[redacted] nothing [redacted]
[redacted] to [redacted]
[redacted] hide [redacted]

Chaos Communication Congress
Berlin - bcc, 27.-30.12.2008
http://events.ccc.de/congress/2008/

.....

Proceedings [redacted] Daten[redacted] Solar-powering [redacted] open source PLC tool[redacted] 202c StGB [redacted] Smart Card [redacted] Sys[redacted]ing hackerspaces[redacted] Cyborgs and [redacted] Terrorist All-Stars [redacted] Staat [redacted] Virenprogrammierer? [redacted] and Georgia? [redacted] digitale Intimsphäre [redacted] Reverse Engineering [redacted] Mühsams Tagebücher [redacted] Hacking the iPhone [redacted] Beyond Asimov [redacted] MSP430 BSL [redacted] Cold Boot [redacted] Locating Mobile Phones [redacted] European security [redacted] DNS [redacted] coreboot [redacted] Garage Doors [redacted] Gödel Jahres[redacted] [redacted]-Privacy [redacted] Climate [redacted]-Disk-Encryption [redacted] Internet Applications [redacted] Hacking [redacted] Atmosphere [redacted] Symbian [redacted] PHP [redacted] Music [redacted] Handschellen [redacted] smart-phone [redacted] MICA*-based wireless [redacted] TCP Denial [redacted] Short Attention [redacted] Security [redacted] Trust Situation [redacted] Swarm Robotics [redacted] Your Life [redacted] All your base [redacted] Banking Malware [redacted] Infinite Library [redacted] Wii [redacted] Fail Tricks: [redacted] Fnord News [redacted] Blinkenlights [redacted] Holodeck! [redacted] Soviet Unterz[redacted] your own GSM [redacted] Spuren [redacted] eVoting [redacted] TI EZ430U [redacted] Brother [redacted] Neusprech [redacted] Privacy [redacted] semantic web [redacted] stream cipher [redacted] Commodore 64 [redacted] Hacking [redacted] Botnets [redacted] anonymity [redacted] Tor Squeezing Attack [redacted] phones [redacted] Anonymous VPN [redacted] SWF [redacted] Attacks with Office Doc[redacted] Personalausweis Cisco IOS [redacted] Revolution [redacted] Wikileaks [redacted] Jeopardy [redacted] got owned [redacted] [redacted] technology sucks [redacted] Privacy [redacted] Quadrature du Net [redacted] RNG in [redacted] OpenSSL package Crafting [redacted] theoretical possible [redacted] social contacts [redacted] RFID [redacted] Pflanzenhacken [redacted] Nightmares [redacted]

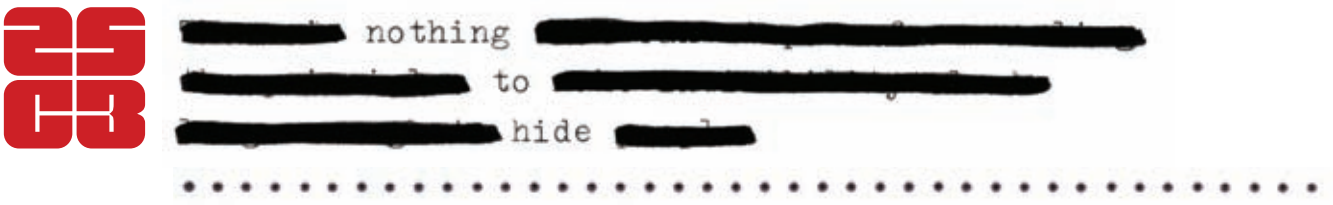
Peter Fnord

Für Wolfgang.

[redacted] nothing [redacted]
[redacted] to [redacted]
[redacted] hide [redacted]

Chaos Communication Congress
Berlin - bcc, 27.-30.12.2008
http://events.ccc.de/congress/2008/

Proceedings



Proceedings of the 25th Chaos Communication Congress
December, 27th - 30th 2008, Berlin Congress Center, Alexanderplatz.

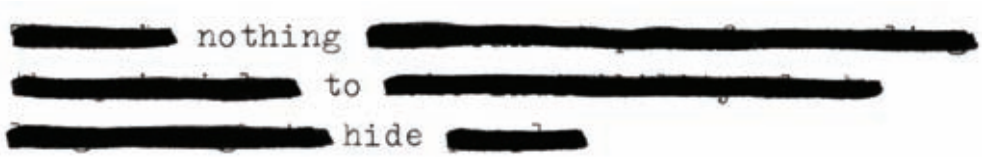
25C3: An event of the Chaos Computer Club.
https://events.ccc.de/congress/2008/

Cover:	Evelyn Schubert
Data-Gardening:	Sven Klemm
Editor:	Matthias 'wetterfrosch' Mehldau
Publisher:	Art d'Ameublement Marktstraße 18 in 33602 Bielefeld
Vertrieb:	FoeBuD e.V. Unterstützungsbedarf Marktstraße 18 in 33602 Bielefeld, https://shop.foebud.org/
Font-Family:	Myriad Pro
ISBN:	978-3-934636-06-4
ISSN:	1867-8556

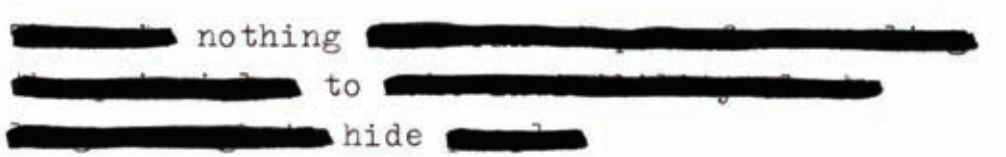
Program planning
under the patronage of the Wau Holland Foundation.

