Practical Cold boot attack on IoT device - Case study on Raspberry Pi -

Yoo-Seung Won[1], Jong-Yeon Park[2], Dong-Guk Han[3], Shivam Bhasin[1]
[1]Temasek Laboratories, Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore
[2]System LSI Business, Samsung Electronics, 1, Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do, Republic of Korea
[3]Dept. of Financial Information Security, Kookmin University, Samsung Electronics, 1, 77 Jeongneung-ro, Seongbuk-gu, Seoul, Republic of Korea
Phone : (+65) 6592-3070, Email : yooseung.won@ntu.edu.sg

***Abstract*—Volatile memory like SDRAM, forms an integral part of any computer system. It stores variety of data including sensitive data like passwords and PIN. The data stored in SDRAM is wiped off on power-off. However, by bringing the RAM to freezing cold temperature before power off, the data can persist for several seconds, allowing recovery through cold boot attacks. In this work, we investigate the vulnerability of IoT device such as Raspberry Pi against cold boot attack for the first time. Our study found that even though the boot sequence is different from laptop, personal computer, and smartphone, we demonstrate that it is still possible to steal the RAM data, even when the bootloader is not public. The net cost of the attack was under 10 dollars and 99.99% of the RAM data was successfully recovered.**

Keywords—Cold boot attack, IoT device, SDRAM

I. Introduction

Internet of things (IoT) has been instrumental in bringing futuristic concepts like smart home and smart city to the present. However, the wide deployment of these IoT devices bring in new security and reliability issues. Device-level vulnerabilities are one of the main concerns for such IoT devices as highlighted by widespread malwares like Mirai [7]. Majority of these devices are budget constrained and thus cannot afford advanced protection mechanism.

In this work, we focus on recovery of sensitive data from volatile random access memories (RAM) of IoT devices. Many researchers have demonstrated that the RAM data persist for a short period of time (seconds) even after power-off, owing to the data remanence properties of RAM. This duration can be prolonged (to minutes) if the RAM is forced to operate at extremely low temperature [1]. RAM data can contains sensitive user information. For example, the password of OS booting is stored at RAM in PC and laptop. More precisely, after freezing the RAM data, if an adversary steals the RAM data within seconds, he/she can recover the whole RAM data in victim's product including sensitive information.

In 2008, Halderman *et al.* [1] show that the RAM data from several laptops can be recovered, while the RAM is kept at -50°C. Also, in numerous kinds of boot scenario such as preboot execution environment (PXE), USB, extensible firmware interface (EFI), and iPods, the RAM data can be retrieved in seconds. They also demonstrated that the secret key of AES and RSA can be retrieved through the recovered RAM data, which means that it is possible to regain secret information even though some data is loss. In 2013, Müller *et al.* [2] showed that cold boot attack still allows to recover RAM data, even though the RAM is dedicated to smart phone such as Galaxy Nexus device. Consequently, many secret information about PIN, secret key, and decrypted mount/data can be recovered. Unlike to previous studies, some papers [3], [4] showed that it is possible to extract the data from newer generation of memories like DDR3 and DDR4 RAM even though there is memory scrambler countermeasure. More precisely, they used the zeroization for RAM data to retrieve scramble key because the scrambler structure is very simple. Recently, in CHES conference, Albrecht *et al.* [5] showed that the secret key of post-quantum cryptography can be recovered from a memory dump obtained through cold boot attack which might have some errors.

To the best of our knowledge, the security of IoT devices has not been studied under the threat of cold boot attacks. One issue with IoT devices is that the memory is either on chip or implemented as a stacked package. Even though a stand-alone RAM is not available, we show that it still possible to recover RAM data with 99.99% success rate.

In this paper, we perform a vulnerability analysis of popular Raspberry Pi model B+ against cold boot attacks. Even though Raspberry Pi uses a ROM boot as an initial boot sequence, it is still possible to steal the RAM, owing to the fact that the RAM is enabled after ROM boot. After analyzing the boot sequence, we first analyze the attack possibility on L2 cache and SDRAM. In L2 cache, we cannot recover the cache memory, because the ROM boot has L2 cache initialization in public documentation [6]. However, the SDRAM is not initialized in ROM boot of public documentation [6], which means an adversary can try to recover the RAM. Exploiting this vulnerability, we report a practical cold boot attack on Raspberry Pi. The net cost of the attack is under $10.

