

Setting Up a Security Test Environment Using VMWare and Overview of Live Security Boot Disks (Part 1 of 2)

By Jason Drury

Security Sites

- I am always interested in what sites people check to stay current on the latest security news, exploits, and tools
- Here is what I “try” to check on a daily basis
 - SANS ISC - <http://isc.sans.org/>
 - PacketStorm - <http://packetstormsecurity.org/>
 - Milw0rm - <http://www.milw0rm.com/>
 - A Hacker/Security Forum (from my bookmarks)
 - A Hacker/Security Blog (from my bookmarks)
 - <http://delicious.com/binaryman123> - My security/hacking bookmarks
- Goes along with the rest of the presentation because if there is a new exploit that effects your network, you can test it safely using your virtual test lab that I am about to describe

A Virtual Testlab

- Pro's

- **Biggest One** – learn how to hack without getting arrested and/or bringing your companies network down
- Test new tools without jeopardizing the stability of your daily work machine
- Safely dissect and learn about the latest malware
- Learn different operating systems and have them coexist on one machine
- No need to purchase a bunch of hardware, find space, rats nest of cables, etc
- Free (well mostly)

A Virtual Testlab

- Pro's

- Ability to take “snapshots” of the current state of a VM and revert back to that state if you managed to hose it. This is very helpful feature when dissecting malware
- The host OS's are saved as files, so you can back these up, move them, etc.
- You can “pause” a machine in it's current state and “play” it at a later time. Helpful if you need to

A Virtual Testlab

- Con's (not many)
 - Your host machine has to have good hardware specs (more on this later)
 - If you are not careful you could accidentally infect other machines on your network when dissecting malware

Virtual Machine Software

- There are quite a few free Virtual Machine software products
 - VMWare Server (the one I use and the one that this talk will focus on) – download from <http://www.vmware.com/download/server/>
 - VirtualBox (heard good things about it) - <http://www.virtualbox.org/>
 - Xen - <http://www.xen.org/>
 - Parallels (for Mac) - <http://www.parallels.com/>
 - OpenVZ (for Linux) - http://wiki.openvz.org/Main_Page
 - Microsoft – <http://microsoft.com/virtualization>
 - Even More - http://en.wikipedia.org/wiki/Comparison_of_virtual_machines

Virtual Machine Software - Hardware

- What I run my VM test lab on

- Dell Latitude D630 laptop

- Core2 DUO CPU
- 2GB RAM (would not recommend less than this because you normally need 2 VM's guests running at once with at least 512MB of RAM which leaves 1GB left for the host)
- 300 GB External USB Harddrive – no performance problems, but sometimes VMWare can't locate them and I have to reboot my machine or readd them

- Additional Recommendations

- 2nd External USB Harddrive – to backup your VM's – There are 2 people in this world, those who lost data by not properly backing up, and those that will.
Backup your backups!

Virtual Machine Terminology

- **Host** – the machine that the VM software is installed on
- **Guest** – the VM running within the host
- **Hypervisor** – *also called virtual machine monitor (VMM), is a computer hardware platform virtualization software that allows multiple operating systems to run on a host computer concurrently.* – Wikipedia
 - **Type 1 (native or bare metal)** – runs directly on the hardware, not within another OS – VMWare ESX server, Xen
 - **Type 2 (hosted)** – runs on top of an OS – VMWare Server, VirtualBox

VMWare Networking

- **Latest version of VMWare server has made VM networking much easier. This use to be my least favorite thing when dealing with VMWare.**
- **3 different options:**
 - **NAT – uses the host ip address to communicate with the outside world, no way to speak with the guest from the network**
 - **Bridged – guest has it's own unique ip address and it can speak to the world and the world can speak to it (just like if it was a separate PC on the network)**
 - **Host-Only – only the host and other guest VM's (if they are in the same network) can speak with it. What you want to use as much as possible, especially when dissecting malware**

VMWare

- **Vmware Demo**

- **Create a new virtual machine**
- **Add an existing VM to the datastore**
- **Show different networking options**