1st edition, 400 copies

Last update:	December, 15th 2008
Printer:	Druckerei Wollenhaupt Unter dem Felsenkeller 30 in 37247 Großalmerode
Paper:	Printed on FSC-certified paper by a FSC-certified printer.
Environment:	Climate-neutral print. The for the production this book emitted carbon-dioxide was compensated in cooperation with the natureOffice.de-program which supports facilities for renewable energy in developing countries. Strömchen!



Day 2008-12-27



18:30 CET

Das Grundrecht auf digitale Intimsphäre ... p. 31
Festplattenbeschlagnahme in neuem Licht
Saal 1: Society with Ulf Buermeyer, 46halbe

Chip Reverse Engineering ... p. 32
Saal 2: Hacking with Karsten Nohl, starbug *... see paper on p. 155*

Erich Mühsams Tagebücher in der Festungshaft ... p. 33
Ein Idylle aus der Analogsteinzeit der Überwachung
Saal 3: Society with Johannes Ullmaier

20:30 CET

Hacking the iPhone ... p. 34
Pwning Apple's Mobile Internet Device
Saal 1: Hacking with pytey, MuscleNerd, planetbeing

Beyond Asimov - Laws for Robots ... p. 35
Developing rules for autonomous systems
Saal 2: Society with Frank Rieger

Cracking the MSP430 BSL ... p. 36
Part Two *... see paper on p. 165*
Saal 3: Hacking with Travis Goodspeed

21:45 CET

Advanced memory forensics: The Cold Boot Attacks ... p. 37
Recovering keys and other secrets after power off *... see paper on p. 133*
Saal 1: Hacking with Jake

Locating Mobile Phones using SS7 ... p. 38
Saal 2: Hacking with Tobias Engel

Collapsing the European security architecture ... p. 39
More security-critical behaviour in Europe! *... see paper on p. 263*
Saal 3: Society with Gipfelsoli

23:00 CET

Why were we so vulnerable to the DNS vulnerability? ... p. 46
Saal 1: Hacking with Effugas

coreboot: Beyond The Final Frontier ... p. 47
Open source BIOS replacement with a radical approach to boot.
Saal 2: Hacking with Peter Stuge

Messing Around with Garage Doors ... p. 48
Breaking Remote Keyless Entry Systems with Power Analysis
Saal 3: Hacking with Timo Kasper, Thomas Eisenbarth

24:00 CET

Kurt Gödel – I do not fit into this century ... p. 49
Ein audiovisuelles Live-Feature
Saal 1: Culture with 46halbe, Marcus Richter, Ina Kwasniewski, Kai Kittler

[redacted] nothing [redacted]

[redacted] to [redacted]

[redacted] hide [redacted]

Chaos Communication Congress
Berlin - bcc, 27.-30.12.2008
http://events.ccc.de/congress/2008/

.....

.....
Lectures .
.....

Day 1

2008-12-27 | 11:30 CET | 01:00 h | Saal 1 | lecture | Society

Datenpannen

Forderungen nach dem Jahr der Datenverbrechen

Wer nichts zu verbergen hat, hat nichts zu befürchten? Die zuständigen Mitarbeiter halten sich strikt an das Gesetz? Überwachung hat für die Betroffenen keine negativen Folgen? Im Jahr 2008 sind diese Irrtümer so häufig widerlegt worden wie noch nie: Datenskandale bei LIDL, Telekom und dutzenden anderen, per Internet zugängliche Meldedaten, Massenverkauf von Bank- und Telefondaten – eine Liste ohne Ende im Datenskandaljahr 2008.

Wir nehmen die wichtigsten deutschen Datenskandale des Jahres unter die Lupe. Was war die Ursache? Welche Rechte habe ich als (möglicherweise) Betroffener? Und wie kann man das in Zukunft verhindern?

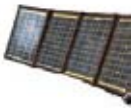
Damit ist es aber nicht getan, wir wollen gleichzeitig die Forderungen formulieren und diskutieren, die aus diesen Vorfällen folgen. Mit Schäubles neuem kleinen Datenschutzgesetz-Update wird sich jedenfalls nichts grundlegend ändern, daher ist es Zeit, unsere Vorstellungen für die Zukunft des Datenschutzes zu artikulieren.

> http://events.ccc.de/congress/2008/Fahrplan/events/2814.en.html

46halbe

Patrick Breyer

2008-12-27 | 11:30 CET | 01:00 h | Saal 2 | lecture | Making | See paper on p. 225!