The rest of the paper is organised as follows. Section II recalls the boot sequence of the targeted Raspberry Pi board.

Section III describes the cold boot vulnerability and attacker model. The experimental results and further analysis is reported in Section IV. Finally conclusions are drawn in Section V.

II. Preliminary

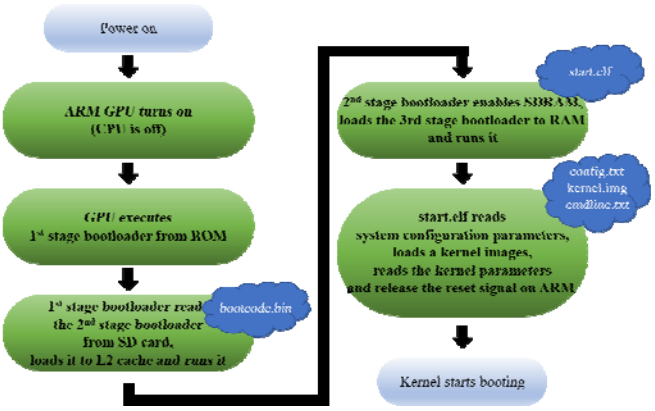

Fig. 1. Boot sequence of Raspberry Pi model B+ for SD card

To analyze vulnerability against cold boot attack, we examined the boot sequence of Raspberry Pi to investigate and determine the process which initializes the RAM data. A dump of the RAM before initialization can reveal sensitive information.

At power-up, ARM GPU turns on while CPU is still off. The GPU executes first stage bootloader from ROM. As the boot loader code is already burned into the ROM, its already executed from and ideally cannot be changed. The first boot loader reads second boot loaders from SD card, loads to L2 cache and then executes it. Next, the third boot loader code is loaded from SD card to SDRAM, and executed after the second stage boot loader enables SDRAM. Finally, `start.elf` optionally reads the system configuration from config.txt and runs `kernel.img`. Normally, `bootcode.bin` and `start.elf` are proprietary of the manufacturer and its source code is not available.

III. Cold Boot Attack against Raspberry Pi model B+

In this section, we introduce adversary assumption and our experimental environment to perform cold boot attack. Finally, we demonstrate that the Raspberry Pi is still vulnerable to cold boot attack by demonstrating memory content recovery.

A. Adversary Assumption

We define the adversary assumption to clarify the motivation for cold boot attack in this section. The final goal of an adversary is to recover RAM contents on Raspberry Pi under the following setting:

- No one can access the RAM while victim's program (burned on a SD card) is in operation except for an authorized person.
- An adversary can physically access the Raspberry Pi and can replace the victim's SD card by adversary SD card and vice versa.
- Victim SD card reveals no sensitive information owing to lack of access rights.

Considering the above assumptions, an adversary cannot acquire the secret information even if an adversary has access to the Raspberry Pi including victim's SD card because the SD card is protected by available security solution. Ideally, when adversary removes the victim's SD card and inserts adversary SD card to run his own application, sensitive information corresponding to victim program is erased from the RAM. The adversary aims to use cold boot attack to recover RAM data containing sensitive information about victim program.

B. Threat of vulnerabilities for cold boot attack on IoT device

In [1], the authors demonstrate that the RAM can be recovered utilizing cold boot attack when the RAM is not dedicated to the laptop. In other words, the RAM can be recovered regardless of the security strength of laptop if the RAM keeps at very low temperature while physically transferring it to another laptop. That is, even though the laptop has the strong security solution, the RAM can be retrieved if the RAM can be physically removed. Of course, the authors in [1] show that it is possible to recover the RAM without removing the RAM from laptop and exploiting the boot step. Nevertheless, they cannot retrieve the RAM data if boot step is locked from an adversary.

On the other hand, the RAM in IoT device cannot be physically separated from its CPU in general. In other words, the RAM is stacked over the CPU and packed. Unlike to the RAM of laptop, cold boot attack can be prevented by initialization of the RAM in the physical initial step because the one cannot separates the RAM from IoT device as well as fix the initial boot step. In other words, it is hard to defend the cold boot attack if there is no RAM initialization in the physical boot sequence.

