An in-depth analysis of the cold boot attack

Can it be used for sound forensic memory acquisition?

R. Carbone
Certified Hacking Forensic Investigator (EC-Council)
Defence R&D Canada - Valcartier

C. Bean
Certified Hacking Forensic Investigator (EC-Council)

M. Salois
Defence R&D Canada - Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2010-296
January 2011

Principal Author

Original signed by Richard Carbone

Richard Carbone

Programmer/Analyst

Approved by

Original signed by Guy Turcotte

Guy Turcotte

Head/System of Systems Section

Approved for release by

Original signed by Christian Carrier

Christian Carrier

Chief Scientist

Abstract

The purpose of this technical memorandum is to examine the technical characteristics behind the cold boot attack technique and to understand when and how this technique should be applied to the field of computer forensic investigations. Upon thorough examination of the technique, the authors highlight its advantages, drawbacks, applicability and appropriateness for use in the acquisition of computer memory contents. The original cold boot attack paper, as conducted by a team of students and researchers in 2008, demonstrated the usefulness of computer memory remanence and how this phenomenon could be used to defeat popular disk encryptions tools and other data hiding techniques necessary for the safe storage of secret data and information. However, the technique is not a panacea and has many drawbacks dictated by the laws of physics, which cannot be overcome by the technique. The authors believe that a thorough understanding of this phenomenon will empower computer forensic investigators to take advantage of it when appropriate but also aim at dispelling various distortions surrounding it.

Résumé

Le but de cet article est d'examiner les caractéristiques techniques de l'attaque par démarrage à froid et de comprendre quand et comment cette technique devrait être appliquée au domaine des enquêtes en informatique judiciaire. À la suite de l'examen approfondi de la technique, les auteurs font ressortir ses avantages et inconvénients, son applicabilité et sa pertinence pour l'acquisition du contenu de la mémoire d'ordinateur. L'article original sur l'attaque par démarrage à froid, telle que menée par une équipe d'étudiants et de chercheurs en 2008, a démontré l'utilité de la rémanence de la mémoire d'ordinateur et comment ce phénomène peut être utilisé pour percer les outils populaires de chiffrement de disque et autres techniques de dissimulation de données nécessaires au stockage sécurisé de données et d'information secrètes. Cependant, cette technique ne constitue pas une panacée et comporte plusieurs inconvénients découlant des lois de la physique que la technique ne peut contrecarrer. Les auteurs croient qu'une compréhension approfondie de ce phénomène habilitera les enquêteurs en informatique judiciaire à en tirer profit lorsque requis mais dissipera également certaines des idées fausses qui l'entourent.

This page intentionally left blank.

Executive summary

An in-depth analysis of the cold boot attack: Can it be used for sound forensic memory acquisition?

Carbone, R; Bean, C., and Martin S.; DRDC Valcartier TM 2010-296; Defence R&D Canada – Valcartier; January 2011.

In 2008, a team of students and researchers from Princeton University, Wind River Systems and the Electronic Frontier Foundation put forward a seminal research paper examining the phenomena of computer memory remanence. Although that paper has confirmed what had long been theorized by computer security practitioners, no one had to date publicly examined this phenomenon. Their research has shown that the volatility of computer memory, which is taken for granted by many individuals throughout the world, can in fact be used against them. Though this phenomenon constitutes a useful resource for computer forensic practitioners and investigators, it is still not exploited in any systematic manner. Computer memory is an important and valuable asset that should be examined for possible evidence, which can then be used to incriminate or exonerate an individual. As the Princeton team has not specifically examined the applicability of their technique to computer forensics, the authors have endeavoured to do so, along with the evaluation of its appropriateness.

Although computer memory forensics is still very much in its infancy, forensic investigators and practitioners currently have several techniques at their disposal to acquire a computer memory's contents for evidential and analytical purposes. There has been much work and research to date carried out by various enterprising individuals and corporate entities to obtain the contents of memory through software-based means. The authors have studied these software tools and techniques for many months and have concluded that there are instances where software-based memory acquisition is not up to the challenge. In such cases, the cold boot attack may be used. However, the use of this attack is risky and fraught with many potential problems. Thus, it is best left as a solution of last resort. Though promising and useful, the technique is not without substantial risk to the integrity of the contents of computer memory. Even upon adequately carrying out the proposed requirements, there is no guarantee that the contents of the system's memory will remain intact, if even for very short periods of time.

This technical memorandum has been written in the interest of the computer forensics community. It examines the advantages and drawbacks to implementing this memory acquisition technique and provides ample technical information and important contextual knowledge.

This work was carried out over a period of four months as part of the Live Computer Forensics project, an agreement between DRDC Valcartier and the RCMP (SRE-09-015, 31XF20). The results of this project will be of great interest to the Canadian Forces Network Operation (CFNOC) in their mission of securing DND networks and investigating computer incidents.

Sommaire

An in-depth analysis of the cold boot attack: Can it be used for sound forensic memory acquisition?

Carbone, R; Bean, C., and Martin S.; DRDC Valcartier TM 2010-296; R et D pour la défense Canada – Valcartier; janvier 2011.

En 2008, une équipe d'étudiants et de chercheurs de l'université Princeton, de Wind River Systems et de l'Electronic Frontier Foundation ont publié un article phare qui examinait le phénomène de rémanence de la mémoire d'ordinateur. Même si cet article confirmait ce que les praticiens en sécurité informatique avaient théorisé depuis longtemps, personne à ce jour n'avait examiné publiquement ce phénomène. Leur recherche a démontré que la volatilité de la mémoire d'ordinateur, tenue pour acquise par plusieurs individus de par le monde, peut être utilisée contre eux. Bien que ce phénomène constitue une ressource utile pour les praticiens et enquêteurs en informatique judiciaire, il n'est pas encore exploité de façon systématique. La mémoire d'ordinateur est un atout important et précieux qui devrait, lorsque possible, être examiné pour la présence de preuves qui pourraient servir à incriminer ou exonérer un individu. Comme l'équipe de Princeton n'a pas spécifiquement examiné l'applicabilité de leur technique à l'informatique judiciaire, les auteurs de la présente ont entrepris de le faire, en plus de l'évaluation de sa pertinence.

Bien que l'examen de la mémoire d'ordinateur à des fins judiciaires en soit encore à ses balbutiements, les praticiens et enquêteurs en informatique judiciaire disposent présentement de plusieurs techniques pour acquérir le contenu de la mémoire d'un ordinateur pour fins d'analyse et de preuve. On compte de nombreux travaux et recherches menés par des individus et firmes entreprenants dans le but d'acquérir le contenu de la mémoire d'un ordinateur par des moyens logiciels. Les auteurs, qui ont étudié ces outils et techniques pendant plusieurs mois, ont conclu qu'il se présente des cas où l'acquisition de la mémoire par des moyens logiciels n'est pas efficace et proposent à la place d'utiliser la technique de l'attaque par démarrage à froid. Ils ont examiné de près diverses méthodes pour appliquer la technique et ont développé une pratique exemplaire. Bien que prometteuse et utile, la technique n'est pas sans risque important pour l'intégrité du contenu de la mémoire d'ordinateur et même en suivant à la lettre les spécifications, il n'y a pas de garantie que le contenu de la mémoire du système demeurera intact, même pour de très courtes périodes de temps.

Ce mémorandum technique a été rédigé à l'intention de la communauté de l'informatique judiciaire. Il examine les avantages et inconvénients de la mise en application de cette technique d'acquisition de mémoire et offre une information technique abondante et une importante connaissance contextuelle

Ce travail a été accompli sur une période de quatre mois dans le cadre du projet « Live Computer Forensics », une entente entre RDDC Valcartier et la GRC (SRE-09-015, 31XF20). Les résultats de ce projet seront d'un grand intérêt pour le Centre d'opérations des réseaux des Forces canadiennes (CORFC) dans leur mission de protection des réseaux du MDN et d'investigation des incidents informatiques.

Table of contents

List of tables

Acknowledgements

The authors would like to thank Mr. Yves van Chestein for reviewing this document and providing valuable feedback and comments. In addition, the authors would also thank Mr. David Ouellet for providing his comments concerning his own endeavours with the cold boot attack technique.

Target audience

This technical memorandum has been written specifically for those involved in or practicing computer or digital forensics. Therefore, it is assumed that the reader is familiar with the field of computer forensics.

Disclaimer

The cold boot attack can prove of significant use to qualified computer forensic investigators, both in government and in the commercial and private sectors. As such, the reader should neither construe nor interpret the work of the authors as condoning the use of the aforementioned techniques and capacities for illicit purposes.

Furthermore, the aforementioned authors of this technical memorandum absolve themselves in all ways conceivable with respect to how the reader may use, interpret, or construe this technical memorandum. The authors assume absolutely no liability or responsibility, implied or explicit.

Moreover, the onus is on the reader to be properly equipped and knowledgeable to use or work with liquid or gaseous refrigerants that are often volatile and highly dangerous to work with.

Finally, the authors, the Government of Canada, the Minister of National Defence (Canada), the Department of National Defence (Canada) and Defence Research & Development Canada are henceforth absolved of all wrongdoing, whether intentional, unintentional, construed or misunderstood on the part of the reader. If the reader does not agree to these terms then this technical memorandum should be readily returned to the Department of National Defence (Canada). Only if the reader agrees to these terms should he or she continue in reading it beyond this point. It is further assumed by all participants that if the reader has not read said Disclaimer upon reading this technical memorandum and has acted upon its contents then the reader assumes all responsibility for any repercussions that may result from the information and data contained herein.

This page intentionally left blank.

1 Introduction

The cold boot attack, as put forward by the Princeton research team [1], is a technique designed to take advantage of a long known yet little used technique for acquiring the contents of a computer system's memory. This technique consists in lowering the temperature of the memory chips far below ambient temperature in order to allow them to retain their content longer, whereupon the system is rebooted with specialized software that captures the content of those chips. The remainder of this introduction gives more detail and presents the rest of the work.

The objective of this technical memorandum is to examine the overall technologies employed in carrying out a cold boot attack [1], including the advantages, disadvantages and other necessary technical details required to successfully carry out such an attack for the purpose of acquiring the contents of computer memory. While the attack itself is rather simple to carry out, in contrast to many other forms of computer exploitation, the acquisition of computer memory is a volatile process that can readily result in a corrupted memory acquisition or worse yet, absolutely no data at all for post-mortem analysis.

This technical memorandum has been written for the computer forensic investigator who may have to perform such a memory acquisition at one time or another in the function of his or her duties. However, since little public information or literature is currently available to the computer forensic community other than the research put forward by the Princeton team [1], the authors have not only condensed this information for the reader but are adapting its consequences for use with computer forensics. Moreover, this technical memorandum brings together important information for the reader by presenting this material in a comprehensible format, including a technical discussion of how and why the cold boot attack does or does not work, as well as its advantages and drawbacks.

This technical memorandum is not, however, an examination of computer memory analysis. This specific vein of research is outside the scope of this research and warrants an altogether separate technical discussion to sufficiently examine the subject matter.

1.1 Usefulness of the technique

The cold boot attack is a powerful tool in the computer forensic investigator's bag of tools. Although it is far from being the suggested technique of choice, in extreme circumstances it may be the only viable option available and may make all the difference between succeeding in finding evidence and leaving empty handed.

Confusion continues to permeate regarding this technique with respect to how and why it works. Based upon numerous experiments conducted by the authors prior to writing, much independent data has been amassed in order to better understand the underlying phenomenon.

The cold boot attack technique does present some rather serious limitations, which may or may not be encountered at any given time when conducting an investigation against a given suspect system. This is because there are many variables to take into account in order to assess with certainty whether a cold boot attack-based memory acquisition will actually work or result in

complete memory data loss. While the Princeton researchers did provide some technical details on how their experimentation was carried out, they did not offer direct information detailing their success and failure rates while carrying out said research[1] [1]. Therefore, it will be difficult to compare the results obtained from the authors or that of future experiments and investigations conducted by computer forensic investigators since there is no direct baseline against which to compare.

The approach implemented by the authors while carrying out their cold boot attack experimentation is similar to what would be expected from a forensic investigator in the field attempting to carry out such an attack. The technical information and background provided herein, when coupled together with the experimentation carried out by the authors, should give the reader sufficient material with which to carry out his own cold boot attack -based memory acquisition.

1.2 What is the cold boot attack?

The cold boot attack [1] is sometimes known as a *platform reset attack*, *cold ghosting attack* or *iceman attack* [7], all of which refer to the same thing. However, this technical memorandum uses only the term cold boot attack.

The Princeton team [1] validated theories that had long been suspected but had never actually been confirmed in public sources [2, 3, 12, and 16]. They confirmed that simply turning off a computer does not necessarily ensure that all its memory contents are lost. Secondly, they have shown that the rate of decay for computer memory is dependent on two key variables, time and temperature, and that these are intricately conjoined.

The technique [1] examined by the Princeton team demonstrates the feasibility of recovering a computer system's memory after power removal due to the phenomenon of computer data memory remanence. Computer data memory remanence is a physical phenomenon caused by the use and storage of electrical charges used in computer memory chips. These memory chips store bits and bytes of computer memory as microscopically small electrical charges. Because computer memory is in fact a small stored charge, it takes time for these charges to sufficiently dissipate to return to their pre-charged ground state. At its ground state, all memory contents are effectively wiped out.

The Princeton researchers found that by flash-freezing[2] these memory chips, the electrical properties of computer memory could be altered to increase the time needed for these electric charges to degrade to their ground state. In so doing, it becomes possible in many instances to preserve a computer system's memory contents beyond the scale of microseconds before the electrical charges sufficiently dissipate or degrade. Flash-freezing, in this case, was carried out by inverting a can of dust removal spray, at which time the gas-based propellant, stored as a liquid inside the can, is propelled out. This liquid easily reaches temperatures of -40°C or colder, depending on the propellant used [18].

[1] Without this information, their results are called in to question.
[2] Flash-freezing is the process of suddenly bringing an object at ambient temperature down to a very cold temperature where some of its physical properties may change.

The objective of the Princeton team was to discover a method for defeating all-disk encryption schemes [1]. They postulated that many disk encryption tools store vital information in memory including cryptographic keys and passwords. However, since standard memory acquisition is not always feasible because access to the suspect computer system's console is not always possible, they sought an alternative method of acquiring a suspect system's memory.

Princeton's research team has shown that not only is it possible to acquire a computer system's memory using non-traditional means but that important cryptographic keys can be reliably retrieved from the flash-frozen memory. The research team provided source code for the various tools they used to carry out their research, several of which were used by the authors to independently verify and confirm the results.

In short, the memory is obtained in a linear fashion from the system's lowest accessible memory address to its highest. Memory is then saved in the form of a digital copy or image and if successful, constitutes an exact (or near exact) bit-copy of memory (assuming no memory degradation), similar to bit-copies of hard disk drives. Therefore, although not explicitly posited by the Princeton team, a cold boot attack-based memory acquisition is simply a means for imaging and acquiring a copy of a system's Random Access Memory.

The cold boot attack has many useful applications in the field of computer forensics. Although the Princeton researchers primarily developed the technique for recovering cryptographic keys from memory, it can conceivably be used to recover any memory content. This of course assumes that memory contents have not decayed or been somehow corrupted by the cold boot attack or post-attack memory acquisition software.

The cold boot attack can be used from an offensive or defensive posture, although the authors have taken an altogether neutral position. However, from the perspective of the forensic investigator, the overall outlook should be neutral, with emphasis placed on obtaining undamaged and untainted evidence that could potentially be later used in court.

1.3 Use of the cold boot attack software and source code

The Princeton team not only successfully demonstrated that memory temperature has a direct effect on the window of opportunity for memory acquisition but they also provided source code to the tools they used in order to carry out their exploits [1]. The authors tested many but not all of their source code throughout their own experiments.