Solar-powering your Geek Gear

Alternative and mobile power for all your little toys

This talk will show you how to solar-power your laptop, PDA, cell phone, portable fridge or almost any other small device. Topics discussed include choosing the right solar panel, using (or not using) a voltage regulator, buffering the energy, some real applications as well as instructions on how to build a small and simple device to measure your power and energy savings.

Do you want to use your laptop in the garden or in the park without needing to pull long cords? Need to recharge your cell phone, PDA or camera in the wilderness? Are you just curious about solar energy or just want to keep your drinks cool on a hot summer day? Well, then you should attend this lecture!

Contents of the lecture:
• Motivation
• Decide what you want to have powered
• Choosing the appropriate solar panel
• Connectors, adapters, plugs
• The universal Buck/Boost voltage regulator
• Building your own device to measure voltage, current, power and energy
• Applications

> http://events.ccc.de/congress/2008/Fahrplan/events/2904.en.html

script

2008-12-27 | 12:45 CET | 01:00 h | Saal 2 | lecture | Community

U23

The Hackerspace's Junior Academy

Organize and operate a workshop for young people. Show them how your hackerspace works. Gain their attraction in having fun with hardware, electronics, microprocessors, software or hacking. Become known to new persons. Create networks of brains for new, cool projects. Let them experience the amazing power of teamwork!

In 2002, some people at the Chaos Computer Club Cologne discussed, how they could attract young people, especially students and pupils, to the ideas and lifestyle of a hacker, and gain new members. The result of this discussion was the concept for a project directed at young nerds and geeks, featuring a challenge which is only solvable as a group. This idea turned out to be so successful, that up to now there have been six recurrences.

The talk will explain the main design patterns which evolved in this six year period. We will introduce our motivation and goals for this project, and present the patterns for preparation, implementation and review.

We will save some time at the end for a short Q&A session, and fd0 is available in the hardware hacking room (in the basement) for a chat.

http://koeln.ccc.de/u23 *U23 at CCC Cologne*

> http://events.ccc.de/congress/2008/Fahrplan/events/2827.en.html

fd0

Lars Weiler

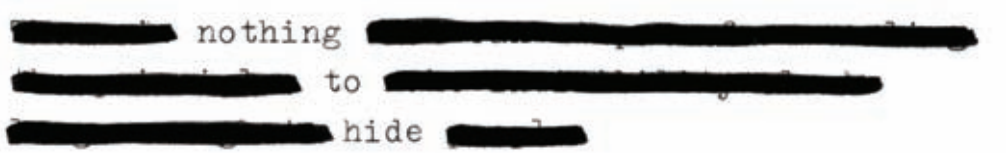
red_hood

[redacted] nothing [redacted]
[redacted] to [redacted]
[redacted] hide [redacted]

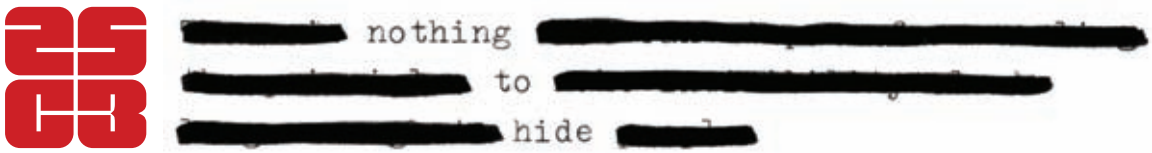
Chaos Communication Congress
Berlin - bcc, 27.-30.12.2008
http://events.ccc.de/congress/2008/

.....

.....
Papers .
.....



Papers



Papers

Hacking

This paper was released February 21, 2008 and published in *Proc. 2008 USENIX Security Symposium*. For the most recent revision, related source code, and videos of demonstration attacks, visit http://citp.princeton.edu/memory.

Lest We Remember: Cold Boot Attacks on Encryption Keys

J. Alex Halderman*, Seth D. Schoen†, Nadia Heninger*, William Clarkson*, William Paul‡, Joseph A. Calandrino*, Ariel J. Feldman*, Jacob Appelbaum, and Edward W. Felten*

*Princeton University †Electronic Frontier Foundation ‡Wind River Systems

{jhalderm, nadiah, wclarkso, jcalandr, ajfeldma, felten}@cs.princeton.edu

schoen@eff.org, wpaul@windriver.com, jacob@appelbaum.net

Abstract

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. We show that this phenomenon limits the ability of an operating system to protect cryptographic key material from an attacker with physical access. We use cold reboots to mount successful attacks on popular disk encryption systems using no special devices or materials. We experimentally characterize the extent and predictability of memory remanence and report that remanence times can be increased dramatically with simple cooling techniques. We offer new algorithms for finding cryptographic keys in memory images and for correcting errors caused by bit decay. Though we discuss several strategies for partially mitigating these risks, we know of no simple remedy that would eliminate them.

1 Introduction

Most security experts assume that a computer's memory is erased almost immediately when it loses power, or that whatever data remains is difficult to retrieve without specialized equipment. We show that these assumptions are incorrect. Ordinary DRAMs typically lose their contents gradually over a period of seconds, even at standard operating temperatures and even if the chips are removed from the motherboard, and data will persist for minutes or even hours if the chips are kept at low temperatures. Residual data can be recovered using simple, nondestructive techniques that require only momentary physical access to the machine.

