International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-4, January 2014

# An Overview of Cold Boot Attack and Prevention Using Cryptography

Santosh Nandi
Lecturer, Panskura Banamali College,
Panskura,Paschim Medinipur,W.B. INDIA.
Nandisantosh21@gmail.com

## Abstract.

From past few years, we have seen that so many computer Threats arises and. Among them one of the most threatening attacks is Cold Boot attack. Even though a target machine uses full disk encryption, cold boot attacks can retrieve unencrypted data from RAM. Cold boot attacks are based on the remanence effect of RAM which says that memory contents do not disappear immediately after power is cut, but that they fade gradually over time. This effect can be exploited by rebooting a running machine, or by transplanting its RAM chips into an analysis machine that reads out what is left in memory. However, in this paper, we want to give the brief information about Cold Boot attack and its effect in the area of cryptography.

**Keywords:** Threats, Cold Boot attack, RAM, Memory.

## 1. Introduction

Contrary to popular assumption, DRAMs used in most modern computers retain their contents for several seconds after power is lost, even at room temperature and even if removed from a motherboard. Although DRAMs become less reliable when they are not refreshed, they are not immediately erased, and their contents persist sufficiently for malicious (or forensic) acquisition of usable full-system memory images. Researchers at Princeton University have shown that there are surprisingly large numbers of machines where the contents of RAM survive undamaged well after the system BIOS or boot code has finished running, and these can be exploited. To demonstrate this, we will try to capture and analyze the memory content after the system is powered off. this kind of attack is known since the 1990s. However, only in 2008, Halderman et al. have shown that cold boot attacks can be well deployed in practical scenarios.

## 2. What Is Cold Boot Attack?

In cryptography, a cold boot attack [1] [2] [6] is a type of side channel attack in which an attacker with physical access to a computer is able to retrieve user's specific sensitive information from a running operating system after using a cold reboot to restart the machine from a completely "off" state. The attack relies on the data remanence property of DRAM and SRAM to retrieve memory contents which remain readable in the seconds to minutes after power has been removed.

It has been known since the 1970s that DRAM cell contents survive to some extent even at room temperature and that retention times can be increased by cooling. In a 1978 experiment, a DRAM showed no data loss for a full week without refresh when cooled with liquid nitrogen.

Machines using newer memory technologies tend to exhibit a shorter time to total decay than machines using older memory technologies, but even the shorter times are long enough to facilitate most of the attacks.

Launching an Attack:

**Step 1: Powering Off the Machine:**

Simple reboots The simplest attack is to reboot the machine and configure the BIOS to boot an imaging tool. A warm boot, invoked with the operating system's restart procedure, provides software an opportunity to wipe sensitive data prior to shutdown. A cold boot, initiated using the system's restart switch or by briefly removing and restoring power, will result in little or no decay depending on the memory's retention time.

Restarting the system in this way denies the operating system and applications any chance to scrub memory before shutting down.

**Step 2: Fetching the Contents of the Ram:**

or this , simply place the dram in other machine and start the system, Or alternatively , keep the ram in the same machine , attach the bootable USB flash drive in the USB PORT , and reboot the system . Note that the boot priority of the system must be set to 'External USB Drive' and not to 'Internal hardDrive'. Otherwise the system will reboot again into its native Operating System.

Having done this, the memory-imaging tool or scrapper present on USB Drive starts executing. It fetches the memory dump present on the RAM into the USB Drive.

**Step 3: Making the Memory Dump Readable:**

After taking the memory dump of the RAM in a USB drive. It can now be analyzed. For this purpose, the data can be read straight out of the dump either by dumping it to a flat-file using 'dd' or by examining it in-place. For our experiment, we will dump the data to a flat-file. We also extract the human-readable content to a separate file.

## 3. Cold Boot attack and Cryptography

In cryptography [3], a cold boot attack (or to a lesser extent, a platform reset attack) is a type of side channel attack in which an attacker with physical access to a computer is able to retrieve encryption keys from a running operating system after using a cold reboot to restart the machine. [1][2] The attack relies on the data remanence property of DRAM and SRAM to retrieve memory contents which remain readable in the seconds to minutes after power has been removed.

To execute the attack, a running computer is cold-booted. Cold-booting refers to when power is cycled “off” and then “on” without letting the operating system shut down cleanly, or, if available, pressing the “reset” button. A removable disk with a special boot sector is then immediately booted (e.g. from a USB flash drive), and used to dump the contents of pre-boot memory to a file.[4] Alternatively, the memory modules are removed from the original system and quickly placed in a compatible machine under the attacker's control, which is then booted to access the memory. Further analysis can then be performed against the information that was dumped from memory to find various sensitive data, such as the keys contained in it (automated tools are now available to perform this task for attacks against some popular encryption systems [3]).