The authors used the Princeton team's USB-based *bios_memimage*[3] software to create a low-level bootable USB flash drive that has the built-in ability to automatically acquire a cold boot attacked system's memory. The USB flash drive becomes bootable using the *dd* command. This software has been designed to compile under Linux and supports both 32 and 64-bit PC systems. USB-based acquired memory is written to the USB device and is then recovered using another compiled software tool found within the *bios_memimage* package named *usbdump*. Most modern PC systems support booting from a USB device.

[3] This software is available from http://citp.princeton.edu/memory/code as compilable source code.

In cases where the system does not support it, the Princeton team also devised a PXE-bootable scheme for acquiring the memory of a cold boot attacked system. This, however, requires the implementation of a configured and networked PXE server, entailing far more effort to implement than the USB-based memory acquisition device.

Of particular interest are the Princeton team's two other tools, *aeskeyfind* and *rsakeyfind*[3], designed to find and extract *Advanced Encryption Standard* and *Rivest, Shamir and Adleman* keys from acquired memory. Experiments conducted by the authors (see Section 4) have demonstrated that when memory images are intact or near intact, these encryption key detection tools do in fact work. The results of these two tools were then validated against the results obtained by an altogether different cryptographic key extraction tool known as *interrogate* (http://sourceforge.net/projects/interrogate), which succeeded in detecting and extracting exactly the same detected encryption keys as with *aeskeyfind* and *rsakeyfind*.

Unfortunately, older PCs used by the authors could not boot from USB-based devices. Therefore, a minimalist DOS-based CD-ROM with memory acquisition software (e.g. *Memdump* available from http://www.tssc.de/products/tools/memdump) was created. This DOS-bootable CD-ROM, based on a FAT32-compatible DOS (e.g. FreeDOS version 1.0 or MS-DOS 7.1) writes acquired memory to a FAT16 or FAT32 partition on one of the cold boot attacked system's disks for post-acquisition retrieval. Of course, this necessitates that these systems have access to a FAT-based filesystem with enough free space.

1.4 A note about maintaining the suspect system's state

Firstly, it is important for the computer forensic investigator to understand that in order to correctly implement the cold boot attack technique against a suspect computer system, extreme care must be taken to ensure the success of the endeavour. Specifically, the investigator must appreciate that in such an approach there are many variables to contend with. These include, but are not limited to, the current temperature of the memory vs. the temperature that it must be brought down to in order to sustainably maintain the charge of the memory's bits and the length of time the charge can be maintained for vs. the amount of time required to power off the system (post-cold boot attack) and reinitialize power.

Attempts by the investigator to verify some of these variables could potentially alter the state of the suspect system and its memory contents, depending on the methods used. For example, if an investigator attempts to quantify the suspect system's temperature or motherboard capacitance, either by introducing an instrument into the computer or by verifying BIOS[4] readouts, the system's state is very likely to change. Fortunately, in the case of temperature verification, thermal imaging solutions are available (e.g. thermal imaging solutions available from http://www.flir.com). Unfortunately, attempting to measure the residual capacitance of the motherboard is altogether very likely to cause drastic changes to the suspect system and its state. Many of these issues are more closely examined in later sections.

[4] Many PC BIOS chipsets fully support temperature readouts of processor, memory and other important system components. This information is usually accessible from a high-level system monitoring application from within an operating system.

Changes should only be made to the system's state when the investigator is actually ready to carry out the cold boot attack and as a last resort when all other forensics techniques have been tried and all the necessary results captured. Post flash-freeze, devices should only be connected to the suspect system after power has been completely removed or prior to the system's BIOS completing its POST routine. Other variables exist that can have an impact on the process although temperature and time are the most important.

Many of these variables are to some degree controllable by the investigator including, which supplemental devices are attached to the suspect system prior to a cold boot attack. The investigator can directly influence the temperature of the memory by flash-freezing it, although whether this is sufficient in all cases is likely to vary. Control can, to some extent, be exercised in terms of the time required to remove and re-establish the power to the suspect system, although the amount of time of it takes from the instant power is removed to powering back on the system will likely vary by system model and make. The investigator can also remove the flash-frozen memory chips and reinsert them into another controlled system of known behaviour. Furthermore, the investigator can also choose the method by which memory will be extracted (e.g. USB flash drive, DOS-bootable CD, PXE, etc.).

2 Technical background

In today's complex age of information technology, the necessity now exists where corporations, law-enforcement and various government agencies must proactively survey members of the population. Due to society's continually increasing reliance on information technology, acts of criminal or culpable behaviour are often carried out electronically and evidence resides within the electronic devices used to carry out these acts. However, this evidence is subject to varying levels of volatility. Investigative procedures against a suspect digital device may well recover evidence of wrongdoing. Unfortunately, not all evidence is well preserved. Evidence contained within durable digital media such as hard disk drives, CDs, DVDs, etc. is generally easily recoverable after the fact. Other forms of evidence such as the state of a suspicious network connection or a system's memory contents are not generally well preserved beyond the instant of investigative acquisition.

Computer memory is easily erased or altered the instant power is removed and for this reason, it is important to leave the crime scene intact until direct action to acquire a given suspect system's memory is required. With the current practice of shutting down the device and bringing it to a controlled lab, important evidence is often lost from the device's most volatile area, memory. In some cases, the memory's contents is not required at all and in other cases it proves too difficult to acquire, either because physical access to the memory is hindered by the computer system in some way or the architectural differences are too complex or cumbersome to overcome. In cases where memory is acquirable but acquisition cannot be carried out due to restricted access to the device's console or restricted access permissions, the cold boot memory attack can be adapted for forensic purposes.

In fact, if the procedure is well executed and few complications arise, the acquired memory may be forensically sound enough to not only provide useful indications of a suspect's actions but may even hold up in court. However, the integrity of the acquired memory will inevitably vary by situation and computer system. Nevertheless, this powerful and new computer forensic can be exploited by law enforcement.

A computer memory snapshot will capture the system state at the time of acquisition, including but not limited to running processes, threads, network connections, open files, locks, pipes and sockets, accessible devices, etc. Moreover, data processed from various applications may also be readily extractable from acquired memory images, such as current Internet connections, current chat sessions, opened emails, and so on.

2.1 Legal aspects

Crime has always existed and likely always will. The manner in which crime is committed, however, has changed with time. Over the last few years, there has been an undeniable surge in computer-related crime. Traditionally, computer forensic investigators concentrated primarily on less volatile evidence, specifically data and evidence stored on digital storage media devices including hard disk drives and optical drives. Computer memory forensics is very much in its infancy and even today, among the vast array of available memory acquisition software, there is still much room for improvement [19]. The cold boot attack technique, when applied to computer

forensics, enables the computer forensic investigator to go beyond the current limitations of software and hardware memory acquisition, thereby enabling the acquisition of additional evidence that may be of vital importance to an ongoing investigation.

However, the use of computer memory as evidence in a court of law is not yet commonplace as problems continue to plague both its acquisition and its analysis. If a computer forensic investigator can demonstrate due diligence in its acquisition and can reasonably demonstrate its intactness, then it may nevertheless be admissible. Furthermore, computer memory could prove a vital tool by enabling law enforcement to better understand the actions of the suspect and tie them together with various strings of evidence found on the suspect system. This evidence can then be used in a court of law to either exonerate or condemn.

This technical memorandum is not, however, a legal guide or compendium for the computer forensic investigator. Instead, the investigator should already be familiar with the laws governing his province, territory or state and with the rules of acquisition and collection of evidence.

It is unlikely that the majority of incidents that a computer forensic investigator responds to will necessitate computer memory acquisition. Often, he will face a *fait accompli* and the suspect system will have either been turned off or rebooted. The evidence will then have to be gleaned from available and acquirable digital media.

2.2 Why use the cold boot attack

When faced with a memory acquisition, the computer forensic investigator must understand that there are preferred methods to acquiring computer memory over the cold boot attack method. The primary method is software-based and requires that the computer forensic investigator have or gain administrative privileges to the suspect system and obtain access to the suspect system's console. Without these two preconditions, a direct software memory acquisition cannot take place [19]. Acquisition could occur over the network[5] but this requires that the network infrastructure already be in place[6]. The other method is hardware-based and can consist of either FireWire memory acquisition [22, 23, 24, 25, and 26] or PCI card [27, 28] memory acquisition. Unfortunately, many computer systems, even modern ones, do not support FireWire and many that do may have it disabled. PCI memory acquisition requires the insertion of a PCI card into the suspect system. When inserted into a running computer system, the PCI card may short it out, damaging it and the evidence. Moreover, this method is altogether a prototype and is not particularly ready for field use.

However, when the preferred method of software acquisition cannot be carried out, the forensic investigator will have to decide whether to approach the matter using FireWire (if available), PCI or the cold boot attack. All three options being available, that is to say that FireWire is enabled on the suspect computer, that a custom designed PCI card is unlikely to short out the system and that the system is a likely candidate for the cold boot attack, then the choice may be difficult to make. However, in just such a situation, the authors would opt for the cold boot attack since they have the ability to control some of the variables to its success.

[5] This is more common in corporate settings than at a suspect's domicile.
[6] Several popular commercial solutions currently exist to acquire computer memory over the network. Again, however, the infrastructure to do so must already be in place.

2.3 When to use the cold boot attack?

Different circumstances may necessitate the use of the cold boot attack although one case in particular stands out. The most likely scenario would revolve around an uncooperative suspect who simply refuses to unlock his system or provide administrative account credentials to the computer forensic investigator in order to gain access to the system (e.g. unlock the screensaver) to run memory acquisition software. As such, the main reason for using the cold boot attack-based memory acquisition technique is to image and acquire a suspect computer system's memory when it is not otherwise possible to do so using either software (memory acquisition software [19]) or hardware means (FireWire [22, 23, 24, 25, and 26] or PCI card[7] [27, 28]).

Of course, other situations are possible although unlikely to be as common. It is important to remember that under Windows 32 and 64-bit platforms, software memory acquisition can only be carried out using administrative accounts even if the computer forensic investigator has access to the suspect system's console. Hybrid Windows and DOS platforms, including Windows 95, 98, ME and DOS, do not require nor do they enforce the use of privileged users[8].

The problem of administrative privileges cannot simply be resolved by rebooting or powering off the system and rebooting using an ulterior method such as a recovery or Live CD, as all the suspect system's memory contents will be lost. Moreover, using hardware-based memory acquisition such as FireWire is inconsistent at best since not all computer systems support FireWire[9]. To further complicate matters, FireWire can be disabled either in BIOS or in the operating system's Device Manager and even when that is not the case, there is no guarantee that it would work. Of course, this avenue should be considered or explored prior to attempting the cold boot attack technique. However, the possibility exists that FireWire memory acquisition could crash the system. These matters could be further complicated if the computer forensic investigator does not have access to the suspect system's console and cannot therefore validate the level of support, if any, that is presently available from the suspect system. Compounding the matter is that no commercially supported options are available from vendors for implementing a FireWire-based memory acquisition. [22, 23, 24, 25, and 26]

Memory acquisition through the PCI card interface is a novel approach that has merit, but has no commercial support offerings yet. Capable computer engineers can design their own PCI memory acquisition cards from commodity components but this lies outside the scope of the average computer forensic investigator and outside the capability of most local/state law enforcement agencies. For this reason and despite its technical merit, the authors believe that this method is not currently a viable approach for conducting computer memory acquisition. [27, 28]

Therefore, when it is not possible to acquire memory through software- or hardware-based means, carrying out a cold boot attack is the only option left. The investigator should remember that the

[7] In fact, this should be the last approach to examine unless the computer forensic investigator possesses a customized PCI acquisition card.

[8] For dumping memory of a Windows 95, 98, ME or DOS-based system, consider using the *Memdump* tool.

[9] Laptops, especially Mac laptops, support FireWire. Modern desktops now support FireWire but older home/small office PCs did not commonly support it. Furthermore, even if the system and operating system support it, there is no guarantee that it will be able to directly access system memory and circumvent operating system-enforced access methods.

cold boot attack technique is also wrought with many potential problems that may leave the computer forensic investigator no better off. Understanding these pitfalls, as examined within this technical memorandum, should provide sufficient technical information to aid the computer forensic investigator to make an informed decision as to whether to proceed with a cold boot attack or not.

Even in instances where the suspect has been cooperative and the computer forensic investigator has privileged access to the system, it may not be possible to acquire the suspect system's memory. This is particularly true for 64-bit Windows platforms, where software memory-based acquisition support is particularly lacking. Memory acquisition support under Linux-based operating systems is also poorly supported. Other than very few specific live forensic CD applications (e.g. *Helix3 Pro* [19]) which are likely not to work the first time due to complex software library mismatches, the computer forensic investigator will be required to use the operating system's memory access software-based device. This software device, commonly seen under most Linux systems as either */dev/mem* or */dev/kmem*, is highly problematic to image memory from, particularly on modern Linux systems. These memory devices are just as likely to refuse memory access, as they are to enforce size limitations on forced memory dumps. Thus, in such cases as these, the cold boot attack may also be the only option. [19]

2.4 When to acquire memory and what to acquire

Imaging a suspect computer system's memory may or may not be a part of standard practice for the computer forensic investigator. Modern computer forensics has only begun looking at memory forensics and, although some advanced software acquisition tools exist, they are generally limited to 32-bit operating systems. Some tools support 64-bit Windows operating systems, but tend to be limited to specific versions and service packs of 64-bit Windows. Memory forensic analysis is another issue and should be currently considered in its infancy as few tools have any appreciable level of analytical capability against acquired memory images. [19]

Even if computer memory study is not yet a part of standard practice for the computer forensic investigator, then it should be imaged, when possible, in the following cases. Firstly, it should be imaged when there is suspicion of child exploitation as child pornographers have become adept at using encryption and steganography, each of which may leave important remnants in memory.

Obviously, use or suspected use of cryptographic or steganographic software warrants memory acquisition as passwords, cryptographic keys and data remnants may still reside in memory. It is to be expected that suspects with advanced computer skills may even use both types of software to cover their tracks. However, evidence may yet remain and may be obtainable and extractable both from acquired memory images. Moreover, some evidence may still reside on the suspect system's disks that would be acquired upon generating bit-copy disk images of the suspect system's disks and devices unless full disk encryption is used. All (or full) disk encryption schemes (e.g. *TrueCrypt* or other similar software) store their cryptographic keys in memory, which can be recovered through memory acquisition and analysis [1, 29, 30, 31, 32, 33, 34, and 35].

If swap exists, it is therefore important to capture it when imaging computer memory. Software-based memory acquisition tools, including HBGary's *Responder*, have the ability, at least for supported operating systems, to image both memory and swap at the same time, thereby ensuring a higher level of consistency between both spaces. [19]

In cases of suspected botnets or other malicious code including rootkits, viruses and worms, it is important to image a suspect system's memory as direct evidence of the malicious executable code can be found and extracted following a thorough analysis of the acquired computer memory. Many of the actions undertaken by the malware can often be successfully traced based on the system state information contained within an acquired memory image.

However, these reasons do not form an all-inclusive list of possible scenarios or events that should trigger the acquisition of computer memory. Other valid reasons include suspected abuse of the network (where it is used against an ISP or other individual or entity) or where a user has abused a corporate acceptable-use policy for the network. Similarly, other network-related activity including extortion, embezzlement, fraud, phishing, scamming, spamming and other illicit web activity may have left evidence in memory. In so long as the suspect has not rebooted or shut down the suspect computer there is always a fair likelihood that some evidence of such actions will remain in memory, especially network-related transactions.

2.5 The case for and against the cold boot attack

While the cold boot attack is a promising technique for acquiring hard to capture computer memory from a suspicious computer system, there are unique advantages and disadvantages in attempting to do so. The following three subsections will examine the pros and cons of this endeavour, along with other factors affecting the techniques.

2.5.1 Advantages

The primary advantage of a cold boot attack for memory acquisition is that computer memory can be captured with virtually no loss to data or integrity so long as it is done correctly and expeditiously. The exact procedure for carrying out the cold boot attack will vary according to system, platform and architecture. In general, time is very important and the physical memory modules must be properly handled and treated with the utmost care to prevent data loss. Equally important is the decision whether to conduct the cold boot attack directly on the suspect system or extract the chips and use them on an altogether different computer system.