We present a suite of attacks that exploit DRAM remanence effects to recover cryptographic keys held in memory. They pose a particular threat to laptop users who rely on disk encryption products, since an adversary who steals a laptop while an encrypted disk is mounted could employ our attacks to access the contents, even if the computer is screen-locked or suspended. We demonstrate this risk by defeating several popular disk encryption systems, including BitLocker, TrueCrypt, and FileVault, and we expect many similar products are also vulnerable.

While our principal focus is disk encryption, any sensitive data present in memory when an attacker gains physical access to the system could be subject to attack. Many other security systems are probably vulnerable. For example, we found that Mac OS X leaves the user's login password in memory, where we were able to recover it, and we have constructed attacks for extracting RSA private keys from Apache web servers.

As we discuss in Section 2, certain segments of the computer security and semiconductor physics communities have been conscious of DRAM remanence effects for some time, though strikingly little about them has been published. As a result, many who design, deploy, or rely on secure systems are unaware of these phenomena or the ease with which they can be exploited. To our knowledge, ours is the first comprehensive study of their security consequences.

Highlights and roadmap In Section 3, we describe experiments that we conducted to characterize DRAM remanence in a variety of memory technologies. Contrary to the expectation that DRAM loses its state quickly if it is not regularly refreshed, we found that most DRAM modules retained much of their state without refresh, and even without power, for periods lasting thousands of refresh intervals. At normal operating temperatures, we generally saw a low rate of bit corruption for several seconds, followed by a period of rapid decay. Newer memory technologies, which use higher circuit densities, tended to decay more quickly than older ones. In most cases, we observed that almost all bits decayed at predictable times

and to predictable "ground states" rather than to random values.

We also confirmed that decay rates vary dramatically with temperature. We obtained surface temperatures of approximately $-50\,^{\circ}$C with a simple cooling technique: discharging inverted cans of "canned air" duster spray directly onto the chips. At these temperatures, we typically found that fewer than 1% of bits decayed even after 10 minutes without power. To test the limits of this effect, we submerged DRAM modules in liquid nitrogen (ca. $-196\,^{\circ}$C) and saw decay of only 0.17% after 60 minutes out of the computer.

In Section 4, we present several attacks that exploit DRAM remanence to acquire memory images from which keys and other sensitive data can be extracted. Our attacks come in three variants, of increasing resistance to countermeasures. The simplest is to reboot the machine and launch a custom kernel with a small memory footprint that gives the adversary access to the retained memory. A more advanced attack briefly cuts power to the machine, then restores power and boots a custom kernel; this deprives the operating system of any opportunity to scrub memory before shutting down. An even stronger attack cuts the power and then transplants the DRAM modules to a second PC prepared by the attacker, which extracts their state. This attack additionally deprives the original BIOS and PC hardware of any chance to clear the memory on boot. We have implemented imaging kernels for use with network booting or a USB drive.

If the attacker is forced to cut power to the memory for too long, the data will become corrupted. We propose three methods for reducing corruption and for correcting errors in recovered encryption keys. The first is to cool the memory chips prior to cutting power, which dramatically reduces the error rate. The second is to apply algorithms we have developed for correcting errors in private and symmetric keys. The third is to replicate the physical conditions under which the data was recovered and experimentally measure the decay properties of each memory location; with this information, the attacker can conduct an accelerated error correction procedure. These techniques can be used alone or in combination.

In Section 5, we explore the second error correction method: novel algorithms that can reconstruct cryptographic keys even with relatively high bit-error rates. Rather than attacking the key directly, our methods consider values derived from it, such as key schedules, that provide a higher degree of redundancy. For performance reasons, many applications precompute these values and keep them in memory for as long as the key itself is in use. To reconstruct an AES key, for example, we treat the decayed key schedule as an error correcting code and find the most likely values for the original key. Applying this method to keys with 10% of bits decayed, we can reconstruct nearly any 128-bit AES key within a few seconds. We have devised reconstruction techniques for AES, DES, and RSA keys, and we expect that similar approaches will be possible for other cryptosystems. The vulnerability of precomputation products to such attacks suggests an interesting trade-off between efficiency and security. In Section 6, we present fully automatic techniques for identifying such keys from memory images, even in the presence of bit errors.

We demonstrate the effectiveness of these attacks in Section 7 by attacking several widely used disk encryption products, including BitLocker, TrueCrypt, and FileVault. We have developed a fully automated demonstration attack against BitLocker that allows access to the contents of the disk with only a few minutes of computation. Notably, using BitLocker with a Trusted Platform Module (TPM) sometimes makes it *less* secure, allowing an attacker to gain access to the data even if the machine is stolen while it is completely powered off.

It may be difficult to prevent all the attacks that we describe even with significant changes to the way encryption products are designed and used, but in practice there are a number of safeguards that can provide partial resistance. In Section 8, we suggest a variety of mitigation strategies ranging from methods that average users can apply today to long-term software and hardware changes. Each remedy has limitations and trade-offs. As we conclude in Section 9, it seems there is no simple fix for DRAM remanence vulnerabilities.

Online resources A video demonstration of our attacks and source code for some of our tools are available at http://citp.princeton.edu/memory.