IV. Experimental results against Raspberry Pi

In this section we describe the experimental setup and obtained results..

A. Experimental environment

TABLE I. Specification for main target

Target	Raspberry Pi model B+
SoC	Broadcom BCM2835 (Technology: 65nm)
CPU	700 MHz ARM1176JZF-S single core
GPU	Broadcom VideoCore IV @ 250MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1 (with license), 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (L2 cache of 128 KB)
Memory (SDRAM)	SDRAM 512MB (SAMSUNG k4p4g324eq-rgc2)

We first present our main target and experimental environment for cold boot attack.

To perform cold boot attack, we should recognize where the target chip is located on the PCB. Next, we use air duster spray to freeze the RAM. The detailed specification of the target is listed in Tab. I. The RAM is stacked upon the processor. As shown in Fig. 3, we use air duster to freeze the RAM before rebooting with a secondary SD card which is adversary's SD card. This card runs a program to show RAM content at boot-up via monitor or dump the content in a USB memory for further analysis. Thus, the components required to perform cold boot attack are air duster and SD card whose combined cost is under 10 dollars. In summary, the process of our cold boot attack is as below.

1) When victim's program is running on the Raspberry Pi, the secret information is temporarily saved in the RAM.

2) After an adversary approaches to the Raspberry Pi, he/she temporarily freezes the RAM which is attached to the Raspberry Pi using some tools such as air duster.

3) While the RAM keeps very low temperature, the adversary replaces victim's SD card with his/her SD card containing source code to recover the RAM data from last victim's program execution

We investigated the possibility of performing cold boot attack against L2 cache memory and SDRAM on Raspberry Pi model B+. The memory is read through USB communication in bare-metal programming step. In order to avoid the RAM initialization or overlap, the implementation for the RAM recovery is based on bare-metal programming.

The first target for cold boot attack is L2 cache memory as it is activated first in the boot sequence. However, as reported in the public documentation [6], the cache memory is initialized in the first boot loader stage, thus preventing any

Fig. 2. A bitmap image is loaded into memory, then cut power for varying period of time at -30ºC. After 5 seconds (left), the image is almost indistinguishable from the original. It gradually becomes more degraded, a shown after 30 seconds, 60 seconds, and 5 minutes.

cold boot attack. Next, we investigate if SDRAM can be exploited.

Fig. 3. Frozen Raspberry Pi model B+ (Left) and air duster (Right)

B. *Experimental result for SDRAM*

As a next target, we focus on data recovery from SDRAM. As mentioned previously, the third stage boot loader (`start.elf`) is loaded from SD card to SDRAM. This binary source code is not public. Thus, we need to modify the `kernel.img` to read SDRAM via USB communication. If `start.elf` contains the SDRAM initialization, cold boot attack would fail. It would require the adversary to disable SDRAM initialization in the source code of `start.elf`. However, we observed that `start.elf` does not perform the initialization, thus exposing the RAM data.

To compute recovery success metric, we upload the famous Mona Lisa image like as [1], and then recover it after freezing SDRAM and turning off the power while varying the decay time and temperature. As shown in Fig. 2, the recovery ratio is quite high. It is difficult to visually distinguish the difference between original and recovered image. When compared pixel wise, recovery ratio is about 99.718%, *i.e.* only 0.282% of recovered pixel data was erroneous. Actually, the number of error bits is 3371 bits on total 1195360 bits at a temperature -35ºC. If it keeps the lower temperature, we can acquire 99.999% recovery ratio, but that would increase the cost of the attack..

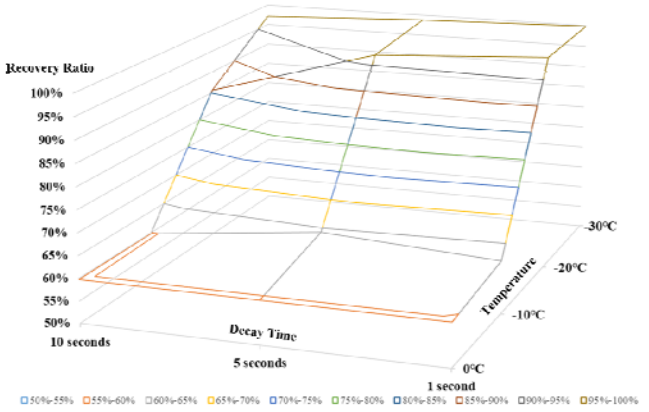

Fig. 4. Recovery ratio of cold boot attack for varying decay time and temperature