Due to the potential for prolonging the life of computer memory, it may be possible to insert the flash-frozen memory into a compatible system that is under the control of the forensic investigator. This system should be of known configuration and behaviour and be ready to accept memory modules from compatible suspect systems. By using such a system, when the investigator is uncertain as to how the suspect system will behave upon removing and reapplying power due to potential start-up issues including boot password, boot device password, or BIOS memory checking, etc. Consequently, the possibility of using such a machine should not be thought of likely, as it may prove valuable in situations where there is reason to doubt the suspect system.

Nonetheless, the contents of flash-frozen memory modules can survive at ambient temperatures for a few minutes, so it is possible to quickly transfer the modules to a nearby machine of known and controlled behaviour. However, there is no guarantee that the removal of said flash-frozen memory will in remain intact upon transfer, even if done in a very short period.

Cryogenically[10] preserved memory may survive anywhere from many minutes to hours without an external power supply to refresh its electrical signals. This could enable a computer forensic investigator to bring said memory back to a suitably equipped computer forensics facility. There, in a controlled environment, the memory can be loaded into computer systems of known integrity with rigorously controlled behaviour.

In most cases, the suspect system can be used to image its own memory although this depends on the type of memory used by the suspect system (ECC or non-ECC), security configurations (e.g. boot password, boot device password, etc.) and BIOS configurations (e.g. fast POST cycle or a long memory-scrubbing POST cycle). The former BIOS configuration is preferred since this type of boot-up is the least likely to change memory contents. The latter, on the other hand is the most likely. As flash-frozen memory may remain intact for several minutes, this should provide a sufficient window of opportunity for the computer forensic investigator to either cryogenically preserve the flash-frozen memory or transfer it to a nearby controlled system.

The fact that not all computer memory decays at the same rate can be further exploited if it is suspected that a given computer system's motherboard may overtly suffer from motherboard residual capacitance. Although memory decay is largely based on time and temperature, the authors' own experiments have demonstrated that motherboard residual capacitance may also play a role. This can prove both advantageous and disadvantageous since not all motherboards have the same level of residual capacitance, even among differing versions of the same motherboard. This phenomenon can certainly help prevent memory from decaying too quickly but it may also adversely affect the investigator's ability to acquire a suspect system's memory. Unfortunately, at this time it is difficult to draw definitive generalizations, as many thousands of experiments against many, many different computer systems would need to be performed in order to adequately define such precepts. Moreover, it is uncertain at this point, what effect the suspect computer system's power supply plays in motherboard residual capacitance as all power supplies rely on capacitors to regulate the electric current.

Finally, a cold boot attack is acquired using the forensic methodology used for digital media (e.g. a bit-copy). Thus, if the memory has suffered little to no degradation, then it can be considered forensically sound. A simple test for soundness can be carried out by extracting text-like keywords[11] from the memory image and examining its output for signs of data degradation. For example, when text-like strings are displayed to the computer screen, look for strange characters where standard text characters should appear. Text-like data is highly likely to be distributed across the entire spectrum of a typical computer memory image, from the lowest to the highest

[10] It is important to understand the difference between the terms cryogenically cooled and flash-frozen as used in this technical research paper. Flash-frozen memory herein refers to memory which has been cooled to low temperatures (-40°C or lower) using an inverted can of dust spray. Cryogenically cooled means supercooled to temperature well below -150°C using super-cold refrigerants including nitrogen and other inert liquefied gases [20].

[11] Text-like strings can be extracted using the UNIX tool *strings* with 7-bit, 8-bit, 16 and 32-bit text encoding.

possible memory address [19]. Furthermore, further studies by the authors show that it is a common occurrence for text-like strings to take up to 10% or more of a typical computer memory image [19]. As such, should the memory prove to have little to no defects in its text-like strings then the authors are of the belief that the memory should be considered sound enough to hold up in a court of law.

2.5.2 Disadvantages

The primary disadvantage of a cold boot attack is that if it is incorrectly executed, the suspect computer system's memory will have either degraded or become irretrievable, as too many bits would have reverted to their ground state. As such, whatever data is recovered from memory would probably be inadmissible as evidence due to excessive degradation.

There are other disadvantages. Two important contributing factors that may prevent a successful cold boot attack are temperature and time, although in the authors' opinion temperature is the more important of the two. Memory chip temperature affects the amount of time a given memory module can retain its bits' electrical charges once power is removed. Even if memory remains at a constant temperature, the progression of time cannot be stopped. Decay will occur, albeit more slowly. Even cryogenically preserved computer memory will eventually decay as the individual capacitors of a given memory module's memory cells leak electrical energy [1, 2, 3, 12, and 16].

Unfortunately, it is not entirely known how long a given memory module will retain its electrical charges. Research carried out by the Princeton research team [1], Skorobogatov [2, 3] and Guttmann [12] indicate that the colder a given memory module is the longer it will retain its charges. However, while cryogenic preservation is likely to help preserve the shell life of the evidence, to what extent it can and for how is not yet adequately understood; thus no definitive answers are available at this time.

Standard computer memory, also commonly known as DRAM (Dynamic RAM), is highly volatile and very rapidly degrades at room temperature if left un-refreshed. Inside a running computer system, DRAM is refreshed multiple times per second to maintain its integrity and signal strength. Little publicly available literature on the specifics of computer memory degradation is available, which complicates the task of the computer forensic investigator. Compounding the matter is the fact that the same memory modules on different motherboards will experience different rates of decay. This is attributable to the residual capacitance of a given motherboard.

Specifically, since all computer motherboards require capacitors to function correctly, (e.g. regulate electrical signals, voltage peaks and dips, etc.) it is normal that each motherboard, when taken as whole, will have its own unique residual capacitance. This is because each capacitor soldered to the motherboard, typically ten or more, can exhibit slight variations even between the same make and model of capacitor. Furthermore, no two motherboards are identical, even among the same make and model as these devices are not built for absolute precision but instead to be within acceptable manufacturing tolerances. Taken together, these variations create the phenomena of motherboard capacitance, a little discussed and altogether unexamined issue among computer security professionals. These variations in residual capacitance can in fact be a key factor as to why the cold boot attack may or may not be successful when implemented against a given system. Although altogether impossible to gauge whether one system of the same make

and model has a higher or lower capacitance than its identical counterpart, it does explain why, when temperature and time are controlled (at least within margins of acceptable tolerances), one cold boot attack succeeds and the another fails against an identical system.

Another plaguing issue is that in order to acquire post cold boot attacked memory, a low memory footprint operating system is required. Even this type of operating system will still overwrite some memory, even though this can be controlled. The authors have found that such operating systems typically do not overwrite any pre-existing memory contents above the first megabyte of addressable memory. Unfortunately, not all PC systems are bootable from a USB device and therefore an alternative such as a DOS-bootable CD-ROM (or floppy) can be used so long as no extended or expanded memory drivers are loaded and no devices or programs are loaded into high memory.

Further complicating the implementation of the cold boot attack are safety concerns. Specifically, concerns with the use of refrigerants and cryogenic products. Refrigerants/propellants such as difluoroethane, trifluoroethane or tetrafluoroethane are commonly found in dust spray cans and expel a very cold liquid that freezes on contact with any room temperature article if sprayed upside down. The expelled propellant can have a range of several feet or more. A significant exposure to the propellant on human skin is very likely to result in severe frostbite and, if used recklessly, may even require hospitalization. These products are also volatile organic compounds and are a cause for health concern upon long-term occupational exposure. Moreover, these products are strong gases, so it is important to use these products in a well-ventilated area or they might displace enough air to dangerously lower the amount of oxygen in the room. In addition, although rarely seen, the refrigerant/propellant of some dust cans is ignitable. Specifically, difluoroethane and trifluoroethane are highly flammable and have an explosive potential [37, 38, and 39]. So long as the refrigerant/propellant is in its gaseous phase it is sensitive to static discharges, sparks and open flames but care must be exercised, although once in its solid phase (ice) its potential for ignition significantly diminishes. Finally, if the refrigerant/propellant ignites, its fumes are highly toxic to humans[12]. Thus, attempting to remove flash-frozen memory with the hands can not only result in frostbite but can inadvertently warm up parts of a given memory module thereby resulting in differing rates of decay between handled and unhandled memory modules.

Furthermore, cryogenic products such as nitrogen, while stable and inert compared to variations of fluoroethane gas, are very dangerous to work with and be around, especially if improperly handled as their use requires special gloves, suit and a facemask[13]. They must be transported in suitable containers such as vacuum flasks and are not available for use in handheld canister form. Only small amounts can be transported as liquid nitrogen has a propensity to rapidly evaporate and expand, sometimes violently so [40]. Moreover, when pouring a small amount onto an object such as a computer memory module, the liquid very quickly evaporates and in small confined spaces such as closed offices, can be asphyxiating. Good ventilation is once again required and the liquid should be emptied onto any object with suitable room to work as it can bubble and

[12] According to Air Liquide (France), both difluoroethane and trifluoroethane are extremely flammable and upon ignition readily release carbonyl fluoride, carbon monoxide and hydrogen fluoride, all of which are highly toxic to humans. Only tetrafluoroethane is inert and does not readily react or ignite under normal atmospheric conditions. [37, 38, and 39]

[13] This is to avoid splashing on the faces and eyes.

splash and cause extreme frostbite and cold burns. It can also render objects extremely brittle. Extreme caution must be used in applying it to memory modules and in their removal and reapplication elsewhere. Finally, because of liquid nitrogen's extremely cold temperature, it can cause gaseous oxygen to condensate, which can react explosively.

Removal of memory from a suspect system and reinsertion into a controlled target system can pose specific challenges. The target system must be known to accept the suspect memory modules and must not scrub or otherwise modify memory during its POST cycle. This is even more challenging when acquiring high speed memory modules and using them in systems which may not be compatible with that speed as there is no guarantee that the target system will be able to clock-down the inserted suspect memory. The reverse is true for very slow suspect memory. The overall architecture, size and shape of a given suspect memory module may be the same but the target system may simply not recognize it.

The use of computer memory riser boards complicates the matter of flash-freezing since there is far more area that must be covered by the refrigerant, thereby increasing the possibility of a mishap. Moreover, when swapping cold boot attacked memory it is important to determine that the controlled target system is both compatible with the suspect system's memory and supports the same memory data density.

It is also important to understand that working in humid or damp environments can very quickly lead to condensation of air moisture onto very cold objects such as flash-frozen computer memory. This increases the risk of potential shock when handling such objects and the risk of an electrical short circuit. Working in an environment of low humidity may help reduce the level of condensation but increases the risk of electrical discharge to the memory, which could also damage the sensitive memory modules.

It is also difficult to ensure that the suspect memory, if reinserted into a different target system, will not attempt to scrub the memory clean. Generally, this is only an issue if the memory is ECC-based. However, there is no guarantee that using ECC memory in a non-ECC capable system, or the reverse, will work. Instead, it is best to insert flash-frozen non-ECC memory into another non-ECC target system and flash-frozen ECC memory into an ECC target system that has had its ECC memory scrubbing capability disabled.

Finally, not all computer systems and memory modules are as susceptible to the cold boot memory attack. The last significant factor that can affect the success of a cold boot attack is the size and charge of the individual memory cell capacitors in a given memory module. As memory density increases, the size of the individual capacitors generally decreases, thereby affecting the residual charge remaining in each capacitor the instant power is removed from the suspect system. Even liberal application of refrigerant for flash-freezing may not be sufficient to prevent the memory cell capacitors from rapidly degrading to their ground state. This specific issue is more closely examined in Section 2.6.

2.6 Variations and other factors affecting the efficacy and ability to carry out a cold boot attack

Not all computer systems are equally susceptible to the cold boot attack. The Princeton research team concentrated on PCs but other computer systems are at risk as well. Many of these other systems automatically scrub their memory upon system initialization. Such computers, based on the authors' experience, include most high-end RISC-based workstations and servers, including those from SGI, Sun and IBM, as well as ECC memory-enabled PCs[14]. Producing an exhaustive list of computer systems that re-initialize system memory upon bootstrapping is outside the scope of this technical memorandum. Furthermore, it is important to recall that the overwhelming majority of PCs are based on CISC processors, with the only notable exception being Itanium-powered computers. Although Mac-based PowerPC systems are not examined herein, their memory is also generally compatible with PC-based memory and therefore should be susceptible to the cold boot attack.

While most non-ECC capable PCs are incapable of re-initializing their memory at bootstrap, some can and do. Unfortunately, it is often not obvious to determine upon first glance which computer systems will scrub their memory and which will not. Furthermore, the investigator cannot simply shut down the system and look at the system's BIOS settings. Doing so will corrupt or wipe memory contents. Moreover, there is no definitive list of which PC BIOS' currently scrub their memory. It is likely though that most PC servers with copious amounts of memory are likely to do so. This information may be available from the PC manufacturer's web site. Therefore, before powering off any computer system, it is important to ascertain, where possible, if memory scrubbing will occur. If in doubt, cool the memory, remove it, cryogenically preserve it if possible, and load it into a machine of known configuration and behaviour as quickly as possible.

A similar problem exists for RISC-based computer systems. Many RISC-enabled workstations and servers will destructively check system memory at bootstrap, even those that do not make use of ECC memory. However, so long as root-like system access can be gained to the operating system and assuming that the system is UNIX-based, the computer forensic investigator stands a good chance of successfully carrying out a live memory acquisition and will therefore not require the use of the cold boot memory attack technique. If the system, however, runs a trusted UNIX-like operating system including Trusted Solaris, Trusted IRIX, Trusted AIX, etc., then it is very unlikely that any direct access to system memory will be possible and the computer forensic investigator will have to attempt a cold boot attack. Some RISC-based systems enable the operator to perform a memory dump from its BIOS (or PROM, EEPROM, NVRAM, firmware, etc.) before any destructive memory checking occurs, although these systems are rare.

When considering the acquisition of RISC memory, ECC or not, it is important to consider, prior to carrying out a cold boot attack against the system and its memory, whether it will be supported by a given target PC. Although this may sound strange, in many instances RISC computer memory is entirely PC compatible. Therefore, it is likely that a cold boot attack will work against a variety of RISC computer systems. This is because many RISC workstation and server vendors

[14] ECC-enabled PCs are generally high-end workstations and servers. However, not all PCs that support ECC memory necessarily ship with it. Generally, ECC memory will reinitialize at system bootstrap unless otherwise specified in the system BIOS (but not all BIOS support this option).

and manufacturers comply with the various JEDEC specifications thereby ensuring compatibility with industry norms. Thus, many of these systems are compatible with their PC counterparts because their memory modules tend to use the same bus interface, voltages and clock timings[15] as PC memory. Other systems are, however, altogether incompatible with PCs. For example, DEC Alpha SGI MIPS/RISC workstations and servers are by design incompatible with PCs.

Another point to consider is the relationship between memory temperature and memory density. Standard computer memory is DRAM-based and relies on microscopic electronic cells that are in turn based on tiny capacitors that store a single very small electric charge denoting a single bit of memory. The ability for a capacitor to store a charge is proportional to its surface area. Therefore, the smaller the capacitor and hence the greater its memory density, the less charge each memory cell can hold. Thus, the effect temperature has on the dissipation of a capacitor's charge is a direct one although not necessarily proportional. In effect, the relationship can be seen this way: the colder the memory chips, the slower the discharge (or leak) of the various electrical charges and therefore the longer, the time before a given capacitor sufficiently discharges to its ground state.

2.7 Understanding computer memory fundamentals and its effects on the cold boot attack

It is very important to bear in mind that in general, the higher the memory density of a given memory module, the less it will be affected by cold temperatures. The investigator must therefore be mindful of such things. Although some research has already been conducted on the effect temperature has on computer memory, both dynamic and static [1, 2, 3, 12, and 16], the findings are far from conclusive. Studies [1, 2, 3, 12, and 16] do not provide sufficient information to draw any definitive conclusions concerning the correlation between temperature reduction and the effect it has on memory decay with respect to memory density.