2 Previous Work

Previous researchers have suggested that data in DRAM might survive reboots, and that this fact might have security implications. To our knowledge, however, ours is the first security study to focus on this phenomenon, the first to consider how to reconstruct symmetric keys in the presence of errors, the first to apply such attacks to real disk encryption systems, and the first to offer a systematic discussion of countermeasures.

We owe the suggestion that modern DRAM contents can survive cold boot to Pettersson [33], who seems to have obtained it from Chow, Pfaff, Garfinkel, and Rosenblum [13]. Pettersson suggested that remanence across cold boot could be used to acquire forensic memory images and obtain cryptographic keys, although he did not experiment with the possibility. Chow *et al.* discovered this property in the course of an experiment on data lifetime in running systems. While they did not exploit the

	Memory Type	Chip Maker	Memory Density	Make/Model	Year
A	SDRAM	Infineon	128Mb	Dell Dimension 4100	1999
B	DDR	Samsung	512Mb	Toshiba Portégé	2001
C	DDR	Micron	256Mb	Dell Inspiron 5100	2003
D	DDR2	Infineon	512Mb	IBM T43p	2006
E	DDR2	Elpida	512Mb	IBM x60	2007
F	DDR2	Samsung	512Mb	Lenovo 3000 N100	2007

Table 1: Test systems we used in our experiments

property, they remark on the negative security implications of relying on a reboot to clear memory.

In a recent presentation, MacIver [31] stated that Microsoft considered memory remanence attacks in designing its BitLocker disk encryption system. He acknowledged that BitLocker is vulnerable to having keys extracted by cold-booting a machine when it is used in "basic mode" (where the encrypted disk is mounted automatically without requiring a user to enter any secrets), but he asserted that BitLocker is not vulnerable in "advanced modes" (where a user must provide key material to access the volume). He also discussed cooling memory with dry ice to extend the retention time. MacIver apparently has not published on this subject.

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 [29], a DRAM showed no data loss for a full week without refresh when cooled with liquid nitrogen. Anderson [2] briefly discusses remanence in his 2001 book:

> [A]n attacker can ... exploit ... memory remanence, the fact that many kinds of computer memory retain some trace of data that have been stored there. ... [M]odern RAM chips exhibit a wide variety of memory remanence behaviors, with the worst of them keeping data for several seconds even at room temperature...

Anderson cites Skorobogatov [40], who found significant data retention times with *static* RAMs at room temperature. Our results for modern DRAMs show even longer retention in some cases.

Anderson's main focus is on "burn-in" effects that occur when data is stored in RAM for an extended period. Gutmann [22, 23] also examines "burn-in," which he attributes to physical changes that occur in semiconductor memories when the same value is stored in a cell for a long time. Accordingly, Gutmann suggests that keys should not be stored in one memory location for longer than several minutes. Our findings concern a different phenomenon: the remanence effects we have studied occur in modern DRAMs even when data is stored only momentarily. These effects do not result from the kind of physical changes that Gutmann described, but rather from the capacitance of DRAM cells.

Other methods for obtaining memory images from live systems include using privileged software running under the host operating system [43], or using DMA transfer on an external bus [19], such as PCI [12], mini-PCI, Firewire [8, 15, 16], or PC Card. Unlike these techniques, our attacks do not require access to a privileged account on the target system, they do not require specialized hardware, and they are resistant to operating system countermeasures.

3 Characterizing Remanence Effects

A DRAM cell is essentially a capacitor. Each cell encodes a single bit by either charging or not charging one of the capacitor's conductors. The other conductor is hard-wired either to power or to ground, depending on the cell's address within the chip [37, 23].

Over time, charge will leak out of the capacitor, and the cell will lose its state or, more precisely, it will decay to its *ground state*, either zero or one depending on whether the fixed conductor of the capacitor is hard-wired to ground or power. To forestall this decay, the cell must be *refreshed*, meaning that the capacitor must be re-charged to hold its value. Specifications for DRAM chips give a *refresh time*, which is the maximum interval that is supposed to pass before a cell is refreshed. The standard refresh time (usually on the order of milliseconds) is meant to achieve extremely high reliability for normal computer operations where even infrequent bit errors could cause serious problems; however, a failure to refresh any individual DRAM cell within this time has only a tiny probability of actually destroying the cell's contents.

We conducted a series of experiments to characterize DRAM remanence effects and better understand the security properties of modern memories. We performed trials using PC systems with different memory technologies, as shown in Table 1. These systems included models from several manufacturers and ranged in age from 9 years to 6 months.

3.1 Decay at operating temperature

Using a modified version of our PXE memory imaging program (see Section 4.1), we filled representative memory regions with a pseudorandom pattern. We read back these memory regions after varying periods of time without refresh and under different temperature conditions, and measured the error rate of each sample. The error rate is the number of bit errors in each sample (the Hamming distance from the pattern we had written) divided by the total number of bits we measured. Since our pseudorandom test pattern contained roughly equal numbers of zeros and ones, we would expect fully decayed memory to have an error rate of approximately 50% .

