

Cold Boot Attack Tools for Linux

By Kyle Rankin

(Reprinted from the Linux Journal)

If you have used a computer for any reasonable length of time, you've learned about the difference between RAM storage and hard drive storage. Besides the fact that RAM is faster than hard drive storage, we also typically think that anything stored in RAM lasts only until the computer loses power, while data stored on a hard drive persists even when the computer is unplugged. Anyone who has lost power while working on a school assignment can attest to the temporary nature of RAM storage.

The Cold Boot Attack

It turns out that what we have learned about RAM isn't entirely true. On February 21, 2008, a paper titled "Lest We Remember: Cold Boot Attacks on Encryption Keys" was released. In this paper, the researchers describe their discoveries about RAM persistence and how they can be exploited. The researchers found that RAM isn't automatically erased when it no longer has power. Instead, RAM degrades over time, and even after a few seconds without power, you still can recover a significant amount of data. They also found that if you chill the RAM first, using liquid nitrogen or even a can of compressed air turned upside down, you can preserve the RAM state for more than 30 seconds up to minutes at a time—more than enough time to remove the RAM physically from a machine and place it in another computer.

By itself, although this discovery is surprising, what's most interesting are some of the implications if RAM contents can survive a reboot. It turns out that a number of common disk encryption tools for Windows, Mac and even Linux all store encryption keys in RAM. With this cold boot attack, if people lock their screens or even suspend their laptops, you could pull the power, grab the RAM contents and scrub it for any encryption keys. Essentially, you could compromise all of the common disk encryption techniques if you had a few minutes alone with a computer.

When I heard of this discovery, the first thing that came to my mind wasn't encryption, but forensics. I've written previously about forensics in *Linux Journal* [see "Introduction to Forensics" in the January 2008 issue], and in that article, I discuss the debate over how to respond initially when your server has been hacked. One school of thought favors instantly pulling the power on a compromised server. The idea is that you want to freeze the filesystem in place and don't want to risk that the attacker, or even the investigators for that matter, will destroy evidence. The other school of thought believes that pulling the power would destroy a lot of valuable data that exists only in RAM, so one should gather data from RAM first and then pull the power. With this cold boot attack, now you don't have to make that choice. If a server has been compromised, you can pull power first, and then reboot and grab the contents of RAM.

Cold Boot Attack Tools Released

In the paper, the researchers not only outlined the cold boot attack, they also described tools they had created to take advantage of this flaw. On July 16, 2008, the complete source code for these tools was released to the public at citp.princeton.edu/memory/code. In true UNIX style, each of the tools are small and single-purpose:

RAM imaging tools: the first set of tools enables you to image a system's RAM. Although you potentially could boot off a rescue disk like Knoppix and then copy the memory, the rescue disk itself will overwrite a substantial amount of RAM. With the provided tools, you have a small executable that you can boot either from a USB disk or over the network via PXE. The USB executable dumps the

Cold Boot Attack Tools for Linux

By Kyle Rankin

(Reprinted from the Linux Journal)

entire contents of RAM to the USB disk and then powers off or reboots the host. The attacker then can take the USB disk to another computer and use a corresponding tool to dump the memory from the disk into a file. The PXE executable sets up the target for remote control, so the attacker then can dump the RAM over the network to the PXE server.

Key-scanning tools: the second set of tools on the site can scan the RAM image you have created for encryption keys. The names of the tools are pretty self-explanatory. The aeskeyfind tool searches for AES keys, and the rsakeyfind tool searches for RSA keys.

Download and Build the Cold Boot Attack Tools

Since the source for all of these tools was released, you can download and use them yourself without too much setup. First, go to citp.princeton.edu/memory/code, and download the latest version of the bios_memimage tarball, or the efi_netboot tarball if you want to image a machine that boots with EFI. Then, unpack the tarball. For my examples in this article, I use the bios_memimage package.

The bios_memimage package contains a doc directory with good documentation on the project and how to build and use the source. The tools support both 32- and 64-bit environments. Although the 32-bit version technically will work on a 64-bit system, it can't address all the 64-bit environment's memory space, so you might not get a complete image. To build for a 32-bit environment, enter the bios_memimage directory and type make. To build for a 64-bit environment, enter the bios_memimage directory and type make -f Makefile.64.