2.7.1 Computer memory fundamentals for the forensic investigator

It is important for the computer forensic investigator be fluent in the low-level aspects of computer memory technology. While the investigator has significant control over temperature and time and over using a controlled acquisition PC of known behaviour, he has no control over computer memory densities, memory module cells and the decay of their charges. Understanding these factors is crucial in successfully carrying out a cold boot attack in order to acquire a forensically sound image of a suspect computer system's memory.

Much of the currently available public literature and documentation on the cold boot attack inexplicably avoids any discussion on computer memory fundamentals. Therefore, a brief examination of the topic is necessary in order to better understand how computer memory works with respect to type, memory density and temperature.

Computer memory, also known as Random Access Memory or RAM, consists of millions of cells (or more) that store individual electrical charges using either very tiny capacitors or transistor

[15] Sun memory respects the various JEDEC standards put forward by the organization for memory standards and interoperability [21].

gates. The method in which a given charge is stored depends on the type of computer memory used. Modern computer systems use various types of memory differently and each type is used for distinctive purposes. All forms of RAM are volatile in nature, making them a non-permanent means of storing transient electronically processed data. As a direct consequence of RAM's impermanence, RAM-based computer memory data-remanence is relatively short. Once the main power to a computer system is removed, it is only a short matter of time before all traces of electronically stored data in RAM are irretrievably lost.

The actual amount of time required before data loss occurs varies greatly according to the type of computer memory and can be anywhere between several seconds to many minutes. Even memory that requires many minutes to fully degrade begins degrading shortly after power loss, but important remnants may yet be recoverable. The exact interval of time between integral RAM to any specific level of decay depends on too many factors to allow for a meaningful scientific analysis herein. Nevertheless, understanding these various factors will better enable the computer forensic investigator to make assessments that are far more accurate.

It is therefore important to briefly review the most widely found forms of computer memory in use. The two most common are SRAM and DRAM.

SRAM is generally restricted to hardware components where very high data transfer rates are required such as CPU cache and hardware buffers (e.g. hard disk and optical drive buffers). Typical incarnations of DRAM include older computer memory modules no longer in use but once commonplace among computer systems including the 386, 486 and older Pentium models. DRAM, now obsolete, has been superseded by SDRAM that includes DDR1, DDR2 and DDR3 memory technologies, each of which is a newer form of SDRAM. Physical differences between DRAM and SDRAM become apparent upon close visual inspection although such an inspection may not always be possible for various reasons. There is also a significant improvement in data transfer rates with SDRAM.

DRAM-based memory modules can be distinguished in large part by their bus interface, which consists of a SIMM or DIMM interface. The SIMM interface utilizes a single side of metal contacts on a memory module to connect to the system bus. DIMM-based memory uses two sets of contacts, one on each side of the module. Different types of DRAM, whether they are SIMM or DIMM-based, utilize a specific number of contacts to connect to the system bus. SIMMs are generally limited to 72 or less single-sided memory module contacts (sometimes called pins) while DIMMs are 72 or more dual-sided memory module contacts.

Although SIMM-based memory is still to an extent available today, it is used primarily for upgrading older computer systems and it has not generally been used as main memory in computer systems for more than 10 years. It is quite uncommon in running computer systems today, except for the occasional old computer system that must still be used for running archaic software or hardware. Thus, as time goes by, it is all the more unlikely that a computer forensic investigator will encounter one of these older systems.

Today's modern computers rely almost exclusively on DDR-based DIMMs, including many RISC-based computer systems. Older and proprietary computing architectures including SBUS (Sun), QBUS (DEC VAX, Alpha), HP PA-RISC (HP), IRIX (SGI), etc., have used their own specific memory interfaces and technologies that were either manufactured to specific design

requirements by the computer vendor itself or by a third-party memory module manufacturers. These proprietary bus and memory technologies, although primarily RISC-based, also counted certain CISC-based systems among their ranks. Moreover, these systems have largely been superseded by more recent built upon PCI-based technology. The computer forensic investigator is not likely to encounter one of these older systems although, owing to their high acquisition cost, some may still be in operation today. Carrying out a cold boot attack against one of these systems is a highly complex and risky undertaking that will likely lead to memory corruption.

Certain generic principles can be applied to computer memory. In general, the more contacts a given memory module has, the faster its bus transfer rate and the greater its potential memory density. For example, modern high-density DIMM-based memory modules can easily support 1 to 4 GB per DIMM. These DIMMs typically require 168-pin contacts or greater. Therefore, it is important that the computer forensic investigator attempt to determine the memory type (SIMM or DIMM) and the number of contacts, when possible[16]. It is not always possible to visually distinguish between various DDR1-DDR3 memories since there is no consistency between a given type of DDR memory and the number of contacts it may have. All too often, all SDRAM-based computer memory looks alike. This lack of consistency is likely to be particularly troublesome and may cause an inaccurate assessment of a suspect computer, thereby potentially jeopardizing the integrity of cold boot-acquired memory, as the appropriate procedure may not have been observed. The precautions taken should be commensurate with the type of memory in question, as not all memory requires the same safeguards. For example, very high-density memory should be cryogenically cooled rather than flash-frozen since the memory modules' individual memory cell-based electrical charges are extremely small and rapidly degrade as compared to lower density memory.

It is common for modern computer systems to be sold with memory varying from 1 to 4 GB RAM which is likely spread across multiple memory interfaces (commonly known as memory slots). When considering the amount of RAM a suspect system can support, it is important to ascertain that system's underlying architecture. Modern commodity computers are available in 32 and 64-bit architectures, although most recent systems today support both 32 and 64-bit processing. The 64-bit models are capable of supporting much larger amounts of memory. In fact, 32-bit systems are generally limited to a maximum of 4 GB of memory, unless the system supports advanced memory addressing features such as Physical Address Extension (PAE), but the operating system must also support this feature [17, 19]. Most modern Linux distributions support both 32 and 64-bit computing. Recent Linux 32-bit specific distributions support PAE. 32-bit versions of the Windows operating system support differing amounts of memory [17, 19].

Therefore, the computer forensic investigator must determine if a suspect computer is 32-bit or 64-bit in nature, information which would not only affect the potential amount of memory the system could support but also the manner in which a cold boot attack occurs. The vast majority of 64-bit commodity PC systems, whether found in a corporate or home environment, typically house less than 8 GB RAM or less[17] are prohibitively expensive. Thus, the computer forensic

[16] Depending on the location of the memory within a given computer system, it may not always be possible to clearly observe and classify the memory.

[17] Most modern desktop-based commodity motherboards support between 4 and 6 memory slots; some support 8 or more, although these are uncommon. Thus, large amounts of RAM require memory modules of very high density, which become very expensive when memory modules go beyond 3 GB per module.

investigator will have to search for information concerning a suspect computer, including its architecture type (32 or 64-bit), supported memory options and supported peripherals, including IDE, SCSI, USB and FireWire. A system's support of USB-based devices is very important since most cold boot attacks use USB devices for dumping memory to, although PXE-based memory dumping is also possible. This is examined in more detail in the next section.

2.7.2 DRAM memory decay

Although DRAM is highly volatile and requires continuous refreshing via a sustainable computer-based power source, data remanence can still be observed. It poses serious challenges to the computer forensic investigator whose task it is to recover that data from memory. Simply removing the power from a computer system does not guarantee that the contents of a suspect system's RAM are immediately expunged. In fact, computer memory is capable of retaining much of its previous contents without power for seconds in DRAM/SDRAM [1, 4, 5, 6, 7, 8, 10, 12, and 36] and minutes in SRAM [2]. Exact specifics, however, varied greatly according to memory density, manufacturer and temperature. However, the results are highly indicative of memory decay occurring faster in DRAM/SDRAM than with SRAM and within seconds any contents present in DRAM/SDRAM would be completely data free as too much decay would have occurred by then, thereby thwarting any cold boot attack.

DRAM, in contrast to SRAM, generally uses one capacitor to store each bit within a circuit containing one or more bytes of data. DRAM is typically denser than SRAM and can hold far greater quantities of information per RAM chip as its capacitors are tiny in comparison to SRAM's transistor-based technology[18].

Since DRAM data densities varies by manufacturer it is likely that different memory modules from different companies will require varying units of RAM chips to be of create a similar memory module of equivalent size. Although DRAM is typically denser than SRAM, gains in its density are offset by the fact that DRAM has a higher electrical requirement than SRAM due to the need to continually refresh its bit-storing capacitors. Consequently, DRAM's I/O speed is typically far less than that of SRAM although its speed has been increasing[19]. DRAM's greater density is due to the overall small size of its capacitors and the fact that less circuitry is required for integration into a memory cell. Although on the surface SRAM may appear superior to DRAM, it is prohibitively expensive to equip a computer with sufficient SRAM for modern computing requirements given that its manufacturing costs are significantly higher than that of DRAM.

Electrical refreshes of DRAM occurs every few milliseconds[20]. All the memory modules are modified in tandem, although specific refresh rates or timings are particular to the needs of a

[18] Recall that RAM chips are the components that actually store electronic charges. They are dark or greyish in colour and appear as discrete rectangular ridges on a given memory module. Memory modules typically hold two or more RAM chips.
[19] Consider that newer memory technologies such as SDRAM PC16000 can run up to 2000 MHz (when overclocked) and very high-end modern motherboards can support memory I/O bandwidths of 40 GB/s or more [41].
[20] The JEDEC standard requires that DRAM be refreshed at least every 64 ms [9].

given set of memory modules. Generally, the computer's BIOS will take care of these refresh timings, but this can sometimes be modified by the operator (e.g. overclocking).

DRAM is a capacitance-based memory capable of storing electrical charges for only limited periods. Typically, the capacitance of each bit-based capacitor is in the range of picofarads[21] [13, 14, and 15]. The continuous refresh cycles of DRAM explain why it generally has greater electrical requirements than SRAM[22]. Specifically, DRAM relies on capacitors whose electric charges more readily dissipate while SRAM requires a steady source of electrical refreshment. With SRAM, however, no refresh is required as long as power is maintained to the system.

The ability of DRAM-based capacitors to hold a given electric charge is proportional to the surface area of the conducting and dielectric materials[23] used for holding said charge [11]. In order to increase the density of standard computer memory, manufacturers have had to shrink the size of their DRAM-based memory cells. Consequently, this has also decreased the size of the DRAM memory cell capacitors. As memory density increases, the surface area of each individual capacitor lessens thereby storing less charge per bit of information. This smaller stored charge has the added effect of diminishing the time required for a given capacitor to sufficiently discharge to return to its ground state. This discharge or leaking of electrical energy is the prime cause of DRAM-based memory decay. The surface area of a capacitor has a direct and proportional effect on the charge it can hold and the time required to dissipate back to its ground state.

However, several issues with DRAM decay require additional experimentation. Firstly, it is unknown what the half-life of computer memory is. Although it is possible to statistically model memory decay, it can only be done if certain estimations are made for key factors. Moreover, because the process of computer memory decay is random (a stochastic process) in nature it can only be modeled using a statistical multivariate covariance model. However, these types of analyses are outside the scope of this technical memorandum. While the Princeton team claims to have accurately modeled decayed computer memory they have not provided enough specifics on the matter [1]. The data recreated from partially recovered memory that has been restored according to a given memory decay model would likely not hold up in court. However, jurisprudence may one day provide guidance although to date no such known case has ever occurred in Canada or the United States.

The rate of DRAM-based memory decay is a function of several factors, the most important of which are the size of the capacitors storing each bit of data, the memory's temperature and the residual capacitance of the motherboard. Other factors, including impurities in the semiconductor material itself and other physical forces may play a role including cosmic rays, electromagnetic interference, etc. Though these factors fall outside the control of the investigator, he can to some extent mitigate, if even temporarily, the largest and most important factor, temperature, through use of external means (e.g. flash-freezing).

Thus, the effects of computer memory loss can often be significantly improved using an appropriate amount of cooling. Of course, this requires that the computer forensic investigator

[21] Pico is a metric (SI) unit of measurement equivalent to 10^{-12}.
[22] Notable exceptions exist, especially for high-speed SRAM.
[23] Capacitors are generally made of electrically conductive materials separated by some dielectric material.

have direct physical access to the memory, which may not always be the case. When it is available, the investigator can cool the memory using readily available means such as inverted cans of compressed air or other refrigerants that cool on contact. If moderate-term storage of memory is necessary for transportation back to a computer forensics laboratory then the memory can preserved using liquid nitrogen (or any other suitable non-electrically conductive refrigerant).

The supercooling of computer memory has the dramatic effect of lengthening its lifespan. However, all electrically charged computer memory will eventually sufficiently decay and revert to its ground state. Fortunately, the rate at which this occurs across memory modules of different sizes can significantly vary in time. DRAM memory modules that have a lower bit density generally will have a lower capacitor density thereby permitting the use of larger capacitors by the manufacturers. The larger the capacitor and its surface area the longer individual bit charges can be retained. Conversely, very high-density computer memory will have smaller capacitors in comparison to lower density memory and should therefore dissipate their charge faster. Of course, other factors such as impurities in the dielectric material may also play a significant role.

Of course, these are only generalizations and exceptions will exist. For example, there will be instances of low-density[24] memory with relatively small capacitors and conversely denser memory with larger capacitors. Furthermore, the inability to adequately flash-freeze memory when combined with additional secondary effects such as motherboard-based residual capacitance play a direct role in determining a suspect system's susceptibility to the cold boot attack.

However, it is important for the computer forensic investigator to understand the effects of temperature on computer memory. The cooling of computer memory affects the resistance[25] of the individual capacitors within a given memory module's memory. Under normal operating conditions, these memory cell-based capacitors function as expected. However, once the memory begins to cool below ambient temperatures the electrical resistance of the capacitors begins to increase. Thus, the larger a given capacitor's change in temperature the greater its electrical resistance becomes until its temperature has been sufficiently reduced for Fermi levels to directly affect an increase in electrical resistance. It is this increase in electrical resistance that is responsible for the phenomena commonly known as the cold boot attack. Should this physical phenomena not exist the cold boot attack would not be feasible. This phenomenon exists because capacitors are in fact semiconductors (with varying levels of dielectric doping) and are subject to Fermi levels as pursuant to the amount and type of dielectric materials used therein. [2, 3, 5, 11, 16, 42, 43, 44, 59, and 60]

Thus, the increase in capacitor-based resistance causes the individual memory cell capacitors to retain their electrical charge for longer periods once power is removed from the system. In the case where cooling is extreme (e.g. the use of cryogenic refrigeration) the electrical charges of the individual capacitors remain nearly constant and will require many hours before any noticeable depreciation of charge is observed. Noticeable changes in bit charges may be noticed by the investigator upon examining a memory image of flash-frozen memory when memory flips/flops

[24] Low-density DRAM-based computer memory should therefore have a stronger bit charge.
[25] Resistance is the opposite physical phenomena to conductivity. Many materials, particularly metals become better conductors as temperature decrease. However, capacitors are often made up of dielectric materials which often behave inversely to conductors. [11, 42, 43, 44, 45, 59, and 60]

are spotted. A memory flip/flop occurs when the charge of a memory capacitor significantly changes but has not yet reached a low enough energy level to be at or near its ground state. [2, 3, 11, 12, 16, 42, 43, 44, 59, and 60]

The Princeton research team has claimed that sufficiently flash-frozen memory can be left at room temperature for up to an hour before significant data loss occurs [1]. The authors have not tested this assertion; however, they do not subscribe to it either. Memory supercooled using liquid nitrogen can remain intact for even longer periods of time, on the order of hours to days, but to that end, it must remain appropriately cooled [1, 12, and 16].

2.7.3 SRAM memory

In contrast to DRAM, SRAM memory cells are typically much larger than that of DRAM due to the inclusion of additional circuitry required to store their charge. The speed of SRAM is due directly to its highly effective (and complex) circuitry while the density of DRAM is due to the small size and simplicity of its circuitry.