Our first tests measured the decay rate of each memory module under normal operating temperature, which ranged from 25.5 °C to 44.1 °C, depending on the machine (see Figures 1, 2, and 3). We found that the dimensions of the decay curves varied considerably between machines, with the fastest exhibiting complete data loss in approximately 2.5 seconds and the slowest taking an average of 35 seconds. However, the decay curves all display a similar shape, with an initial period of slow decay, followed by an intermediate period of rapid decay, and then a final period of slow decay.

We calculated best fit curves to the data using the logistic function because MOSFETs, the basic components of a DRAM cell, exhibit a logistic decay curve. We found that 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 our attacks. We ascribe this trend to the increasing density of the DRAM cells as the technology improves; in general, memory with higher densities have a shorter window where data is recoverable. While this trend might make DRAM retention attacks more difficult in the future, manufacturers also generally seek to *increase* retention times, because DRAMs with long retention require less frequent refresh and have lower power consumption.

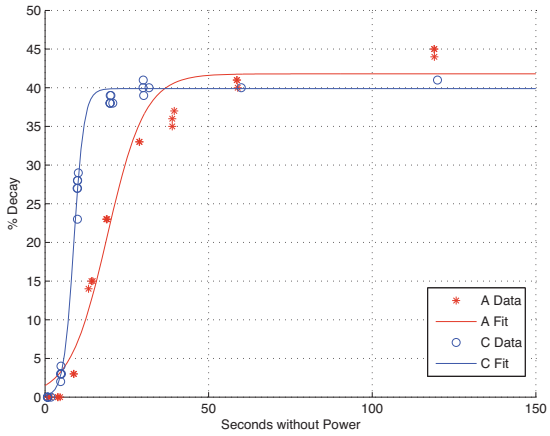


Figure 1: Machines A and C

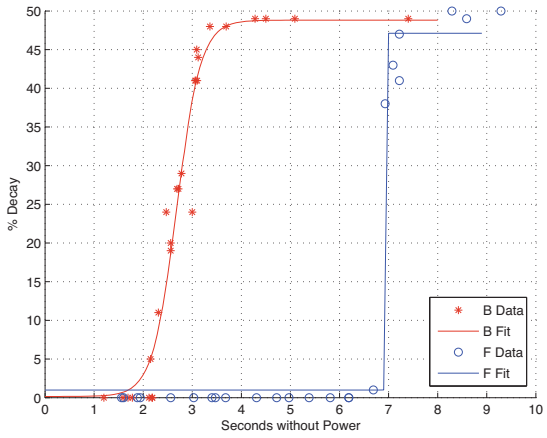


Figure 2: Machines B and F

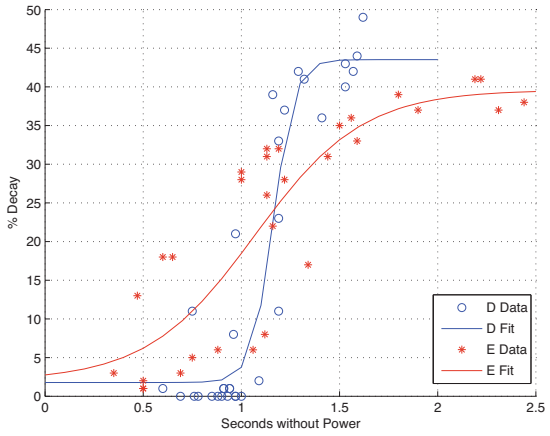


Figure 3: Machines D and E

3.2 Decay at reduced temperature

It has long been known that low temperatures can significantly increase memory devices' retention times [29, 2, 46, 23, 41, 40]. To measure this effect, we performed a second series of tests using machines A–D.

In each trial, we loaded a pseudorandom test pattern into memory, and, with the computer running, cooled the memory module to approximately −50 °C. We then powered off the machine and maintained this temperature until power was restored. We achieved these temperatures using commonly available "canned air" duster products (see Section 4.2), which we discharged, with the can inverted, directly onto the chips.

	Seconds w/o power	Error % at operating temp.	Error % at $-50\,^{\circ}$C
A	60	41	(no errors)
	300	50	0.000095
B	360	50	(no errors)
	600	50	0.000036
C	120	41	0.00105
	360	42	0.00144
D	40	50	0.025
	80	50	0.18

Table 2: Effect of cooling on error rates

As expected, we observed a significantly lower rate of decay under these reduced temperatures (see Table 2). On all of our sample DRAMs, the decay rates were low enough that an attacker who cut power for 60 seconds would recover 99.9% of bits correctly.

As an extreme test of memory cooling, we performed another experiment using liquid nitrogen as an additional cooling agent. We first cooled the memory module of Machine A to $-50\,^{\circ}$C using the "canned air" product. We then cut power to the machine, and quickly removed the DRAM module and placed it in a canister of liquid nitrogen. We kept the memory module submerged in the liquid nitrogen for 60 minutes, then returned it to the machine. We measured only 14,000 bit errors within a 1 MB test region (0.17% decay). This suggests that, even in modern memory modules, data may be recoverable for hours or days with sufficient cooling.

3.3 Decay patterns and predictability

We observed that the DRAMs we studied tended to decay in highly nonuniform patterns. While these patterns varied from chip to chip, they were very predictable in most of the systems we tested. Figure 4 shows the decay in one memory region from Machine A after progressively longer intervals without power.