The attack has been demonstrated to be effective against full disk encryption schemes of various vendors and operating systems, even where a Trusted Platform Module (TPM) secure cryptoprocessor is used.[2] This is because the problem is fundamentally a hardware (insecure memory) and not a software issue. While the focus of current research is on disk encryption, any sensitive data held in memory is vulnerable to the attack. [2]

Compressed air can be improvised to cool memory modules, and thereby slow down the degradation of volatile memory With certain memory modules, the time window for an attack can be extended to hours by cooling them with a refrigerant such as an inverted can of compressed air. Furthermore, as the bits disappear in memory over time, they can be reconstructed, as they fade away in a predictable manner. [2] In the case of disk encryption applications that can be configured to allow the operating system to boot without a pre-boot PIN being entered or a hardware key being present (e.g. BitLocker in a simple configuration that uses a TPM without a two-factor authentication PIN or USB key), the time frame for the attack is not limiting at all:[2]

This is not the only attack that allows encryption keys to be read from memory—for example, a DMA attack allows physical memory to be accessed via a 1394 DMA channel. Microsoft recommends changes to the default Windows configuration to prevent this if it is a concern. [6]

The ability to execute the Cold Boot attack successfully varies considerably across different systems, types of memory, memory manufacturers and motherboard properties, and is more difficult to carry out than software-based methods or a DMA attack.

### 3.1. Full Memory Encryption

Encrypting random access memory (RAM) mitigates the possibility of an attacker being able to obtain encryption keys or other material from memory via a cold boot attack. This approach may require changes to the operating system, applications, or hardware. One example of hardware-based memory encryption was implemented in the Microsoft Xbox.[8] Software-based full memory encryption is similar to CPU-based key storage since key material is never exposed to memory,

but is more comprehensive since all memory contents are encrypted. There are multiple academic papers describing methods of encrypting memory and at least one commercial product from PrivateCore. [5][7]

## 3.2. Dismounting encrypted disks

Most disk encryption systems overwrite their cached encryption keys as encrypted disks are dismounted. Therefore, ensuring that all encrypted disks are dismounted (secured) when the computer is in a position where it may be stolen may eliminate this risk, and also represents best practice. This mitigation is typically not possible with the system disk that the operating system is running on.

## 3.3. Advanced encryption modes

The default configuration for Bitlocker uses a TPM without a boot PIN or external key—in this configuration, the disk encryption key is retrieved from the TPM transparently during the operating system startup sequence without any user interaction. Consequently, the Cold Boot Attack can still be executed against a machine with this configuration, even where it is turned off and seemingly safely secured with its keys in the TPM only, as the machine can simply be turned on before starting the attack.

Two-factor authentication, such as a pre-boot PIN and/or a removable USB device containing a startup key together with a TPM, can be used to work around this vulnerability in the default Bitlocker implementation. In this mode, a PIN or startup key is required when turning the machine on or when waking from hibernation mode (a power off mode). The result is that once the computer has been turned off for a few minutes, the data in RAM will no longer be accessible without a secret key; the attack can only be completed if the device is obtained while still powered on. No additional protection is offered during sleep mode (a low power mode) as the key typically remains in memory with full disk encryption products and does not have to be re-entered when the machine is resumed.

## 3.4. Power management

Shutting down a computer causes a number of well-known encryption software packages to dismount encrypted data and delete the encryption keys from memory. When a machine is shut down or loses power and encryption has not been terminated (such as in the event of sudden loss of power) data may remain readable from tens of seconds to several minutes depending upon the physical RAM device in the machine. Ensuring that the computer is shut down whenever it might be stolen can mitigate this risk. [2][5]

For systems using the hibernation feature (ACPI state S4), the encryption system must either dismount all encrypted disks when entering hibernation, or the hibernation file or partition would need to be encrypted as part of the disk encryption system.

By contrast sleep mode (ACPI states S1, S2 and S3) is generally unsafe, as encryption keys will remain vulnerable in the computer's memory, allowing the computer to read encrypted data after waking up or after reading back the memory contents. Configuring an operating system to shut down or hibernate when unused, instead of using sleep mode, can help mitigate this risk.