Because SRAM uses transistor-based technology (while DRAM is capacitor-based), the electrical charge of each memory cell within a given memory module remains operationally effective in so long as power is maintained to the computer system. Moreover, so long as power is available there is no need for SRAM to refresh. Thus, in comparison to DRAM, SRAM is static in nature. Even once power is removed from a suspect computer, it will take some time [2] for the electrically charged transistor-based memory cells to sufficiently dissipate in order for the bits to revert to their ground states. SRAM, unlike DRAM, is generally found in small amounts and is typically reserved for hardware-based cache or buffers. Although SRAM is very volatile with respect to its constantly changing contents due to its perpetual use by the processor the data remanence of its contents are far less volatile than that for DRAM that requires continuous electrical refreshing from a cyclically based power source (e.g. computer power supply). [8, 9]

Once power is removed, its contents will decay, although SRAM's rate of decay is far less than that of DRAM. It can retain its data integrity for minutes without power [2, 3]. If the SRAM is sufficiently cooled, it can retain its charge for much longer [2, 3]. Nevertheless, as with DRAM, the eventual decay of SRAM is inevitable. According to [2], some non-cooled SRAM memory modules were found to retain their data integrity for upwards of 15 minutes before significant data loss. Therefore, SRAM-based data remanence is a reality that could be exploited by knowledgeable computer forensic investigators and security practitioners and researchers [2, 6, and 7].

However, the vast majority of cold boot attacks will be DRAM-based, as little practical research has thus far been conducted against SRAM-specific memory acquisition. SRAM can typically be found wherever high-speed low-power memory is required such as cache memory for CPUs and hard disk drives and as buffer memory for printers, LCD displays and CD/DVD-based devices [9]. Forensically, certain types of SRAM are easier to recover, specifically those used as hardware buffers due to their typically larger size and built-in bus interface[26] [2]. An Internet

[26] Easily acquirable SRAM generally has a bus interface in the form of pins that connect to the I/O board of the device in question. These pins can be hooked up in an electronics laboratory where the memory chips can be read or written to by the operator/technician.

search shows different computer forensic laboratories, investigators and lawyers offering SRAM data recovery services. However, it is altogether uncertain if CPU-based SRAM recovery is readily feasible, even in well-equipped laboratories, because of its remarkably small size.

The purpose of SRAM, particularly CPU SRAM is to store pre-fetched data the CPU will likely require in the immediate short term. Unlike DRAM contents, which can sit there for long periods, SRAM is continually being filled by newer pre-fetched data, which further complicates its recovery. Moreover, SRAM can only store very limited amounts of data[27].

When a computer system is placed in hibernation or sleep mode rather than the operating system copy out any cryptographic keys to CPU cache (SRAM) the keys are instead written out to the hibernation file. However, in cases where the computer system both has an available TPM unit and the underlying encryption software supports said unit the keying material will continue to remain in the TPM unit rather than be written out the hibernation file. In cases where TPM units are not available or that the encryption software does not support such a unit moving all encryption keys to CPU cache would be provide a feasible countermeasure to hibernation-based key material recovery. Since the computer forensic investigator is generally interested in DRAM-based memory, SRAM-based cold boot attack-based memory acquisition is not a likely approach at this time due to the technological complexities involved. However, as technology changes, some developers may decide to take advantage CPU cache for storing cryptographic keys. Although TPM devices are typically seen as more secure, not every system is equipped with one, thus the use of CPU cache could be seen as a more secure route rather than merely leaving the keying in plain computer memory. [2, 47, 49, 50, 51, 52, and 53]

2.8 Other important technical details concerning the use of memory acquisition hardware and software

A cold boot attack against a suspect computer system should only be carried out in the event that no other methods of acquiring the system's memory are possible. Once the computer forensic investigator has decided that the cold boot attack is the best method possible for acquiring a suspect system's memory, there are several lesser issues left to examine before the investigator is ready to attempt cold boot attack memory acquisitions. These issues are related to the use and exploitation of memory acquisition-specific hardware and software as it relates to capturing memory from a cold boot attacked suspect system.

2.8.1 Hardware-specific issues

Beyond the aforementioned hardware-related issues concerning the implementation of the cold boot attack, it is important for the computer forensic investigator to decide what type of media will be used in initiating and acquiring a cold boot attack-based memory dump from a suspect computer system. The most common form of media likely to be used will be USB-based as the overwhelming majority of PCs have the ability to boot from USB-based devices. Of course, the specific capabilities of these various systems are limited by their BIOS configurations and

[27] SRAM size typically ranges between a few hundred kilobytes, for small data buffers, to a few megabytes for CPU and moderately more for hard disk drive caches (the largest current hard drive caches are currently 64 MB in size).

capacities. The use of USB-based media are likely applicable to more computer platforms than just PCs. Apple-based Mac desktops and workstations have the ability to boot from a variety of devices including USB storage devices, although the implementation of the cold boot attack against these systems has not been examined herein. However, most non-PC based platforms do not support USB-based booting including platforms such as SGI MIPS-based workstations and servers, IBM RISC workstations and servers, HP PA-RISC workstations and servers, and older Sun SPARC-based workstations and servers.

The most commonly used large-storage USB media available are USB flash drives that currently (circa 2009/2010) have capacities of 64 GB[28] and more, although at this size they become expensive. It is therefore suggested that the investigator have several USB flash drive sizes available for immediate use. The authors suggest that the investigator have at least one each of 4 GB, 8 GB, 16 and 32 GB flash drives. The Princeton team's *bios_memimage* software package will also work on USB-based hard drives so larger memory acquisitions are possible.

The vast majority of the experiments carried out herein used on the Princeton team's *bios_memimage* software package. In all but one experiment, it worked without issue. The sole unsuccessful experiment resulted in a partial memory image. The tool has 32 and 64-bit instances. The 32-bit instance was used against systems with 4 GB RAM or less and the 64-bit instance was used against systems with 4 GB or more of memory. The next subsection provides more details on *bios_memimage*, which works only with USB-based devices. Attempts to install the program's bootloader onto a standard hard disk drive with an IDE, SCSI, SATA (or e-SATA), or SAS interface will fail upon boot-up of the disk's installed *bios_memimage* bootloader.

The authors have successfully acquired cold boot attacked memory images of 14 GB in size using a 16 GB flash drive against a Dell Precision 690 workstation. Additional experiments have revealed that this same computer system, when equipped with 24 GB RAM, can be acquired from a USB-based hard disk drive[29], although if a 32 GB flash drive had been available, it is very likely that this too would also have succeeded. Additional experimentation is required to determine the software's practical upper limit. Based on the tool's source code, it appears that by default, in 64-bit mode, upwards of 64 GB RAM is supported although this can be changed.[30]

The investigator may encounter situations where the suspect system does not or is unlikely to provide support for USB-based booting. In such a case, the investigator has several choices. A floppy diskette with a small DOS[31] operating system can be used, or a bootable CD-ROM with a

[28] Capacities of 128 and 256 GB are currently available although their price is upwards of $500 US (or more) for the former and $1,000 (or more) for the latter.
[29] Installing the *bios_memimage* bootloader onto a USB-based connected hard disk drive should work in almost all cases.
[30] Upon reading the source code and documentation for the *bios_memimage* program, it appears that entry "*#define MAPS 63*" from file *x86-64/mapmem.c* can be modified to 64 or higher to support more than 64 GB RAM.
[31] A suitable Linux operating system will not fit onto a 1.44 MB floppy. However, if the suspect system supports LS-120, Zip, or JAZ drives, then said media types with an appropriately configured Linux disk could be used for booting, acquisition initiation and memory acquisition storage. These drives are all limited in size, however, thus it is possible that memory images may have to be stored elsewhere, possibly to an unused hard disk drive partition or a newly inserted hard disk drive via an available internal or external system interface.

suitable operating system to provide appropriate hardware and memory support or a hard disk drive introduced into the suspect system by the investigator. Having tried this, it is important to state that the cold-boot software will not work if booted from a CD or DVD.

Although the Princeton research team does provide PXE (network) booting in its *bios_memimage* program it was never used or tested by the authors. The use of PXE requires that the suspect system have this capability and almost all computer systems built over the last decade support PXE booting including non-PC architectures and platforms. However, to successfully work, it requires that a PXE boot server reside on the same network as the suspect system and that the TFTP service is open on said network. In the opinion of the authors, this method of acquisition is too demanding, difficult and time consuming to implement. Instead, the investigator should pursue other avenues for carrying out the cold boot attack and acquiring the suspect system's memory.

2.8.2 Software-specific issues

This subsection discusses issues related to the different software programs that can be used to carry out a cold boot attack.

2.8.2.1 Bios_memimage

The first software tool of interest is the *bios_memimage* program. This program compiles readily under Linux and BSD but compiling it on other platforms is likely to be a time-consuming endeavour. Under Linux, the program has little requirements to compile successfully. It can be compiled as either 32 or 64-bit using the program *make*. If the current environment is 32-bit, a corresponding set of tools will be compiled. Conversely, if the system is 64-bit, a corresponding set of tools will be built. The source code is freely available but, of course, it is best that these tools be compiled long before they are ever required and placed in the investigator’s toolkit.

The *bios_memimage* program has two key software components, the *scraper* and the *usbdump*.

The *scraper* tool is the actual bootloader that is installed onto a USB-based device to boot a suspect computer system. The memory is dumped to the same device on which the *scraper* bootloader resides. This is its default behaviour and changing it requires a significant rewriting of the tool's source code. Furthermore, it is highly advisable to have a duplicate set of flash drives, where one set contains the 32-bit *scraper* program and the other contains the 64-bit *scraper* implementation.

Once compiled, the *scraper* tool can be found in *bios_memimage/usb/scraper*. The bootloader must be copied to sector 0 of the USB device. It will not work if copied anywhere else as this program contains system bootloading initialization code that replaces the computer's standard partition table code. Although the tool is not a stand-alone operating system it knows enough to initialize itself, gain access to its own disk media and acquire memory. The 32-bit version does not provide PAE support and therefore can only acquire the first 3.2 GB RAM (approximately). Acquiring more memory requires using the 64-bit implementation, assuming the suspect system supports it. Fortunately, most modern systems support both 32 and 64-bit mode. The tool is very simple and runs without user interactions. Once a memory dump is complete the system is

rebooted (although in our view it should have been designed to automatically power off the system), therefore the operator must be attentive so that the suspect system does not boot up from any of its hard disk drives, thereby potentially contaminating non-volatile evidence.

The second software component is *usbdump*. Using the tool *bios_memimage/usbdump/usbdump*, the memory image stored on the USB device can be extracted for analysis on another system. Once the memory is extracted, the device can be used again although in the opinion of the authors it is best that the device be zero-fill wiped prior to reuse to ensure that there is no contamination of the device from previous memory dumps. The bootloader cannot store memory images serially. Thus, if the memory image is not extracted and the device is reused it will be overwritten with the ensuing acquired memory dump.

For most situations, the *bios_memimage* program will be sufficient to acquire and extract memory images from cold boot attacked suspect systems. However, in situations where it is not possible to boot from a USB device, another method must be used. A detailed examination of these methods is outside the scope of this research technical paper. However, a brief discussion on using DOS and Linux as alternative methods is provided in the two following subsections.

2.8.2.2 DOS-based issues

A DOS bootable USB solution may sometimes be appropriate in circumstances where the *bios_memimage* tool does not work correctly against a suspect system. The authors did encounter one relatively recent system in their experiments that did not work correctly with *bios_memimage* and resulted in only a partial memory dump. However, since DOS solutions are likely to be used on older systems that do not support USB booting, it makes sense to use a prepared DOS-bootable CD-ROM with the necessary tools and software for acquiring memory images including the provision of USB device support[32].

If a floppy is the only option for booting then a feature-limited DOS-like operating system should be used. From here, the DOS tool *Memdump*[33] can be used to acquire a memory image [19]. However, it is limited to a maximum of approximately 3.6 GB RAM[34] since neither *Memdump* nor DOS supports PAE [19]. Although most DOS systems do not support USB storage out of the box, third-party add-ons are available that can provide the capability. Furthermore, DOS-based memory acquisition does not require that extended or expanded memory be available as the *Memdump* tool is entirely capable of acquiring high memory without them [19].

A bare-bone DOS approach can be applied on various non-USB media including IDE, SATA (and e-SATA), SCSI and SAS hard drives and other removable media including LS-120, Zip and JAZ drives. In the event that the suspect system has less than 4 GB RAM, then a minimal DOS solution using *Memdump* is ideal and is the preferred solution for several reasons. The first and most important reason is that DOS does not use high memory (memory beyond the first 1 MB) unless specifically instructed to do so using appropriately loaded memory drivers (see

[32] DOS USB drivers do exist although they are limited in their support for a broad range of devices.
[33] *Memdump* is available from http://www.tssc.de/products/tools/memdump. Although it is a commercial and proprietary tool it is available free for use, but no source code is available.
[34] Strangely, FreeDOS supports a tested maximum of 3.6 GB high memory in contrast to Windows systems which support no more than 3.2 GB memory.

CONFIG.SYS [54] and *AUTOEXEC.BAT* [55] for more information). Secondly, DOS is a very small and highly effective operating system that easily fits within the first 1 MB of linear memory, thereby having a very small memory footprint. Thirdly, DOS is very stable, especially modern DOS implementations including FreeDOS[35] and MS-DOS 7.1[36], which run on 32 and 64-bit PC platforms [19]. Finally, as has always been true of DOS, its strength lies largely on third-party tools. Various DOS systems can gain USB storage capabilities for holding memory dumps to an externally connected USB drive using several small third-party memory resident drivers (kept below the first 1 MB memory). Although Linux is in many ways superior to DOS, there are reasons why it should only be used in very specific cases that will be examined in the next subsection.

Unfortunately, there are also several caveats to using DOS to which the investigator must be attentive. The first is that a FAT16 partition has a maximum size of 2 GB and a maximum file size of 2 GB [56]. Therefore, if the amount of memory to be dumped is larger than 2 GB a FAT32 partition must be used. FAT32 supports a maximum file size of 4 GB[37] and a maximum partition size of 8 TB [56]. Both MS-DOS 7.x and FreeDOS natively support FAT16 and FAT32 [19]. It is interesting to note that MS-DOS 7.1 actually enforces a hard file size limit of 2 GB on FAT32, while FreeDOS does not and permits the maximum allowable theoretical file size [19]. In light of this information it is therefore suggested that the forensic investigator use FAT32 partitions and use FreeDOS instead of MS-DOS 7.x in order to reduce potential restrictions on the size of allowable memory dumps, up to approximately 3.6 GB in size [19].

Instead of using FAT16 or FAT32, the investigator can choose to use NTFS. However, using an NTFS partition to store a DOS-acquired memory dump requires using a program capable of providing NTFS read/write capability. In the DOS world, the only program currently capable of providing this functionality is Avira's *NTFS4DOS*[38] that is freely available but is no longer supported. This tool, however, consumes precious DOS memory, which could potentially leave no additional memory for other tools such as *Memdump*. The tool itself or some of the DOS device drivers and command shell could be loaded "high" to free room for running *Memdump* but then the memory footprint of the operating system grows. These are issues that an investigator using NTFS over FAT32 will have to contend with. However, they can be mitigated by customizing his DOS bootable device. In the authors' opinion, where memory acquisition from DOS systems is concerned, FAT32 should be the filesystem of choice.