There seem to be several components to the decay patterns. The most prominent is a gradual decay to the "ground state" as charge leaks out of the memory cells. In the DRAM shown in Figure 4, blocks of cells alternate between a ground state of 0 and a ground state of 1, resulting in the series of horizontal bars. Other DRAM models and other regions within this DRAM exhibited different ground states, depending on how the cells are wired.

We observed a small number of cells that deviated from the "ground state" pattern, possibly due to manufacturing variation. In experiments with 20 or 40 runs, a few "retrograde" cells (typically $\sim 0.05\%$ of memory cells, but larger in a few devices) always decayed to the opposite value of the one predicted by the surrounding ground state pattern. An even smaller number of cells decayed in different directions across runs, with varying probabilities.

Apart from their eventual states, the *order* in which different cells decayed also appeared to be highly predictable. At a fixed temperature, each cell seems to decay after a consistent length of time without power. The relative order in which the cells decayed was largely fixed, even as the decay times were changed by varying the temperature. This may also be a result of manufacturing variations, which result in some cells leaking charge faster than others.

To visualize this effect, we captured degraded memory images, including those shown in Figure 4, after cutting power for intervals ranging from 1 second to 5 minutes, in 1 second increments. We combined the results into a video (available on our web site). Each test interval began with the original image freshly loaded into memory. We might have expected to see a large amount of variation between frames, but instead, most bits appear stable from frame to frame, switching values only once, after the cell's decay interval. The video also shows that the decay intervals themselves follow higher order patterns, likely related to the physical geometry of the DRAM.

3.4 BIOS footprints and memory wiping

Even if memory contents remain intact while power is off, the system BIOS may overwrite portions of memory when the machine boots. In the systems we tested, the BIOS overwrote only relatively small fractions of memory with its own code and data, typically a few megabytes concentrated around the bottom of the address space.

On many machines, the BIOS can perform a destructive memory check during its Power-On Self Test (POST). Most of the machines we examined allowed this test to be disabled or bypassed (sometimes by enabling an option called "Quick Boot").

On other machines, mainly high-end desktops and servers that support ECC memory, we found that the BIOS cleared memory contents without any override option. ECC memory must be set to a known state to avoid spurious errors if memory is read without being initialized [6], and we believe many ECC-capable systems perform this wiping operation whether or not ECC memory is installed.

ECC DRAMs are not immune to retention effects, and an attacker could transfer them to a non-ECC machine that does not wipe its memory on boot. Indeed, ECC memory could turn out to *help* the attacker by making DRAM more resistant to bit errors.

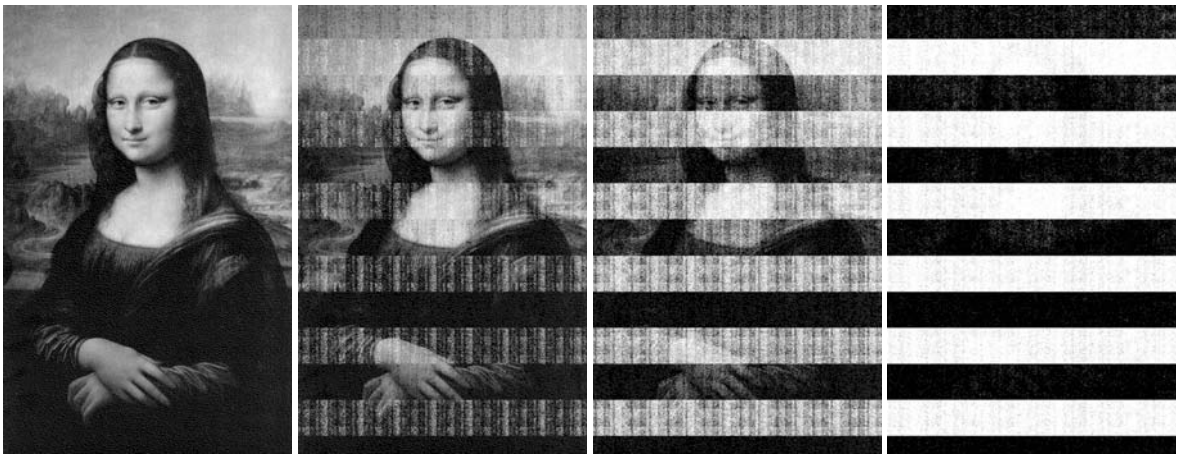


Figure 4: We loaded a bitmap image into memory on Machine A, then cut power for varying lengths of time. After 5 seconds (left), the image is indistinguishable from the original. It gradually becomes more degraded, as shown after 30 seconds, 60 seconds, and 5 minutes.

4 Imaging Residual Memory

Imaging residual memory contents requires no special equipment. When the system boots, the memory controller begins refreshing the DRAM, reading and rewriting each bit value. At this point, the values are fixed, decay halts, and programs running on the system can read any data present using normal memory-access instructions.