## 3.5. TCG-compliant systems

Another mitigation method is to use hardware and an operating system that both conform to the "TCG Platform Reset Attack Mitigation Specification",[6] an industry response to this specific attack. The specification forces the BIOS to overwrite memory during POST if the operating system was not shut down cleanly.

However, this measure can still be circumvented by removing the memory module from the system and reading it back on another system under the attacker's control that does not support these measures (as demonstrated in the original paper).

## 3.6. Booting

Although limiting the boot device options in the BIOS may make it slightly less easy to boot another operating system, [4] many BIOSes will prompt the user for the boot device after pressing a specific key during boot. Limiting the boot device options will not prevent the memory module from being removed from the system and read back on an alternative system either. In addition, most chipsets allow the BIOS settings to be reset if the mainboard is physically accessible, allowing the default boot settings to be restored even if they are protected with a password.

## 3.7. CPU-based key storage

Kernel patches for Linux such as TRESOR [4] and Loop-Amnesia [4] modify the kernel of an operating system

so that CPU registers (in TRESOR's case the x86 debug registers and in Loop-Amnesia's case the AMD64 or EMT64 profiling registers) can be used to store encryption keys, rather than RAM. Keys stored at this level cannot easily be read from userland and are lost when the computer restarts for any reason. TRESOR and Loop-Amnesia both must use on-the-fly round key generation due to the limited space available for storing cryptographic tokens in this manner. For security, both disable interrupts to prevent key information from leaking to memory from the CPU registers while encryption or decryption is being performed, and both block access to the debug or profile registers.

A 2010 thesis identified two register areas in modern x86 processors which could potentially be used for key storage: the SSE registers which could in effect be made privileged by disabling all SSE instructions (and necessarily, any programs relying on them), and the debug registers which were much smaller but had no such issues. The author left the latter for others to examine, and developed a proof of concept distribution called paranoix based on the SSE register method.[4]

The developers claim that "running TRESOR on a 64-bit CPU that supports AES-NI, there is no performance penalty compared to a generic implementation of AES",[7] and run slightly faster than standard encryption despite the need for key recalculation.[6] The primary advantage of Loop-Amnesia compared to TReSoR is that it supports the use of multiple encrypted drives; the primary disadvantages are a lack of support for 32-bit x86 and worse performance on CPUs supporting AES-NI.

A second approach to mitigating the cold boot attack is known as "frozen cache" (sometimes known as "cache as RAM"); [1], which disables the CPU's L1 cache and uses it for key storage. Disabling the CPU cache in this manner is disastrous for performance to the point that early experiments appear to indicate such a system would be too slow to be usable for most purposes.[7] Multicore CPUs may mitigate this issue, since only one core would need to have its cache disabled, but it appears examination of this approach has stalled.

## 3.8. soldering

If memory modules are soldered onto a motherboard, then they cannot easily be removed and inserted into another machine under an attacker's control.

## 4. Future work

In future work, we will examine the Cold Boot attack practically with various types of RAMs, in a fixed environment. We also decide to propose a new cryptography which can prevent this type of attack.

## 5. Conclusion

In this work, we have seen that Cold Boot attack also can take place after the system power off. It's attack the RAM (any types of) and capture the sensitive and authenticate information very easily. For This, we have to take proper caution also after the system power off.

## 6. References

1. **"**On the Practicability of Cold Boot Attacks", Michael Gruhn and Tilo M¨uller, Friedrich-Alexander-Universit¨at Erlangen-N¨urnberg, Germany.
2. www.wikipedia.com/coldbootattack.html
3. P. McGregor, T. Hollebeek, A. Volynkin, and M. White, "Braving the Cold: New Methods for Preventing Cold Boot Attacks on Encryption Keys," in Black Hat Security Conference. BitArmor Systems, Inc., aug 2008.
4. M. Albrecht and C. Cid, "Cold boot key recovery by solving polynomial systems with noise," in Proceedings of the 9th international conference on Applied cryptography and network security, ser. ACNS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 57–72.
5. N. Heninger and H. Shacham, "Reconstructing RSA Private Keys from Random Key Bits," in Advances in Cryptology - CRYPTO 2009, ser. Lecture Notes in Computer Science, S. Halevi, Ed. Springer Berlin Heidelberg, 2009, vol. 5677, pp. 1–17.
6. J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Cal, A. J. Feldman, and E. W. Felten, "Least we remember: Cold boot attacks on encryption keys," in In USENIX Security Symposium, 2008.
7. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.pdf.