

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

By Jason Drury

Part 1 - Recap

- Virtual Test Lab

- Pro's and Con's
- Popular virtualization software available today
- Overview of my testlab environment
- Virtualization terms
- Virtualization networking

- Live Security CD's

- BackTrack – general penetration testing
- Samurai – web application testing

- Part 1 of this presentation can be found on the ArkLaTex ISSA website:

<http://www.arklatex-issa.org/index.htm>

- Very soon we will be launching our new site so check back shortly!
- I will also add this presentation to the website soon.

Part 2

- Overview of 2 more Live CD's
 - Helix
 - Ophcrack
- Time permitting – show Backtrack and Samurai since I was unable to last month due to technical difficulties (could not launch the console without being connected to a network, but I figured out that if you plug it into a hub by itself, it is perfectly happy. A loopback NIC would probably work as well)
- Also, since last months presentation I have been using VirtualBox (a WindowsXP guess running in a Slackware Linux host) and I have to say I really like it and find it easier to use then VMWare, although I had trouble with it recognizing my USB hardware.

Disclaimer

- Disclaimer – I have never done forensics for a legal/law enforcement case, but I have been involved in many incident responses (the two go hand and hand and are almost synonymous, but in my opinion forensics is more the law enforcement/legal side and incident response is more the IT side).

Ever since I heard of Helix I wanted to learn more about it and the tools that are included, which is the main motivation for me giving this presentation. The Helix user manual actually has very good information regarding forensics and I learned a lot about forensics just by reading the manual.

Helix Live CD

- Helix is geared towards live forensics and incident response

- You can download it from here: <http://www.e-fense.com/helix/>

- Based on Ubuntu Linux

- One of the biggest benefits of Helix is the fact it will not auto mount swap space, or auto mount any attached devices, which is important in a court of law since you do not want to be accused of tampering with the evidence

- Great user manual (and other docs) and active user forums at the support page: <http://helix.e-fense.com/Docs.html>

- The user manual is a bit out of date (ie. The older Knoppix+Xfce vs the current Ubuntu+Gnome), but it is still a very good read. I highly recommend it. I learned a lot myself reading it.

Helix Live CD

- The binaries are static, so they do not rely on any files on the system. This is very handy when you are doing IR because many times a rootkit replaces your executables (ie. /bin/lis) with rooted one's to hide the rootkit
- If you are going to use Helix for forensics, you will want to become proficient with netcat because many of the disk imaging tools use it to transfer the copied image to a "clean" system and do not allow you to save the file locally due to the threat of contamination. Here is a good netcat tutorial: http://www.ol-service.com/sikurezza/doc/netcat_eng.pdf
- Here is where you can find a list of most of the tools that can be found on Helix (comes in handy especially on the Linux side where it's sometimes hard to locate the tools) <http://www.forensicswiki.org/wiki/Helix>
- Helix can be run in two modes:
 - Windows Mode - approx. 90MB of Windows forensic tools. Run the tools directly off the cd, or run the Helix.exe GUI application (this does rely on system DLL's to run)
 - Linux Mode – Runs as a bootable/live CD. Mounts all drives read only

Helix Live CD

- Demo

- Windows GUI program

- Step through different menus showing available options
 - Show Windows Forensics Toolkit (WFT) options and show report that I ran previously
 - Demo File Recovery tool, Scan for Images
 - Windows Command line

- Linux Live CD

- A few GUI tools – Autopsy probably the best one – it is a front end for Sleuth Kit (<http://www.sleuthkit.org/sleuthkit/desc.php>) – which is very similar to WFT that we talked about above
 - Many command line tools in /usr/bin, /usr/sbin, /usr/local/bin – I wish they would have put them all in a “Hexlix/bin” directory – show tools on pg. 192

Windows Passwords

- First some information on how windows stores passwords

- Passwords shorter then 15 characters are very weak. Why?

They are stored using the much weaker LANMAN (for backward compatibility of course). This is hashing mechanism and this is how LANMAN deals with passwords:

- Uppercase all letters
- Pad up to 14 characters
- Split the password in two separate 7 password fields
- For example, if I chose a password of "PassWord!" LANMAN will store it as follows:
"PASSWOR" and "D!_____"
- During the demo, we will see how quickly this password gets cracked

- Passwords 15 or more characters are stored using the (slightly) better NTLM authentication

- This is the default for Windows NT/2k/XP/2003

- Vista and Windows Server 2008 use the much better NTLMv2 by default

Windows Passwords

- After learning about the 14 character limitation, this is how I choose passwords:
 - Pick a long phrase that I know I will remember, for example “shreveportbossieriswhereilive” – that is a 29 character password, no uppercases and no special characters. How many people would try to brute force a 29 character password!
 - You may say a 29 character password is impossible to remember, but since I started using this method, I have not forgot a password and I rarely mistype it (which is what I use to do often when I was including numbers and special characters).
 - I am now trying to get a 15+ character password policy implemented at my company (not an easy task)

Ophcrack

- Most people are probably familiar with password cracking and brute force, but few probably know about rainbow tables
- Here is the best definition I have found explaining Rainbow Tables (taken from <http://www.antsight.com/zsl/rainbowcrack/>)
 - A traditional brute force cracker try all possible plaintexts one by one in cracking time. It is time consuming to break complex password in this way. The idea of time-memory trade-off is to do all cracking time computation in advance and store the result in files so called "rainbow table". It does take a long time to precompute the tables. But once the one time precomputation is finished, a time-memory trade-off cracker can be hundreds of times faster than a brute force cracker, with the help of precomputed tables.
- So you can use a password cracker such as Ophcrack (<http://ophcrack.sourceforge.net/>) or Rainbow crack (<http://www.antsight.com/zsl/rainbowcrack/>) and find your own rainbow tables (which usually are a few hundred MBs in size)
- Or, better yet, you can use the live Ophcrack cd that also includes rainbow tables

Ophcrack

- Demo

- This is the only time that I am actually going to boot off of a live CD, otherwise Ophcrack will not be able to find the password hashes automatically (although you can still load them manually) and it runs much faster when it can use all of the system's RAM.
- Here is how the passwords are made up for each account. Let's see how fast Ophcrack can crack them
 - testuser1 - 9 char pass with upper case and special char
 - testuser2 - 29 character, all lowercase letters, password
 - testuser3 - 11 chars mix of special chars, upper case, and nums

Resources

- Here are some additional resources (that were not included in the presentation) that you may find useful:
 - The best VMWare Server 2.0 document that I have come across - http://www.virtuatopia.com/index.php/VMware_Server_2.0_Essentials
 - Free rainbow tables: <http://www.freerainbowtables.com/>
 - If you want a good brute force password cracker, I recommend john (both Windows and Linux) and Cain & Abel (Windows)