4.1 Imaging tools

One challenge is that booting the system will necessarily overwrite some portions of memory. Loading a full operating system would be very destructive. Our approach is to use tiny special-purpose programs that, when booted from either a warm or cold reset state, produce accurate dumps of memory contents to some external medium. These programs use only trivial amounts of RAM, and their memory offsets used can be adjusted to some extent to ensure that data structures of interest are unaffected.

Our memory-imaging tools make use of several different attack vectors to boot a system and extract the contents of its memory. For simplicity, each saves memory images to the medium from which it was booted.

PXE network boot Most modern PCs support network booting via Intel's Preboot Execution Environment (PXE) [25], which provides rudimentary startup and network services. We implemented a tiny (9 KB) standalone application that can be booted via PXE and whose only function is streaming the contents of system RAM via a UDP-based protocol. Since PXE provides a universal API for accessing the underlying network hardware, the same binary image will work unmodified on any PC system with PXE support. In a typical attack setup, a laptop connected to the target machine via an Ethernet crossover cable runs DHCP and TFTP servers as well as a simple client application for receiving the memory data. We have extracted memory images at rates up to 300 Mb/s (around 30 seconds for a 1 GB RAM) with gigabit Ethernet cards.

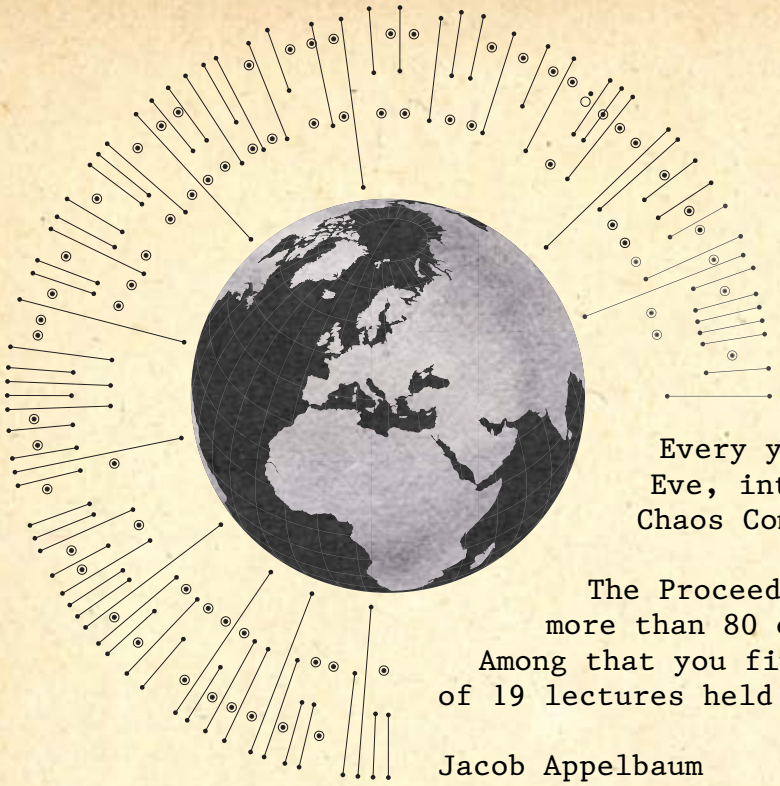
USB drives Alternatively, most PCs can boot from an external USB device such as a USB hard drive or flash device. We implemented a small (10 KB) plug-in for the SYSLINUX bootloader [3] that can be booted from an external USB device or a regular hard disk. It saves the contents of system RAM into a designated data partition on this device. We succeeded in dumping 1 GB of RAM to a flash drive in approximately 4 minutes.

EFI boot Some recent computers, including all Intel-based Macintosh computers, implement the Extensible Firmware Interface (EFI) instead of a PC BIOS. We have also implemented a memory dumper as an EFI netboot application. We have achieved memory extraction speeds up to 136 Mb/s, and we expect it will be possible to increase this throughput with further optimizations.

iPods We have installed memory imaging tools on an Apple iPod so that it can be used to covertly capture memory dumps without impacting its functionality as a music player. This provides a plausible way to conceal the attack in the wild.

4.2 Imaging attacks

An attacker could use imaging tools like ours in a number of ways, depending on his level of access to the system and the countermeasures employed by hardware and software.



Every year between Christmas and New Years Eve, intergalactic Hacker-society meets up at Chaos Communication Congress.

The Proceedings contain a full description of more than 80 events of this years congress. Among that you find academic papers by the speakers of 19 lectures held on 25C3, here a selection:

Jacob Appelbaum
Advanced memory forensics: The Cold Boot Attacks
Recovering keys and other secrets after power off

wesen
Algorithmic Music in a Box
Doing music with microcontrollers

Tor E. Bjørstad
An introduction to new stream cipher designs
Turning data into line noise and back

Rahmstorf
Climate Change
State of the Science

maha/Martin Haase
Neusprech im Schnüffelstaat
Politikersprache zwischen Orwell und Online

Bernhard Fischer
OnionCat - A TOR-based Anonymous VPN
Building an anonymous Internet within the Internet

Jan Torben
Privacy in the social semantic web
Social networks based on XMPP

Sandro Gaycken
The Trust Situation
Why the idea of data protection
slowly turns out to be defective

ISBN 978-3-934636-06-4 02300
9 783934 636064