Live Security CD's

- About a dozen active one's and about the same number of inactive one's can be found here: <http://securitydistro.com/security-distros/>
- Biggest advantage of using these is that the tools are already preinstalled. If you ever had to install Linux apps manually, you know it can sometimes be a nightmare with dependencies, even when using a package manager (such as RPM). Also, sometimes trying to find some of these tools will take you into seedy parts of the internet and you can not always trust the sites you download these tools from nor can you be sure that the tool isn't backdoored. (I actually use VMWare for this also. If there is a tool that I suspect may be backdoored, I install it on a VM and run a sniffer on my host to see if it's sending any suspicious traffic)
- Can boot off of these CD's and it will not make any changes to your host OS

BackTrack

- The most famous of the live security CD's focused on pentesting
<http://www.remote-exploit.org/backtrack.html>
- Merged from two separate security distributions Whax and Auditor Security Collection
- More than 300 different tools preinstalled
- Good/Active Forum: <http://forums.remote-exploit.org/>
- Online course to learn offensive security using backtrack from the makers of backtrack: <http://www.offensive-security.com/training.php>
- Installed on Slackware Linux distro – not the most user friendly Linux distro but BackTrack does a good job of hiding this

BackTrack

- **Although BackTrack is a live CD, I installed it as a VM for two reasons:**
 - There are some tools that were not installed (ie. Nessus) that I wanted to have
 - It runs (much) faster compared vs from the CD
 - Able to update the software (and signatures) to the latest versions. For example the version of Nmap is quite old and Nmap has made many improvements (especially in OS and Service detection) in the latest version
- **My favorite tools on BackTrack**
 - Metasploit
 - Nmap (of course)
 - Entire milw0rm repository
 - Nikto
 - Hydra
 - Maltego
 - John the Ripper
 - Paros Proxy
- **There are still many tools I have not had a chance to play with, especially wireless**

BackTrack

- **Demo**

- **Most tools are located in /pentest**
- **Update milw0rm repository: /pentest/exploits**
- **Update Metasploit Framework**
 - /pentest/exploit/framework3
 - Run “svn update”

BackTrack

- **Autopwn**

- Because Metasploit wasn't easy enough
- Will only work on a (very) unpatched system
- HD Moore (author of Metasploit)'s blog post - <http://blog.metasploit.com/2006/09/metasploit-30-automated-exploitation.html>
- “A recurring theme in my presentations about Metasploit 3.0 is the need for exploit automation. As of tonight, we finally have enough code to give a quick demonstration :-)”
- Good demo to show your manager, boss, customers, why it is a good idea to patch their systems

BackTrack

- **Change to Host Only networking**
- **Autopawn WinXP box**
 - `./msfconsole`
 - `load db_sqlite3`
 - `db_create pentest`
 - `db_nmap -PN -p 445,139,135 <ip>`
 - `db_autopwn -p -t -e`
 - `-p` Select modules based on open ports
 - `-t` Show all matching exploit modules
 - `-e` Launch exploits against all matched targets
 - `sessions -l` (displays exploitable sessions)
 - `sessions -i <session #>` (connect to an exploitable session)

Samurai

- <http://samurai.inguardians.com/>
- *The Samurai Web Testing Framework is a LiveCD focused on web application testing. We have collected the top testing tools and pre-installed them to build the perfect environment for testing applications.*
- Very cool looking
- Built using Ubuntu Linux Distro
- Customized Firefox - webapp pentesting specific addons
- Has many webapp pentesting tools not found on BackTrack

Samurai

- **Some tools that are on here which are not on BackTrack (descriptions taken from MadIrish.net)**
 - Grendel-scan - open source web application vulnerability testing tool
 - w3af - web application attack and audit framework
 - Burp suite - a web application attacking tool
 - Dirbuster - an application file and directory enumeration and brute forcing tool from OWASP
 - WebScarab - a HTTP application auditing tool from OWASP
 - Siege – a HTTP stress tester and benchmarking tool.
 - Wapiti - a web application security auditor and vulnerability scanner
 - Webshag - a web server auditing tool
- **Demo w3af - *The project goal is to create a framework to find and exploit web application vulnerabilities that is easy to use and extend***

Misc

- If you are interested in learning about webapp security, checkout Webgoat:
http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
which is a deliberate insecure webserver with lessons to teach web applications security
- Live CD's to teach hacking/pentesting - <http://heorot.net/livecds/>