There are other issues to using DOS. Chief among them is that a USB device cannot readily be used to boot a DOS operating system. It is possible, however, to configure a USB device such as a flash drive with a FAT32 partition and appropriate boot sector so that it can be made DOS-bootable[39] with all the necessary tools, including *Memdump*. It is also possible to create a DOS-

[35] FreeDOS is freely available and is GPL-licensed (http://www.freedos.org). Full source code is available for interested parties.
[36] MS-DOS 7.1 is an integrated component of MS Windows 98 SE (http://ms-dos7.hit.bg). Although the tool has been GPL-licensed from Microsoft its source code is not available and remains with Microsoft.
[37] Actually, it is 4 GB - 1 byte yielding a maximum FAT32 file size of 4,294,967,295 bytes [57].
[38] The tool is available from http://www.free-av.com/en/tools/11/avira_ntfs4dos_personal.html. It is commercial and proprietary in nature and no source code is available.
[39] Instructions for doing so under Linux can be found at http://wiki.fdos.org/Installation/BootDiskCreateUSB. The authors have only tried this under Linux and using FreeDOS and therefore can provide no guidance for creating an MS-DOS 7.x-based DOS USB device.

bootable CD-ROM[40] that provides USB device support so that once the operating system has finished loading off the CD, the operator has immediate access to an operator-connected (internal or external) FAT32 partitioned and formatted USB device that can be used to store the acquired memory.

If the suspect system does not support USB or CD at all (e.g. circa Pentium I and 486 technologies) then a DOS-bootable hard disk drive is the only option left. In such cases, it is important to remember that many of these older systems were particularly fussy about the maximum size of supported disks and their geometries.

2.8.2.3 Linux-related issues

Although Linux is far more capable than any version of DOS, it has two major problems with respect to cold boot memory acquisition. The primary issue is that it is very memory hungry and will readily consume far more memory than just the first 1 MB of RAM used by DOS [19]. The other issue is that, although it is capable of accessing very large amounts of memory[41], modern Linux kernels limit direct access to system memory, even to the root user, thereby severely restricting the ability of the operator to acquire a suspect system's memory [17, 19].

The first issue can be directly examined by anyone curious enough to run a specific Linux distribution on a given computer system. For example, in one test, a 16 GB RAM Dell Precision 690 workstation was booted with a hard disk drive installed with Fedora Core 11[42] 32-bit (kernel 2.6.30.10-105 i686 PAE). From the GRUB boot loader, it was configured to boot into single-user mode[43]. It was found that even when booted into single-user mode the operating system consumed approximately 76 MB RAM. Another test against the same computer system using Helix3 Pro[44], a Linux live CD, revealed that the operating system consumed approximately 550 MB RAM. These same tests were then carried out on a single processor Pentium IV system with only 2 GB installed RAM and similar results were seen. For these reasons, and unless the investigator takes the time to compile and build his very own small Linux kernel with only the minimum necessary built-in capabilities[45], it is unrealistic to use Linux for cold boot attack memory acquisition. Built-in kernel capabilities would require basic I/O support for all major interface types, possibly network support and possibly large memory support. A minimal toolset would consist of a command line interpreter (e.g. system shell), tools such as *cp*, *mv*, *dd*, *md5sum*, etc., and any other tool necessary to image memory and validate it.

In another experiment, two installations of Fedora Core 8 were compared against one another to verify the amount of memory each consumed in single-user mode. The first disk was a 32-bit

[40] Instructions for doing so can be found at http://www.hiren.info/pages/bootablecd and http://www.nu2.nu/bootcd.
[41] The 64-bit Linux kernel supports true 64-bit and therefore has support a maximum of 16 exabytes (2^{64} = 18,446,744,073,709,551,616 bytes).
[42] See http://fedoraproject.org for more information.
[43] Single-user mode is a special system mode that is used to perform recovery-based actions. It is a system administrative runlevel and is generally reserved for the root user. In this runlevel, very few processes are running on the system thereby consuming fewer resources.
[44] See http://www.e-fense.com/helix3pro.php for more information.
[45] Fortunately, Linux offers several semi-automated manners to select kernel features for inclusion/exclusion for building a new customized kernel, although these are not examined herein.

PAE-enabled kernel while the other was a 64-bit kernel. Both kernels were at the same revision level, 2.6.26.8-57. Both sets of operating system disks were used against the aforementioned Pentium IV system with 2 GB RAM. The 32-bit PAE Fedora Core 8 operating system disk was installed into the system and booted into single-user mode from GRUB. It was found that this operating required about 34 MB RAM. The system was powered off and the 32-bit operating system was removed. In its place, the 64-bit Fedora Core 8 operating system disk was used and it too was booted into single-user mode from GRUB. It was found that this operating system was using up approximately 66 MB RAM. Although this simple experiment is far from conclusive, it does indicate that between 32 and 64-bit kernels there is approximately a 1:2 ratio between memory requirements for the same kernel revision, respectively. Thus, although these results are inconclusive it is reasonable to assume that using a 64-bit Linux kernel for use in cold boot memory acquisition would require about twice as much memory as a 32-bit kernel. Therefore, when possible, the authors suggest that the investigator use a 32-bit PAE-enabled kernel to acquire a suspect system's memory as it is far less likely to overwrite further existing memory contents as compared to a 64-bit kernel. The only exception is in the case where a suspect system is equipped with more than 64 GB RAM. In such a case, a 64-bit kernel must be used.

Certainly, 2.4.x generation kernels consume less memory resources than their modern 2.6.x counterparts. However, these antiquated kernels and their adjoining distributions do not support much of today's modern hardware, including full 64-bit memory addressing, new device interface technologies, peripheral system cards, etc. [58]. Thus, the investigator may have to decide whether to develop proficiency with 2.4.x kernels or 2.6.x kernels. The use of a 2.4.x or 2.6.x kernel is directly related to the second issue that surrounds the use of Linux. More specifically, the issue is such that prior to 2.6.x kernels the root user had direct access to the system's memory device. Under 2.6.x and newer kernels, the root user no longer does [19]. The reason for this has to do with new security mechanisms introduced into the 2.6.x kernel to restrict the ability for rootkits to compromise a Linux system. It may be possible in some instances to disable the protection of direct memory access by booting the kernel appended with boot parameter *strict-devmem=0*. Furthermore, it may be necessary to run the system command *setcap*[46] to remove file-based protection mechanisms, specifically the use of the CAP_SYS_RAWIO raw I/O capability. However, in the authors' own experiments against Red Hat 9 (2.4.x kernel), Fedora Core-based 32 and 64-bit systems (2.6.x kernel), and Ubuntu Debian-based 32 and 64-bit systems (2.6.x kernel), memory acquisition under 2.6.x kernels was very shaky at best, although relatively successful under Red Hat 9. [19]

The results obtained in the experiments conducted in [19] are due in part to the tools used to acquire system memory and the various protection mechanisms around the kernel. Specifically, Red Hat 9, based on a 2.4.x kernel, supports PAE but does not have any kernel memory protection-based mechanisms. Therefore, memory acquisition against its memory device succeeded. However, even when implementing the aforementioned countermeasures to the various Fedora Core and Ubuntu-based systems examined in [19], memory acquisition was very inconsistent and sometimes crashed the test systems.

Linux-based memory acquisition, when the kernel provides direct memory access, is not particularly difficult to do. Various tools can be used such as the system tool *dd* or a third-party

[46] The tool *setcap* is used to set system file capabilities

tool *Memdump*[47] developed by Wietse Venema [19]. Compiling a customized 2.6.x with the memory protection mechanisms left out should work to acquire memory directly from the system's memory device, although the authors did not directly examine this in [19] and therefore cannot provide any definite guidance on the matter.

2.8.2.4 Windows-related issues

The authors have not yet performed any direct analyses concerning the utilization of Windows-based memory. As such, the authors cannot currently provide any comments on this matter. Examination of Windows memory will be done at a later time in a future report by the authors.

[47] See http://www.porcupine.org/forensics/tct.html for more information.

3 Final analysis and conclusion

The authors have successfully tested many different sizes of computer memory acquisition using the Princeton team's USB *scraper* tool. These tests were conducted prior to conducting the various experiments described in Section 4.3 in order to minimize error during experimentation.

Tests varied using both the 32 and 64-bit versions of the *scraper* tool against systems with as little as 256 MB RAM to upwards of 24 GB RAM. However, tests conducted against 32-bit operating systems with the 32-bit version of the *scraper* tool yielded a maximum memory acquisition of 3.2 GB RAM. In order to acquire more memory would require the use of the 64-bit *scraper* tool. Tests where more than 14 GB RAM were to be captured were carried out using a USB-based hard disk drive rather than with the 16 GB USB flash drive. All memory images were extracted using the either the 32 or 64-bit version of the *usbdump* command.

Additional experimentation is required to determine the cold boot acquisition software's practical upper limit. Based on the tool's source code, however, it appears that in 64-bit mode the default upper limit is 64 GB RAM although this value can be changed.[48]

The results obtained while conducting the cold boot attack experiments have yielded interesting if not contradictory findings with respect to the Princeton team's analysis and report. Based solely on the results obtained, some systems appear to be altogether incapable of retaining memory-based data while others appear to be particularly apt. More specifically, some systems appear to be vulnerable to computer memory-based data remanence while others remain relatively immune to it. Several factors may be at play here although the predominance of one factor over another is not entirely clear-cut.

Examining the data and results from the first six experiments conducted in Section 4.3, only two systems exhibited data remanence, specifically the Dell CPx and Dell E521. None of the other systems appeared to exhibit characteristics of data remanence (for specifics concerning these two computer systems please see sections 4.3.2, 4.3.6 and 4.3.7).

The existence of data remanence was determined using three simple tests. The first test was string extraction using the UNIX *strings* command. This command made it possible to find and extract any text-like data string from a file whether textual or binary in nature (such as memory image files). Thus, by using the *strings* tool, it was possible to objectively acquire some measure of data remanence. Depending on how many strings were found and from which portions of memory (low or high memory) it was possible to readily determine whether data remanence had in fact occurred. The second test involved keyword searches and the third test verified for the existence of extractable AES keys to demonstrate the usefulness of the cold boot attack for recovering memory-resident cryptographic keys.

Thus, based on these tests (strings, keyword searches and encryption key detection) it is possible to determine which systems exhibited data remanence. Although strings were detected for each

[48] Upon reading the source code and documentation for the *bios_memimage* program, it appears that entry "*#define MAPS 63*" from file *x86-64/mapmem.c* can be modified to 64 or higher to support more than 64 GB RAM.

experiment, the reader should not take this as indication of data remanence. Instead, even in systems where data remanence was not exhibited, strings were always present in memory based on the completion of each system's POST and BIOS initialization. Furthermore, the use and execution of the *scraper* program also leaves telltale traces in memory. Thus, combining this with the fact that data is inserted into memory by the POST and BIOS initializations satisfactorily explains why systems which do not demonstrator data remanence nonetheless have extractable memory-based strings.

Indeed, an in-depth analysis of the memory images acquired on systems that did not exhibit data remanence shows that all their detectable strings were found within approximately the first megabyte of linear computer memory. This is logical since PC POST and BIOS initializations generally occur within this region of memory. The fact that executable code and data strings from the *scraper* programs were also found here in low memory is also not surprising. However, while the exact amount of low memory required for POST/BIOS initialization and *scraper* code execution differs according to the underlying system, detectable strings were never found beyond the first 1.2 MB of linear computer memory for systems that were deemed unsusceptible to data remanence. However, for the Dell CPx and E521, the vast majority of detectable strings and searchable keywords were instead found in high memory. Moreover, detected and extracted AES encryption keys were only found in high memory.

Thus, upon terminating the various memory analyses for the first six experiments, it became apparent that something was unique to both the Dell CPx and E521. To investigate, additional experiments were required to determine if those properties came from the memory modules or the computer system. Unfortunately, no additional experiments could be conducted against the Dell CPx, as no other compatible memory modules were readily available for further experimentation. However, additional memory modules for the Dell E521 were available from the Dell 3100 and GX620, along with the original memory modules that shipped with the system, 2 Samsung 512MB modules. The Dell 3100 and GX620 each had one pair of Micron 2x512MB memory modules ready for additional experimentation involving the E521.

Three other sets of experiments using the two pairs of Micron 2x512MB memory modules and the Samsung 2x512MB memory modules were conducted. After conducting these final three experiments (see Section 4.3.7 for more information), it was determined that all three sets of memory modules were in fact susceptible to data remanence while in use with the Dell E521. Interestingly, neither pair of Micron 512MB memory modules presented any evidence of data remanence while in use with either the Dell 3100 or GX620. Now, while all three pairs of memory exhibited data remanence, only the pair of Samsung memory modules actually had intact memory images as determined by running the aforementioned integrity tests. These tests indicated that the vast majority of detectable strings and searchable keywords resided in high memory and that four unique *TrueCrypt* AES encryption keys were detected in high memory.

Thus, it can be reasonably concluded that the Samsung memory modules were capable of maintaining the integrity of the electric charges of the capacitor-based memory cells residing in the physical memory modules. Moreover, data integrity was maintained long enough to preserve not only the *TrueCrypt* encryption keys but also to enable both the detection of strings and keyword searches. Further manual analysis on the two Micron modules pair showed that there was indeed data remanence, but the data had degraded too rapidly to be usefully extracted.

Based on these experiments, one can conclude that there is much variation for various make and models of computer memory and systems. Not all memory modules retain their electrical charges for the same amount of time although this also depends greatly on the underlying computer system. This brings to bear the potential for motherboard residual capacitance to play a significant role in maintaining the electrical charges of memory cells.

Unfortunately, there is no known list detailing electrical charge retention for various memory modules and motherboards. Moreover, there will likely be variations between sets of memory modules of the same make and model due to tolerance levels inherent throughout the manufacturing process. The forensic investigator who finds himself forced to use the cold boot attack should do so only as a last resort in order to attempt to forcibly acquire a suspect system's memory. Regrettably, he will find that the odds are stacked against him. There are simply too many factors which need to be known ahead of time and which are simply not known nor are they ever likely to be known to any large extent.

Therefore, in conclusion, the cold boot attack should not be viewed as the primary method for acquiring a suspect computer system's memory. Instead, other techniques including both software and hardware-based acquisition (e.g. FireWire) should be attempted prior to carrying out a cold boot attack against said system. However, should a situation occur where the aforementioned techniques are either not available (e.g. lack of FireWire connection or system login console or remote memory acquisition is not possible) or are ineffectual, then the cold boot attack may be administered assuming that the investigator understands both how and where problem may arise and go awry.

As this study has shown, the cold boot attack cannot be established as being particularly forensically sound or reliable since in most of the experiments conducted herein memory-resident encryption keys could not be consistently found or extracted although they should have been. The same can also be said for the various strings and keyword searches which should have turned up far more strings and keywords than were found for most of the experiments.

Moreover, as has been demonstrated, merely the act of flash-freezing computer memory does not guarantee the successful acquisition of said memory. Other factors and variables already examined have fully examined these issues and their underlying causes. Thus, it is the opinion of the authors of this study that the cold boot attack can be useful in some cases to acquire a suspect system's memory but that this method should not be considered a panacea and instead should be used as a last resort when all other avenues have been exhausted.

Finally, even a successful acquisition which has suffered little to no degradation will likely not stand up in a court of law as sound evidence, at least until jurisprudence has occurred and the integrity of the acquired memory can be demonstrated to be intact using a sound and understandable methodology.

The search continues to establish a more reliable way of acquiring the memory of a suspect's computer...

4 Experiments

4.1 Objective

This section details the various experiments the authors have conducted against the assortment of computer systems outlined in Annex A.1.1. The purpose of these experiments was to determine which computer systems are susceptible to the cold boot attack. Moreover, the authors thought it prudent to determine if any of these systems would also be susceptible to a warm boot attack. A warm boot attack is the same as a cold boot attack but without flash freezing the computer system's memory.

The authors have brought together an array of various computer systems ranging from Pentium II era technology to relatively modern AMD and Pentium IV computer systems to high-end Xeon systems in the hopes of discovering some generic properties governing computer memory data remanence. Details on these systems are provided in Annex A.1.1. These computer systems are using memory modules of different sizes, with the smallest memory modules being 128 MB in size (PC-100/100 MHz) and the largest being 4 GB in size (PC-4200/533 MHz), with several other sizes in between. There is much variety in the fabrication of these memory modules as they are manufactured by different companies including Hyundai, Micron, Patriot, Kingston, and Samsung. The authors believe that there is enough computer memory and platform diversity here to attempt to develop preliminary conclusions that the wider audience of forensics investigators can substantiate.

