues

like faults in the line, interface issues or hardware that is not responding. Monitoring can also be used to send alerts when certain thresholds are exceeded (Carden 2005). These are invaluable tools that are a necessity to the network because without them a network administrator is only reactive to problems instead of proactive. This could be the difference between a healthy network and a slow network.

With all of the design considerations discussed, the final most essential piece to consider is having good security in place. Proper implementation of modularity, high availability and monitoring will not only inherently give the organization excellent security, but it will ensure consistency of the network. A consistent network can determine if a company can stay in business. If the network is down then resources and potential clients are lost due to this inactivity. Security needs to be address at the network design phase because implementing security after the fact will be more costly.

Security on a network has many levels. Since new ways of hacking into a network changes and gets ever easier each day, the network has to be updated and prepared. A good starting point is having effective Antivirus software on all servers and workstations. It is imperative that it is updated regularly to ensure maximum security. Patching and detailed planning of the entire infrastructure and operating systems are imperative. The organizations should create logs and reports that can be viewed easily by a Security Administrator to indentify randomness and differences in the network. This works in conjunction with the monitoring tools that were

included in the design phase. In conclusion, all of this would be routes designated in the infrastructure that determine who gets in and what gets out (Jerome 2005). If there is no blocking of these routes, then one should consider this an open and poor designed network.



Network design is an important process in any company's growth from the beginning. This could be an expensive endeavor or a saving grace to all the technology that is invested in later years. Having the right design could ultimately make or break a company if not planned properly. This is a huge concern in the business world especially since having a presence on the Internet in crucial. Business is done though websites, the sale of web applications and remote access to local resources. Since technology is ever changing, a network that can stay fluid is a must to stay in business. It must be able to keep up with the ever increasing bandwidth needs. If a network is left unmanaged, it could be disastrous. Even though there might not be bottlenecks on the network yet, the architecture of the infrastructure that originated from plans laid down years ago would not be able to cope with the demands that are going to be made in the coming years. Not only would bandwidth become a problem but also demands concerning the availability of the network and the seven days a week twenty four hours a day goal could not be guaranteed (Popov, 2003). Not getting that crucial sale or product could be devastating to a company and this is where designing the network to be in an ever changing computer industry is a must.

The Need for Change, cont.

References:

Eckel, Erik (2006), 10 things you should know about designing a small business network, Retrieved on November 12, 2006 from <http://articles.techrepublic.com.com/5100-1035-6119319.html>

Carden, Philip (2005), Network Baselining and Performance, Retrieved on October 23, 2006 from <http://www.networkcomputing.com/netdesign/base1.html>

Cisco (2006), Catalyst 6500 Advanced Services and Technologies Integration, Retrieved on November 20th 2006 from http://www.cisco.com/en/US/products/hw/modules/ps2706/prod_brochure0900aecd803701f0.html

Jerome (2005), Intrusion Detection, Retrieved on November 28th, 2006 from http://www.linuxexposed.com/index.php?option=com_content&task=view&id=72&Itemid=54

Microsoft (2006), Introduction to Architecture Blue Prints, Retrieved on November 19th 2006 from <http://www.microsoft.com/technet/itsolutions/wssra/raguide/architectureblueprints/default.aspx>

Popov, Oliver Blagoj(2003), Managing IT in an Academic Environment, Amsterdam, NLD: IOS Press, 2003

Thomas, Thomas II (2000), Building Scalable Cisco Networks, New York: McGraw-Hill.



Brian Hitch has 8 years experience in the IT industry. He owns Helios Network Consulting Inc, a consulting company based in Phoenix, AZ specializing in small to medium business audits configurations. He is also an Adjunct Faculty member at the University of Advancing Technology. Brian is currently finishing up his master degree in Information Technology Management scheduled to be completed in August of 2008 and may be reached through this publication.



We speak CVE®!

The INFOSEC Evaluation Methodology (IEM) is NSA's hands-on process for conducting evaluations of customer networks utilizing common technical evaluation tools. Students can expect to learn an easily repeatable methodology that provides each customer a roadmap for addressing their security concerns and increasing their security posture.

8/2 - 8/3, 2008

Las Vegas, NV

8/4 - 8/5, 2008

Las Vegas, NV

Black Hat

9/11 - 9/12, 2008

San Antonio, TX

9/18 - 9/19, 2008

Tempe, AZ

9/25 - 9/26, 2008

Burbank, CA

To register for one of these courses or to get additional information, please contact us at:

(719) 488-4500

info@securityhorizon.com

<http://www.securityhorizon.com>



Master the ghost in the machine.

Some may dream in code. We program the dreams. Gain comprehensive knowledge of application development, object oriented graphical programming. Conjure your calling as an innovator in the realm with a Bachelor of Science in Software Engineering degree.



Discover yourself and make the software spirits dance. UAT.edu/majors

Artificial Life Programming > Technology Management > Computer Forensics > Game Programming > Network Engineering > Network Security > Software Engineering
> Web Architecture > Robotics and Embedded Systems > Digital Animation > Digital Art and Design > Digital Video > Game Design > Game Art and Animation