We investigate recovery ratio of the RAM data at various temperatures and decay time. The minimum recovery ratio is 59.7%. This represent the number of zeros in the image pixels when compared to an image with all pixels at zero. From a visual perspective, one sees nothing in the recovered data however one expects to see the image of Mona Lisa. Moreover, the normal temperature of chip surface is 33.6 ºC. As shown in Fig. 4, there is the recovery ratio is really low at 0ºC and -10ºC, which is very similar to the result of 33.6 ºC. However, when the temperature goes lower than -20ºC, the recovery ratio is higher than 99% at 1 second decay time. In terms of decay time, the recovery ratio drops to about 85.5 % from 99.2 % at -20ºC, at a decay time as low as 10 seconds. Moreover, it is not hard to make -20ºC if we just spray the air duster to chip surface for a short time. However, the recovery ratio keeps about 99% at -30ºC regardless of our decay time. This temperature is easily achievable with the air duster. As a result, we can achieve a recovery ratio as high as 99% for a considerable time if the temperature keeps around -30ºC.

Additionally, the total time to recover the 512 MB RAM takes about 5 minutes. Of course, it is not required to maintain the low temperature while recovering the RAM contents. We need to only milliseconds while replacing the SD card from victim's one.

C. Mitigation

The presented cold boot attack can be a serious threat to IoT devices where security features are often nonexistent. The use of older RAM technologies motivated by low-cost requirements further adds to the problem. The problem can be solved by adoption of secure boot with secure initialization of memory at the firmware level. This is even important for the future generations of devices which are likely to adopt magnetic memory based non-volatile memories as RAM and will not lose program data even on power off unless triggered by the secure boot or secure shutdown feature.

V. Conclusion

Even though the boot sequence of main CPU dedicated in IoT device is totally different from PC, laptop, and smartphone, it still remains the physical vulnerability of RAM against cold boot attack. If the RAM in IoT device contains sensitive information, it is very easy to reveal for an equipped adversary. The total cost of the reported attack is under $10, which makes it a serious threat against billions of IoT devices deployed into the wild. For the further work, we will expand our attack to the other IoT device. We noticed that the boot sequence for newer Raspberry Pi such as Pi 3 and 4 is identical to the target we studied in this paper. These new targets support LPDDR3 and LPDDR4 making it an interesting case study. We also plan to investigate the vulnerability of disk encryption in IoT devices against cold boot attack.

Acknowledgment

This research is supported in parts by the National Research Foundation, Singapore, under its National Cybersecurity Research & Development Programme / Cyber-Hardware Forensic & Assurance Evaluation R&D Programme (Award: NRF2018NCR-NCR009-0001).

References

[1] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest We Remember: Cold Boot Attacks on Encryption Keys", Proceedings of the 17th USENIX Security Symposium, pp. 45-60, July, 2008.

[2] T. Müller, M. Spreizenbarth, and F. C. Freiling, "Frost: Forensic Recovery of Scrambled Telephones", Proceedings of the 11th international conference on Applied Cryptography and Network Security, pp.373-388, June, 2013.

[3] J. Bauer, M. Gruhn, and F. C. Freiling, "Lest we forget: Cold-boot attacks on scrambled DDR3 memory", Proceedings of the Third Annual DFRWS Europe, pp. S64-S74, Jan. 2016.

[4] S. F. Yitbarek, M. T. Aga, R. Das, and T. Austin, "Cold Boot Attacks are Still Hot: Security Analysis of Memory Scramblers in Modern Processors", 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 313-324, October, 2017.

[5] M. R. Albrecht, A. Deo, and K. G. Paterson, "Cold Boot Attacks on Ring and Module LWE Keys Under the NTT", IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(3), pp. 173-213, 2018.

[6] Broadcom BCM2835 ARM Peripherals, available at https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm28 35/BCM2835-ARM-Peripherals.pdf.

[7] M. Antonakakis et al., "Understanding the mirai botnet",26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 1093-1110, 2017.