The experiments consisted of loading a preinstalled operating system from the computer's hard disk drive and running an instance of *TrueCrypt* to determine if any traces of cryptographic activity can be coaxed from the acquired memory images. Either a warm boot or a cold boot attack will be performed where the result will be the acquisition of a memory image. This memory image will then be analysed to determine if any data remanence has occurred and what, if anything, can be extracted.

4.2 Notes

4.2.1 A note about TrueCrypt

TrueCrypt[49] is free and open source disk encryption software for Windows 7/Vista/XP, Mac OS X, and Linux. The functionality of interest for these experiments is the possibility of creating a volume file that, once mounted with the right password/keyfile combination, will look to Windows like a regular partition. *TrueCrypt* lets the user choose from various combinations of encryption algorithms (AES-256, Serpent, and Twofish) and hashes (RIPEMD-160, SHA-512 and Whirlpool).

The Princeton tool, *aeskeyfind*, was used in the analysis of the acquired memory. This tool is supposed to find AES keys in acquired memory, even if some bits have swapped in the key.

[49] See http://www.truecrypt.org for more information.

4.2.2 A note about *scraper* memory ranges

The *scraper* memory acquisition tool, a key component of the Princeton team's *bios_memimage* software package, acquires a computer system's memory according to various memory ranges. These ranges differ in size depending on the amount of memory residing on the host system.

4.2.3 A note about BIOS memory area

After having completed much experimentation using the *scraper* and *usbdump* tool from the *bios_memimage* software package, the authors have found that in every experiment the computer system's BIOS information was always in approximately the first 1 MB of memory. This fact explains why even though many memory images captured throughout the various experiments did not succeed in preserving previous memory contents through the cold boot attack, there was always some data that was consistently acquired, specifically the BIOS data.

4.3 Experiments

4.3.1 Experiments against System 1

4.3.1.1 Background

Windows XP Professional 64-bit was installed onto a Dell Precision 690 workstation (see Table 61, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 2 and *TrueCrypt 6.0a* was installed. The only partition on the system disk was an NTFS partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal hard disk drive was set as the primary boot device and the first of the two internal CD/DVD drives was set as the secondary boot device. The BIOS was configured to automatically power up the computer the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive), it sufficed to press the F12 key at the system's POST in order to access the system boot menu. Finally, the system's BIOS was configured for fast-booting in the hope of circumventing memory ECC checking.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 250 GB USB-enabled hard disk drive was used (see Table 68, Annex 1.2 for more details). A 64-bit version of the *bios_memimage* software package *scraper* program was compiled and copied to sector 0 of the aforementioned hard disk drive.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

Upon completion of each specific experiment, the USB hard disk drive was connected to a Linux workstation for memory image extraction and analysis. Once completed, the USB device was zero-fill wiped to ensure there would be no potential contamination for future experiments. Due to the size of the memory image, a 64-bit version of the *usbdump* program had to be used to successfully extract a memory image larger than 4 GB from the USB device.

Therefore, four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2

- Experiment 3: Cold boot attack attempt 1
- Experiment 4: Cold boot attack attempt 2

The experimental data from this computer system and its analysis are examined further on.

4.3.1.2 Experimental specifics

4.3.1.2.1 Warm boot specifics

The computer system was powered on and upon the successful booting of Windows the administrative user was logged in to the system. At this time, the *TrueCrypt 6.0a* program was started up and the aforementioned cryptographic volume *vol.tc* was mounted. Upon successfully mounting the cryptographic volume the system's power was immediately removed and then quickly returned whereupon the aforementioned USB hard disk drive was attached to one of the system's front USB ports. Since the system's BIOS had been previously configured to power on the instant power was reattached it was not necessary to press or hold down the system's power button.

Once power was re-established and the system's POST was visible on the system's displays the F12 key was pressed to access the boot menu. Upon completion of the system's POST, the boot menu appeared at which time the attached USB device was selected for booting the system where the 64-bit *scraper* program booted and began its memory acquisition. Upon completing its memory acquisition the system was automatically rebooted by the *scraper* program at which time the operator presiding over the experiment removed the USB hard disk drive from the system. The system was then manually powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on and booted back into Windows XP 64-bit. Then the second instance of this experiment was carried out.

4.3.1.2.2 Cold boot specifics

The system was booted up into Windows whereupon the administrative user was logged into the system. *TrueCrypt 6.0a* was started up and the cryptographic volume mounted. Upon its successful mounting the system's side chassis panel was opened in order to expose its internal components. Upon identifying the system's memory riser board all memory modules were sufficiently flash-frozen. Once fully flash-frozen power was briefly removed and then reconnected. Upon seeing the system's POST, the USB hard disk drive was connected to the system's front USB port and the F12 key was pressed to access the system boot menu. From the boot menu the USB device was selected for booting at which time the *scraper* program was initiated and began its memory acquisition.

Upon the successful completion of the memory acquisition, the system was automatically rebooted by the *scraper* program whereupon the operator presiding over the experiment removed the USB hard disk drive from the system. The system was then manually powered off and left in this state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back

on and allowed to boot into Windows XP and the second instance of this experiment was carried out.

4.3.1.3 Collected data

Various information and notes taken by the authors while conducting these experiments are presented in the following tables:

Table 1: System 1 – Data collected for Experiment 1

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	17,177,942,016
Scraper detected memory ranges (in bytes)	Range 0: 650,240 Range 1: 3,218,648,064 Range 2: 12,884,901,888 Range 3: 1,073,741,824
Scraper detected USB disk size (in bytes)	951,246,848
Time required to acquire memory	Approximately 14 minutes
Recovered memory image size (in bytes)	17,177,942,016
Name given to extracted memory image	690_warm1.dd

Table 2: System 1 – Data collected for Experiment 2

Time required to remove power and reattach it	Approximately 2 seconds
Scraper total detected memory (in bytes)	17,177,942,016
Scraper detected memory ranges (in bytes)	Range 0: 650,240 Range 1: 3,218,648,064 Range 2: 12,884,901,888 Range 3: 1,073,741,824
Scraper detected USB disk size (in bytes)	951,246,848
Time required to acquire memory	Approximately 14 minutes
Recovered memory image size (in bytes)	17,177,942,016
Name given to extracted memory image	690_warm2.dd

Table 3: System 1 – Data collected for Experiment 3

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	17,177,942,016
Scraper detected memory ranges (in bytes)	Range 0: 650,240 Range 1: 3,218,648,064 Range 2: 12,884,901,888 Range 3: 1,073,741,824
Scraper detected USB disk size (in bytes)	951,246,848
Time required to acquire memory	Approximately 14 minutes
Recovered memory image size (in bytes)	17,177,942,016
Name given to extracted memory image	690_cold1.dd

Table 4: System 1 – Data collected for Experiment 4

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	17,177,942,016
Scraper detected memory ranges (in bytes)	Range 0: 650,240 Range 1: 3,218,648,064 Range 2: 12,884,901,888 Range 3: 1,073,741,824
Scraper detected USB disk size (in bytes)	951,246,848
Time required to acquire memory	Approximately 14 minutes
Recovered memory image size (in bytes)	17,177,942,016
Name given to extracted memory image	690_cold2.dd

4.3.1.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running, it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys in memory. Of course, this assumes that the memory contents had not already been cleared by the system's ECC memory-checking mechanism. Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 5: System 1 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,339 / 6,247 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	1,017 / 5,934 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 6: System 1 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,339 / 6,247 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	1,017 / 5,934 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 7: System 1 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,192 / 5,999 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	939 / 5,696 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 8: System 1 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,192 / 5,999 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	940 / 5,745 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

4.3.2 Experiments against System 2

4.3.2.1 Background

Windows XP Professional 32-bit was installed onto this system, a Dell Dimension E521 desktop (see Table 62, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 3 and *TrueCrypt 6.0a* was installed. The only partition on the system disk was an NTFS partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal hard disk drive was set as the primary boot device and the secondary boot device was set to the internal CD/DVD drive. The BIOS was configured to automatically power up the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive) at boot time it sufficed to press the F12 key at the system's POST in order to access the system boot menu.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 250 GB USB-enabled hard disk drive was used (see Table 68, Annex A.1.2 for more details). A 64-bit version of the *bios_memimage* software package *scraper* program was compiled and copied to sector 0 of the aforementioned hard disk drive.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

Upon completion of each specific experiment, the USB hard disk drive was connected to a Linux workstation for memory image extraction and analysis. Once completed the USB device was zero-fill wiped to ensure there would be no potential contamination for future experiments. Due to the size of the memory image, a 64-bit version of the *usbdump* program had to be used to successfully extract a memory image larger than 4 GB from the USB device.

Therefore, four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2
- Experiment 3: Cold boot attack attempt 1

- Experiment 4: Cold boot attack attempt 2

The experimental data from this computer system and its analysis are examined further on.

4.3.2.2 Experimental specifics

4.3.2.2.1 Warm boot specifics

The computer system was powered on and upon the successful booting of Windows the administrative user was logged in to the system. At this time, the *TrueCrypt 6.0a* program was started up and the aforementioned cryptographic volume *vol.tc* was mounted. Upon successfully mounting the cryptographic volume the system's power was immediately removed and then quickly returned whereupon the aforementioned USB hard disk drive was attached to one of the system's front USB ports. Since the system's BIOS had been previously configured to power on the instant power was reattached it was not necessary to press or hold down the system's power button.

Once power was re-established and the system's POST was visible on the system's displays the F12 key was pressed to access the boot menu. Upon completion of the system's POST, the boot menu appeared at which time the attached USB device was selected for booting the system where the 64-bit *scraper* program booted and began its memory acquisition. Upon completing its memory acquisition the system was automatically rebooted by the *scraper* program at which time the operator presiding over the experiment removed the USB hard disk drive from the system. The system was then manually powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on and booted back into Windows XP 32-bit. Then the second instance of this experiment was carried out.

4.3.2.2.2 Cold boot specifics

The system was booted up into Windows whereupon the administrative user was logged into the system. *TrueCrypt 6.0a* was started up and the cryptographic volume mounted. Upon its successful mounting the system's side chassis panel was opened in order to expose its internal components. Upon identifying the system's memory modules, they were all sufficiently flash-frozen. Once fully flash-frozen, power was briefly removed and then reconnected. Upon seeing the system's POST, the USB hard disk drive was connected to the system's front USB port and the F12 key was pressed to access the system boot menu. From the boot menu the USB device was selected for booting at which time the *scraper* program was initiated and began its memory acquisition.

Upon the successful completion of the memory acquisition, the system was automatically rebooted by the *scraper* program whereupon the operator presiding over the experiment removed the USB hard disk drive from the system. The system was then manually powered off and left in this state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back on and allowed to boot into Windows XP and the second instance of this experiment was carried out.

4.3.2.3 Collected data

4.3.2.3.1 Warm boot attack

Various information and notes taken by the authors while conducting these experiments are presented in the following tables:

Table 9: System 2 – Data collected for Experiment 1

Time required to remove power and reattach it	Approximately 2 seconds
Scraper total detected memory (in bytes)	4,159,172,608
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,621,650,432 Range 2: 536,870,912
Scraper detected USB disk size (in bytes)	951,246,336
Time required to acquire memory	Approximately 4 minutes
Recovered memory image size (in bytes)	3,622,301,696
Name given to extracted memory image	E521_warm1.dd

Table 10: System 2 – Data collected for Experiment 2

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	4,159,172,608
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,621,650,432 Range 2: 536,870,912
Scraper detected USB disk size (in bytes)	951,246,336
Time required to acquire memory	Approximately 4 minutes
Recovered memory image size (in bytes)	3,622,301,696
Name given to extracted memory image	E521_warm2.dd

Table 11: System 2 – Data collected for Experiment 3

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	4,159,172,608
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,621,650,432 Range 2: 536,870,912
Scraper detected USB disk size (in bytes)	951,246,336
Time required to acquire memory	Approximately 4 minutes
Recovered memory image size (in bytes)	3,622,301,696
Name given to extracted memory image	E521_cold1.dd

Table 12: System 2 – Data collected for Experiment 4

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	4,159,172,608
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,621,650,432 Range 2: 536,870,912
Scraper detected USB disk size (in bytes)	951,246,336
Time required to acquire memory	Approximately 4 minutes
Recovered memory image size (in bytes)	3,622,301,696
Name given to extracted memory image	E521_cold2.dd

4.3.2.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running, it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys in memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 13: System 2 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,059,764 / 8,158,082 / 227,562 / 2,675
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	711,091 / 3,191,729 / 178,934 / 2,202
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	53 / 53 / 19 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	26 / 25 / 12 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	860 / 805 / 204 / 0
AES keys founds / Unique keys found	None / None

Table 14: System 2 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,000,610 / 6,680,431 / 182,418 / 1,011
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	613,574 / 2,970,877 / 123,868 / 707
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	41 / 41 / 65 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	38 / 37 / 22 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	750 / 710 / 390 / 0
AES keys founds	3 256-bit keys / 3 unique 256-bit keys

Table 15: System 2 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,111,859 / 9,250,124 / 291,357 / 686
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	733,581 / 3,145,853 / 160,912 / 500
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	29 / 29 / 40 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	15 / 15 /10 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	716 / 662 / 194 / 0
AES keys founds / Unique keys found	None / None

Table 16: System 2 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,014,018 / 8,208,387 / 265,326 / 390
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	727,228 / 3,386,873 / 174,874 / 325
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	28 / 28 / 24 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	13 / 11 / 12 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	412 / 392 / 127 / 0
AES keys founds / Unique keys found	None / None

4.3.3 Experiments against System 3

4.3.3.1 Background

This system, a Dell Latitude E6500 laptop (see Table 63, Annex A.1.1 for more details), was booted using an Ubuntu 9.04 64-bit Linux Live CD with Linux kernel 2.6.28-11 generic #42-Ubuntu SMP x86_64. The laptop's internal hard disk drive was removed and the system was running solely from the optical drive. Using a small supplementary USB-based flash drive the 64-bit version of *TrueCrypt 6.3* was installed atop the Live CD operating system

From the BIOS, the internal optical drive was set as the primary boot device and the secondary boot device was set to the absent hard disk drive. The BIOS was then configured to automatically power up the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive) at boot time it sufficed to press the F12 key at the system's POST in order to access the system boot menu.

The laptop's battery was also removed throughout the various experiments in order to ensure that any additional sources of electrical power that could potentially skew the results were absent.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the aforementioned small supplementary USB-based flash drive and was accessed from that location for each of the experiments. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 16 GB USB flash drive was used (see Table 67, Annex A.1.2 for more details). A 64-bit version of the *bios_memimage* software package *scraper* program was compiled and copied to sector 0 of the aforementioned flash drive.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

Upon completion of each specific experiment, the USB flash drive was connected to a Linux workstation for memory image extraction and analysis. Once completed the USB device was zero-fill wiped to ensure there would be no potential contamination for future experiments. Due to the size of the memory image, a 64-bit version of the *usbdump* program had to be used to successfully extract a memory image larger than 4 GB from the USB device.

Therefore, four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2

- Experiment 3: Cold boot attack attempt 1
- Experiment 4: Cold boot attack attempt 2

The experimental data from this computer system and its analysis are examined further on.

4.3.3.2 Experimental specifics

4.3.3.2.1 Warm boot specifics

The computer system was powered on and booted from the Live CD where upon completion of the boot the predefined system user was automatically logged in. At this time, the small supplementary USB flash drive was connected to the system and mounted. From there the *TrueCrypt 6.3* program was installed atop the Live CD and upon its installation, the pre-existing cryptographic volume on the USB flash drive was mounted. Upon successfully mounting the cryptographic volume the system's power was immediately removed and then quickly returned whereupon the aforementioned small supplementary USB flash drive was removed and substituted with the 16 GB flash drive all while waiting for the system's POST to appear. Since the system's BIOS had been previously configured to power on the instant power was reattached it was not necessary to press or hold down the system's power button.

Once power was re-established and the system's POST was visible on the system's displays the F12 key was pressed to access the boot menu. Upon completion of the system's POST, the boot menu appeared at which time the attached USB device was selected for booting the system where the 64-bit *scraper* program booted and began its memory acquisition. Upon completing its memory acquisition the system was automatically rebooted by the *scraper* program and at which time the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on and booted back into its Linux Live CD operating system. Then the second instance of this experiment was carried out.

4.3.3.2.2 Cold boot specifics

The system was booted up using the Ubuntu Live CD where upon completion of the boot the predefined system user was automatically logged in. At this time, the small supplementary USB flash drive was connected to the system and mounted. Then the *TrueCrypt 6.3* program was installed atop the Live CD and upon its installation, the pre-existing cryptographic volume on the USB flash drive was mounted. Upon its successful mounting the system's side chassis panel was opened in order to expose its internal components. Upon identifying the system's memory modules, they were all sufficiently flash-frozen. Once fully flash-frozen power was briefly removed and then reconnected at which time the aforementioned small supplementary USB flash drive was removed and substituted with the 16 GB flash drive all while waiting for the system's POST to appear. Upon seeing the system's POST, the F12 key was pressed to access the system boot menu. From the boot menu the USB device was selected for booting at which time the *scraper* program was initiated and began its memory acquisition.

Upon the successful completion of the memory acquisition, the system was automatically rebooted by the *scraper* program whereupon the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off and left in this state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back on and again booted using the Ubuntu Live CD and then the second instance of this experiment was carried out.

4.3.3.3 Data collected

Various information and notes taken by the authors while conducting these experiments are presented in the following tables:

Table 17: System 3 – Data collected for Experiment 1

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	4,282,295,296
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,744,781,312 Range 2: 536,682,720
Scraper detected USB disk size (in bytes)	3,246,391,296
Time required to acquire memory	Approximately 6.5 minutes
Recovered memory image size (in bytes)	4,282,295,296
Name given to extracted memory image	E6500_warm1.dd

Table 18: System 3 – Data collected for Experiment 2

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	4,282,295,296
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,744,781,312 Range 2: 536,682,720
Scraper detected USB disk size (in bytes)	3,246,391,296
Time required to acquire memory	Approximately 6.5 minutes
Recovered memory image size (in bytes)	4,282,295,296
Name given to extracted memory image	E6500_warm2.dd

Table 19: System 3 – Data collected for Experiment 3

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	4,282,295,296
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,744,781,312 Range 2: 536,682,720
Scraper detected USB disk size (in bytes)	3,246,391,296
Time required to acquire memory	Approximately 6.5 minutes
Recovered memory image size (in bytes)	4,282,295,296
Name given to extracted memory image	E6500_cold1.dd

Table 20: System 3 – Data collected for Experiment 4

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory (in bytes)	4,282,295,296
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 3,744,781,312 Range 2: 536,682,720
Scraper detected USB disk size (in bytes)	3,246,391,296
Time required to acquire memory	Approximately 6.5 minutes
Recovered memory image size (in bytes)	4,282,295,296
Name given to extracted memory image	E6500_cold2.dd

4.3.3.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Linux operating system with *TrueCrypt 6.3* running it should be possible to extract both various Linux and application-related strings and potentially AES-based encryption keys in memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 21: System 3 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	3,615 / 10,520 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	2,188 / 9,633 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Linux"	1 / 1 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "kernel"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 22: System 3 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	3,615 / 10,520 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	2,188 / 9,637 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Linux"	1 / 1 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "kernel"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 23: System 3 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	3,615 / 10,520 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	2,188 / 9,637 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Linux"	1 / 1 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "kernel"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 24: System 3 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	3,615 / 10,520 / 1 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	2,188 / 9,633 / 1 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Linux"	1 / 1 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "kernel"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

4.3.4 Experiments against System 4

4.3.4.1 Background

Windows XP Professional 32-bit was installed onto this system, a Dell Dimension 3100 desktop (see Table 64, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 3 and *TrueCrypt 6.0a* was installed. The only partition on the system disk was an NTFS partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal hard disk drive was set as the primary boot device and the first of the two internal CD/DVD drives was set as the secondary boot device. The BIOS was configured to automatically power up the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive) at boot time it sufficed to press the F12 key at the system's POST in order to access the system boot menu.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 16 GB USB flash drive was used (see Table 67, Annex A.1.2 for more details). Since this system's memory was less than 4 GB, a 32-bit instance of the *scraper* program was compiled and copied to sector 0 of the aforementioned flash drive.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

Upon completion of each specific experiment, the USB flash drive was connected to a Linux workstation for memory image extraction and analysis. Once completed the USB device was zero-fill wiped to ensure there would be no potential contamination for future experiments. Since the memory image would be less than 4 GB in size a 32-bit version of the *usbdump* tool was used to extract the memory image from the USB device.

Therefore, four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2
- Experiment 3: Cold boot attack attempt 1

- Experiment 4: Cold boot attack attempt 2

The experimental data from this computer system and its analysis are examined further on.

4.3.4.2 Experimental specifics

4.3.4.2.1 Warm boot specifics

The computer system was powered on and upon the successful booting of Windows the administrative user was logged in to the system. At this time, the *TrueCrypt 6.0a* program was started up and the aforementioned cryptographic volume *vol.tc* was mounted. Upon successfully mounting the cryptographic volume the system's power was immediately removed and then quickly returned whereupon the aforementioned USB flash drive was attached to one of the system's front USB ports. Since the system's BIOS had been previously configured to power on the instant power was reattached it was not necessary to press or hold down the system's power button.

Once power was re-established and the system's POST was visible on the system's displays the F12 key was pressed to access the boot menu. Upon completion of the system's POST the boot menu appeared at which time the attached USB device was selected for booting the system where the 32-bit *scraper* program booted and began its memory acquisition. Upon completing its memory acquisition the system was automatically rebooted by the *scraper* program and the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on and booted back into Windows XP 32-bit. Then the second instance of this experiment was carried out.

4.3.4.2.2 Cold boot specifics

The system was booted up into Windows whereupon the administrative user was logged into the system. *TrueCrypt 6.0a* was started up and the cryptographic volume mounted. Upon its successful mounting the system's side chassis panel was opened in order to expose its internal components. Upon identifying the system's memory modules, they were all sufficiently flash-frozen. Once fully flash-frozen power was briefly removed and then reconnected. Upon seeing the system's POST, the USB flash drive was connected to the system's front USB port and the F12 key was pressed to access the system boot menu. From the boot menu the USB device was selected for booting at which time the *scraper* program was initiated and began its memory acquisition.

Upon the successful completion of the memory acquisition, the system was automatically rebooted by the *scraper* program whereupon the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off and left in this state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back on and allowed to boot into Windows XP and the second instance of this experiment was carried out.

4.3.4.3 Data collected

Various information and notes taken by the authors while conducting these experiments are presented in the following tables:

Table 25: System 4 – Data collected for Experiment 1

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,063,422,976 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,062,767,616 bytes
Scraper detected USB disk size	3,246,391,296
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,063,422,976 bytes
Name given to extracted memory image	3100_warm1.dd

Table 26: System 4 – Data collected for Experiment 2

Time required to remove power and reattach it	Approximately 2 seconds
Scraper total detected memory	1,063,422,976 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,062,767,616 bytes
Scraper detected USB disk size	3,246,391,296 bytes
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,063,422,976 bytes
Name given to extracted memory image	3100_warm2.dd

Table 27: System 4 – Data collected for Experiment 3

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,063,422,976 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,062,767,616 bytes
Scraper detected USB disk size	3,246,391,296
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,063,422,976 bytes
Name given to extracted memory image	3100_cold1.dd

Table 28: System 4 – Data collected for Experiment 4

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,063,422,976 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,062,767,616 bytes
Scraper detected USB disk size	3,246,391,296
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,063,422,976 bytes
Name given to extracted memory image	3100_cold2.dd

4.3.4.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys in memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 29: System 4 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	554 / 3,622 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	470 / 3,556 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 30: System 4 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	554 / 3,622 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	470 / 3,556 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 31: System 4 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	554 / 3,622 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	470 / 3,558 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 32: System 4 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	554 / 3,624 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	470 / 3,558 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

4.3.5 Experiments against System 5

4.3.5.1 Background

Windows XP Professional 32-bit was installed onto this system, a Dell OptiPlex GX620 (see Table 65, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 3 and *TrueCrypt 6.0a* was installed. The only partition on the system disk was an NTFS partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal hard disk drive was set as the primary boot device and the secondary boot device was set to the internal CD/DVD drive. The BIOS was configured to automatically power up the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive) at boot time it sufficed to press the F12 key at the system's POST in order to access the system boot menu.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 16 GB USB flash drive was used (see Table 67, Annex A.1.2 for more details). Since this system's memory was less than 4.0 GB, a 32-bit instance of the *scraper* program was compiled and copied to sector 0 of the aforementioned flash drive.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

Upon completion of each specific experiment, the USB flash drive was connected to a Linux workstation for memory image extraction and analysis. Once completed the USB device was zero-fill wiped to ensure there would be no potential contamination for future experiments. Since the memory image would be less than 4 GB in size a 32-bit version of the *usbdump* tool was used to extract the memory image from the USB device.

Thus, in all four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2
- Experiment 3: Cold boot attack attempt 1

- Experiment 4: Cold boot attack attempt 2

The data from the computer system (data) and the analysis of the memory images (analyses) are examined further on in this section.

4.3.5.2 Experimental specifics

4.3.5.2.1 Warm boot specifics

The computer system was powered on and upon the successful booting of Windows the administrative user was logged in to the system. At this time, the *TrueCrypt 6.0a* program was started up and the aforementioned cryptographic volume *vol.tc* was mounted. Upon successfully mounting the cryptographic volume the system's power was immediately removed and then quickly returned whereupon the aforementioned USB flash drive was attached to one of the system's front USB ports. Since the system's BIOS had been previously configured to power on the instant power was reattached it was not necessary to press or hold down the system's power button.

Once power was re-established and the system's POST was visible on the system's displays the F12 key was pressed to access the boot menu. Upon completion of the POST, the boot menu appeared at which time the attached USB device was selected for booting the system where the 32-bit *scraper* program booted and began its memory acquisition. Upon completing its memory acquisition the system was automatically rebooted by the *scraper* program and the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on and booted back into Windows XP 32-bit. Then the second instance of this experiment was carried out.

4.3.5.2.2 Cold boot specifics

The system was booted up into Windows whereupon the administrative user was logged into the system. *TrueCrypt 6.0a* was started up and the cryptographic volume mounted. Upon its successful mounting the system's side chassis panel was opened in order to expose its internal components. Upon identifying the system's memory modules, they were all sufficiently flash-frozen. Once fully flash-frozen power was briefly removed and then reconnected. Upon seeing the system's POST, the USB flash drive was connected to the system's front USB port and the F12 key was pressed to access the system boot menu. From the boot menu the USB device was selected for booting at which time the *scraper* program was initiated and began its memory acquisition.

Upon the successful completion of the memory acquisition, the system was automatically rebooted by the *scraper* program whereupon the operator presiding over the experiment removed the USB flash drive from the system. The system was then manually powered off and left in this state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back on and allowed to boot into Windows XP and the second instance of this experiment was carried out.

4.3.5.3 Data collected

Various information and notes taken by the authors while conducting this portion of the experiments are presented in the following tables:

Table 33: System 5 –Data collected for Experiment 1

Time required to remove power and reattach it	Approximately 1.5 seconds
Scraper total detected memory	1,071,803,392 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,071,148,032 bytes
Scraper detected USB disk size	3,246,391,296 bytes
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,071,803,392 bytes
Name given to extracted memory image	GX620_warm1.dd

Table 34: System 5 –Data collected for Experiment 2

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,071,803,392 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,071,148,032 bytes
Scraper detected USB disk size	3,246,391,296 bytes
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,071,803,392 bytes
Name given to extracted memory image	GX620_warm2.dd

Table 35: System 5 –Data collected for Experiment 3

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,071,803,392 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,071,148,032 bytes
Scraper detected USB disk size	3,246,391,296 bytes
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,071,803,392 bytes
Name given to extracted memory image	GX620_cold1.dd

Table 36: System 5 – Data collected for Experiment 4

Time required to remove power and reattach it	Approximately 1 second
Scraper total detected memory	1,071,803,392 bytes
Scraper detected memory ranges	Range 0: 655,360 bytes Range 1: 1,071,148,032 bytes
Scraper detected USB disk size	3,246,391,296 bytes
Time required to acquire memory	Approximately 2.5 minutes
Recovered memory image size	1,071,803,392 bytes
Name given to extracted memory image	GX620_cold2.dd

4.3.5.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys in memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 37: System 5 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	720 / 4,357 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	607 / 4,231 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 38: System 5 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	720 / 4,357 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	607 / 4,231 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 39: System 5 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	720 / 4,358 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	607 / 4,232 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 40: System 5 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	720 / 4,357 / 0 / 0
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	607 / 4,231 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

4.3.6 Experiments conducted against System 6

4.3.6.1 Background

Windows XP Professional 32-bit was installed onto this system, a Dell Latitude CPx laptop (see Table 66, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 3 and *TrueCrypt 6.0a* was then installed. The only partition on the system was a FAT32 partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal CD-ROM drive was set as the primary boot device and the secondary boot device was set to the internal hard disk drive. The BIOS did not support wake on power so in order to start the system after power is removed the power button must be pressed by the operator. The system does not have the ability to boot from a USB device therefore a customized FreeDOS CD was used for carrying out memory acquisition. Using the memory-dumping tool *Memdump* from the CD, it was possible to acquire memory images and then dump them to the FAT32 systems partition.

The laptop's battery was removed throughout the various experiments in order to ensure that any additional sources of electrical power that could potentially skew the results were absent.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

Two specific sets of memory acquisition experiments were conducted herein. The first set was conducted to verify if computer memory could be recovered without the flash-freezing of memory, a procedure similar to warm-booting a computer system. The second set carried out implemented the actual cold boot attack. Each set of experiments was done twice.

The acquired memory images, all stored in the root directory of the FAT32 system partition, were at end of all the experiments transferred to the Linux workstation for post-mortem analysis using the 16 GB USB flash drive (see Table 67, Annex A.1.2 for more details). There was no need to use the Linux workstation's *usbdump* memory extraction tool since each memory dump was in an accessible raw binary format.

Thus, in all four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack attempt 1
- Experiment 2: Warm boot attack attempt 2
- Experiment 3: Cold boot attack attempt 1
- Experiment 4: Cold boot attack attempt 2

The data from the computer system (data) and the analysis of the memory images (analyses) are examined further on in this section.

4.3.6.2 Experimental specifics

4.3.6.2.1 Warm boot specifics

The computer system was powered on and the administrative user was logged in to the system. Then the *TrueCrypt 6.0a* application was started and the cryptographic volume was mounted. Upon successfully mounting said volume the system's power was immediately removed and then quickly reconnected all while holding a finger on the system's power button so that the instant power was reattached the system would begin booting up.

With power reattached and the system having been started up, the CD-ROM drive was booted loading FreeDOS into memory. The FreeDOS command line environment was used for memory acquisition. Using the memory dump tool *Memdump* all memory dumps were written directly to the system partition's root directory that was recognized as drive C:\. Once the first memory dump was completed the system was powered off for approximately one minute to allow any residual data in memory to return to its ground state at which time the system was powered back on, the FreeDOS CD was removed, and the system booted back into Windows XP. Then the second instance of this experiment was carried out.

4.3.6.2.2 Cold boot specifics

The computer system was powered on and the administrative user was logged in to the system. Then the *TrueCrypt 6.0a* application was started and the cryptographic volume was mounted. Upon successfully mounting said volume, the system was turned over on its left side in order to gain access to its memory modules. The memory module cover was then removed thereby exposing the modules. Upon identifying the memory modules, they were all sufficiently flash-frozen. