Prevention of Coldboot Attack on Linux Systems

Measures to Prevent Coldboot Attack

Siddhesh Patil
Information Technology, Atharva College of Engineering
Mumbai University
Mumbai, India

Ekta Patel
Information Technology, Atharva College of Engineering
Mumbai University
Mumbai, India

Nutan Dhange
Assistant Professor
Information Technology, Atharva College of Engineering
Mumbai University
Mumbai, India

***Abstract*— Contrary to the popular belief, DDR type RAM modules retain stored memory even after power is cut off. Provided physical access to the cryptosystem, a hacker or a forensic specialist can retrieve information stored in the RAM by installing it on another system and booting from a USB drive to take a RAM dump. With adequate disassembly and analytic tools, this information stored in the RAM dump can be deciphered. Hackers thrive on such special-case attack techniques to gain access to systems with sensitive information. An unencrypted RAM module will contain the decryption key used to access an encrypted file system. Bits of data are stored in capacitors where a charge or a discharge denotes a 0 or a 1. Even on withdrawal of power from the RAM module, the capacitors still retain their values for a certain time frame. This window of time is highly vulnerable to a cold boot attack, and it can be extended by using proper cooling techniques. The very first demonstration of a cold boot attack was given by Alex Halderman and a team of researchers at Princeton University. Since then, Cold boot attacks have been widely demonstrated even on modern Android handsets with the use of a custom recovery. We go through various flaws present in modern RAM technology and take a look at some of the counter-measures that ensure safety. This paper presents an approach to design a preventive technique that will reduce the possibility of a cold boot attack.**

Keywords — Coldboot; Sidechannel; random access memory; Linux; preventive measures; Data remanence;

I. INTRODUCTION

Most computer users assume that switching off their machine removes any data held in random access memory (RAM) as it is referred to as volatile memory, and anything contained in RAM is considered lost when a computer is switched off. But as RAMs are made up of semiconductors, the contents are not lost immediately from it when the power supply is disconnected. It exhibits a property called data remanence. Data retention is observed for a period of time ranging from several seconds to several minutes. Random access memory or physical memory is used by the computer to temporarily hold data currently used by a given application. All the data manipulated is written temporarily in RAM for example texts, saved files, passwords and encryption keys. The more recent the activity, the more likely it is for the data to still be in RAM. So it may contain sensitive information, which if exploited, can cause loss for an individual or an organization. An attacker having physical access to a computer could recover some or all the important data from that session. So it is important that the sensitive data should be either wiped off or should be overwritten by scrambled information so that it erases all traces from that session on that computer. RAM modules exhibit a data remanence flaw which can be exploited. Most systems do not have techniques to ensure prevention of side-channel attacks. Unencrypted RAMs are more susceptible to cold boot attack techniques. We implement measures to ensure that data stays secure on system shutdown. Data security is critical for most of the business and even home computer users. RAM is used to store non-persistent information into it. When an application is in use, user-specific or application-specific data may be stored in RAM. This data can be sensitive information like client information, payment details, personal files, bank account details, etc. All this information, if fallen into wrong hands, can be potentially dangerous. The goal of most unethical hackers / attackers is to disrupt the services and to steal information. He/she can take a RAM dump and gain all the contents of the RAM. The aim of this project is to prevent cold boot attack on Linux system so that all the sensitive data retained by RAM for a window of time can be protected by attackers, thus keeping it from being misused. The scope of the project lies in the fact that most security based operating systems still are vulnerable to side channel attacks. Military based systems housing sensitive information are targets to such attacks. This being a very specific case scenario can still be avoided. A CEO or a high position executive using an Android operating system based smartphone or a government official can be open to such attacks. Our methods to prevent cold boot attacks can be implemented on:

1) Android smartphones with a recovery partition (most smartphones ship with a recovery mode).

2) Sensitive systems that are likely to cold boot attacks

II. REVISTING PREVIOUS RESEARCH

Data remanence:
Information is stored in bits on capacitors. Each capacitor houses one bit of information as a 0 or a 1. A low charge denotes a 0 and a high charge is read as a 1. This positive logic methodology enables a sequential storage of information on capacitors. DRAM cells need to be continually refreshed as they lose their charge after a certain period of time. DRAM specifications assign a specific refresh time interval after which the cells are refreshed. J. Alex Haldeman's research [1] showcased an experiment to demonstrate data remanence in different RAM modules.

Effect of temperature on decay rate:

Using a modified version of PXE memory imaging program (Ref. 1, J. Alex Halderman), memory regions in the RAM were filled with pseudorandom patterns. These memory regions were read after varying periods of time under different temperature conditions. Error rates were measured from each sample. Error rate is the number of bit errors in each sample (Hamming distance from the pattern prescribed) and divided by the total number of bits measured. The pseudo random number test patterns contained equal number of 0s and 1s,

a full decayed memory was expected to have an error rate of app roximately 50%

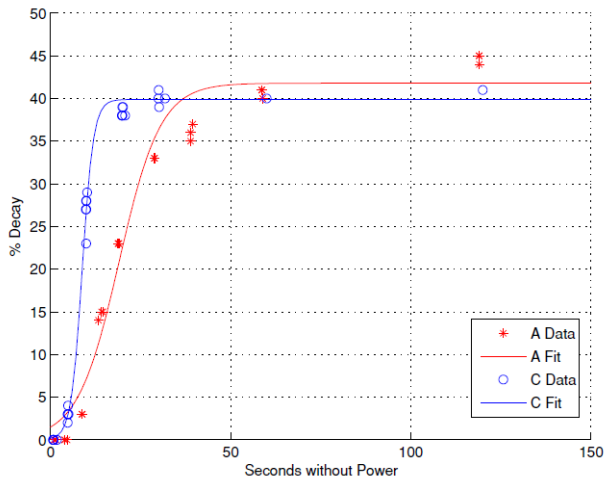


Fig.2.1

Under normal operating temperatures (25.5 to 44.1 degree Celsius) each memory module exhibited a varying rate of data loss. The decay curves showed a similar shape with an initial slow decay followed by a rapid decay. Higher data retention times means less frequent refresh intervals, which saves power. Manufacturers increase the retention times for the added benefit, the negative side of this implementation is the increased retention times.

Decay rate at reduced temperatures:

Lower temperatures can significantly increase the retention times, to measure this effect in DRAMs a second series of experiments was conducted (Ref.1, J. Alex Halderman). A pseudo random test pattern was loaded into the memory. This time the computer was running a memory module to approximately -50 degrees Celsius. The machine was then Powered off and maintained at that temperature until the power was restored. These temperatures were achieved using canned air products. A significant lower rate of decay was observed at these temperatures. As an extreme test another experiment was conducted using liquid nitrogen where the memory module was submerged in it for 60 minutes. The decay rate was observed to be at merely 0.17%. This indicated that memory modules can retain memory for hours or days provided that they are housed in sufficient cooling environment s.

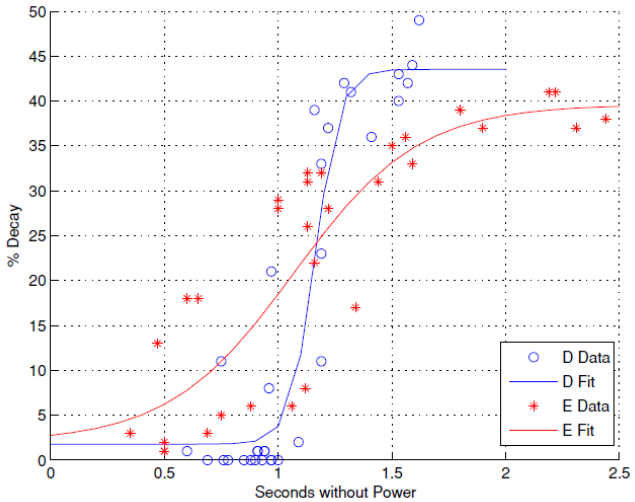


Fig.2.2

Decay rates comparison:

Test machines	*Seconds w/o power*	*Error % at operating temperature*	*Error % at -50℃*
A	60 300	41 50	(no errors) 0.000095
B	360 600	50 50	(no errors) 0.000036
C	120 360	41 42	0.00105 0.00144
D	40 80	50 50	0.025 0.18

III. STANDARDS IN RAM TECHNOLOGY

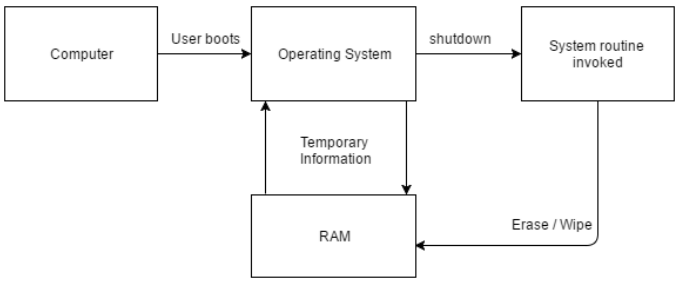
DDR type RAMs offer varied benefits over Static-RAM technology which is much more suited for CPU caches. DDR RAMs introduced in 2000s enabled a dual data rate which let memory transfer capabilities on both the rising and falling edges of the clock signal (Duplexing). This meant that information could move nearly twice as quick as older SDRAMs which offered simplex rates. Memory can run on lower clock rates using less energy to achieve faster transfer speeds. DDR2 introduced in 2003 refined the DDR type 1 technology by adding an internal clock running at half the speed of the data bus. It's twice as fast as DDR1 with 2x transfer rates consuming less power. DDR3 type RAMs released in 2007 offered an internal clock running at half of the speed of the data bus of DDR2 which meant it was twice as faster as compared to DDR2. Cold boot attacks have been successfully demonstrated on DDR3 RAM technology.

IV. PROPOSED SYSTEM

To reduce the possibility of a cold boot attack we propose a system routine or a command line utility that can be manually executed. The executable will be coded in C or C++ depending on the libraries that would be necessary as we go along the design. Upon shutdown of the computer the memory in the RAM will be wiped. The routine will work as follows:

- The user starts the computer normally and does his work.
- Note that at this point, data is being stored in the random access memory and is in use.
- Once the user is done working on the computer he will issue a shutdown or reboot command on the Linux terminal with additional flags if necessary.
- As soon as the command is issued the RAM-wipe routine will fire up and wipe the contents of the RAM.

System Block Diagram:



Provisions will be made in such a way that the system routine can be manually invoked upon will. The executable will be a result of compilation of the source code which will be machine dependent. On different architectures the executable will differ, however the functionality essentially will be the same. Our primary architecture in focus here is Intel's x86 based. Android smartphones ship with ARM architectures with the exception of a few running on Intel's x86 based platform. On ARM architectures running an Android operating system the execution strategy may differ.

Modules:

Our routine consists of a single module that resides in the operating systems binaries. It will be elevated with the right permission set and paths to execute it. We will look at each of the steps involved in the functionality of the routine. Erasing the data or deletion will happen as follows:

4.1 Overwriting procedure

The overwriting procedure can be invoked in a secure mode. It is the first step in the RAM-wipe routine. The procedure will do a 35 to 38 times overwrite on the memory. After each pass the disk cache will be flushed.

4.2 Truncation

Truncation means the file will be split or shrunk to a smaller size. The 2nd step of the routine is to truncate the file, this essentially will separate the disk blocks and even if the attacker retrieves this information the contents will not belong to the same disk block or sequential disk blocks. This will highly delay the process of reverse engineering the RAM memory dump and retrieving the actual file.

4.3 Renaming

The file will be renamed so the attacker won't be able to draw a conclusion from the retrieved information. This is the 3rd step in the data erasing technique.

4.4 Deleting

Finally, the file will be deleted, this will be achieved by unlinking the file nodes.

Note that the routine will not be efficient if there is a hardware – backed cache controller that enables disk caching. Caching will essentially store some information for faster start-ups or to prevent overheads. If the system has hardware cache controllers, then it is essential to disable them for a complete secure wipe.

IV. HARDWARE AND SOFTWARE DETAILS

Hardware	Software
1) Intel x86 /AMD x64 / ARM 32/64 bit processor.	1) Linux kernel based operating system (Ubuntu 15.04 / 16.04 LTS).
2) DDR type random access memory module	2) GCC toolchain (Architecture specific).
	3) Git for repository and CVS management

VI. DESIGN DETAILS

Use case scenario:

A use case describes a system's behaviour as it responds to a request that originates from outside of that system. In other words, a use case describes "who" can do "what" with the system in question. The use case technique is used to capture a system's behavioural requirements by detailing scenario driven threads through the function requirements. A defined purposeful, interaction between a system and a human of non-human actor that is playing a specify role outside the system.

VII. OUTPUT

With the in-depth analysis of the subject, study of various techniques involved in exploitation of the vulnerability, a routine will be the output. This routine will likely be a part of the operating system's core mechanics while leaving a room for manual execution without the system intervention. Systems using this routine will be highly secure from the possibility of a cold boot attack.

REFERENCES

[1] J. Alex Halderman, Seth D Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph Calandrino, Ariel Feldman, Jacob Applebaum, Edward Felten, "Lest we remember : Cold boot attacks on encryption keys" at the 17th USENIX security symposium.

[2] S. Lindenlauf, H. Hofken, M. Schuba, "Cold Boot Attacks on DDR2 and DDR3 SDRAM" at Availability, Reliability and Security (ARES) 2015 10th international conference.

[3] Jos Wetzels, "Hidden in snow, revealed in the thaw : Cold boot attacks revisited" in Seminar Information Security Technology - Kerckhoffs Institute, 2014.