Note: I noticed when I compiled the code on my environment, the build errored out with an undefined reference to `__stack_chk_fail`. This is due to GCC's new stack protection. As a workaround, edit the pxe/Makefile file and change the line that reads:

```
CFLAGS= -ffreestanding -Os -Wall -I../include -march=i386
```

to:

```
CFLAGS= -ffreestanding -Os -Wall -I../include -march=i386 -fno-stack-protector
```

USB-Based Cold Boot Attacks

Once the code has compiled successfully, you are ready to install the tools. The procedure is different for the USB and PXE tools. For the USB tool, you need a USB drive that you are willing to erase and that is big enough to fit the RAM you want to dump. In the usb directory is a bootable image called scraper.bin. Connect your USB disk (in my example, /dev/sdb), and then use the dd tool as root to overwrite the beginning of the drive with the boot image:

```
$ sudo dd if=scraper.bin of=/dev/sdb
19+1 records in
19+1 records out
9792 bytes (9.8 kB) copied, 0.0101028 s, 969 kB/s
```

Now the disk is ready. Go to the machine you would like to image, connect the USB drive, and then force a CPU reset or pull and then restore the power quickly. Then, set the BIOS to boot from the

Cold Boot Attack Tools for Linux

By Kyle Rankin

(Reprinted from the Linux Journal)

USB key. This will vary depending on the computer. On some BIOSes, you will press F12 or some other key to see a list of boot options; others require you to enter the BIOS configuration to change the boot order. In any case, once you boot from the USB key, the scraper tool immediately will start dumping the contents of RAM to the disk. Once it has completed, it will attempt an APM power-off or otherwise will reset the machine. Then you can unplug the USB drive and return to your machine.

You can use the provided `usbdump` tool under the directory of the same name to dump the RAM from the USB disk to your local drive. Simply specify the USB drive as an argument and then redirect the output to a file of your choice:

```
$ sudo ./usbdump /dev/sdb > memdump.img
recover segment0 [base: 0x0 size: 653312]
recover segment1 [base: 0x100000 size: 1062993920]
```

PXE-Based Cold Boot Attacks

The PXE-based scraper works somewhat differently from the USB-based scraper. First, if you don't already have a PXE server, you need to configure one. That process is out of the scope of this article, but I explained how to set up a PXE server in the article "PXE Magic" in the April 2008 issue of *Linux Journal*. Once you have a functional PXE server, copy the `pxe/scraper` binary to your `tftp` directory and change your `pxelinux` configuration so that it points to that file.

Next, connect the target system to the network (or if you set up the PXE server on a laptop, just connect the target system to the laptop via a crossover cable). Then, initiate a CPU reset or power off, and then immediately power on the target system. As with USB booting, different BIOSes have different ways to boot from PXE. On some BIOSes, you can press a function key, and others require that you change the boot order from the BIOS configuration.

Once the target machine gets a DHCP address and boots from the network, it will display a status message and then wait for the `pxedump` utility to connect. Unlike with the USB-based scraper, the PXE scraper doesn't automatically dump the memory over the network. Instead, you need to execute the `pxedump` binary found under the `pxedump` directory as follows:

```
$ ./pxedump target_machine_IP_address > memdump.img
```

Scan the Memory Dump

Once you have a dump from the target system's RAM, what can you do with it? Well, one of the primary things you can do is to scan the image for encryption keys. On the same page as the `bios_memimage` package, you will find tarballs for `aeskeyfind` and `rsakeyfind` utilities. To use these utilities, simply extract the source from the tarball and then run `make` within the source directory. Each source tree includes a `README` file that describes options with these utilities, but for basic scanning, just execute the `aeskeyfind` or `rsakeyfind` binary with the path to the memory dump as an argument. The tools will output any keys they find.

Unfortunately, there aren't a lot of other publicly available tools out yet that can reconstruct other useful information from a memory dump; however, you always can use the `strings` utility and `grep` to scan the image for keywords:

Cold Boot Attack Tools for Linux

By Kyle Rankin

(Reprinted from the Linux Journal)

```
$ strings memdump.img | grep keyword
```

Cold Boot Attack Limitations

This attack can be very effective, particularly against laptops. That being said, there are a number of limitations to this attack. For one, the machine you attack must be powered on, suspended or hibernated, because the RAM will start to degrade once the machine is powered off. Second, some BIOSes and all systems with ECC RAM will scrub the RAM before it boots an OS. In those cases, you either would have to attempt to disable this scrubbing or chill the RAM and move it to a system that doesn't do any scrubbing.

Resources

Official Page for the Cold Boot Attack: citp.princeton.edu/memory

Direct Link to the Research Paper: citp.princeton.edu/pub/coldboot.pdf

Source Code for Cold Boot Attack Tools: citp.princeton.edu/memory/code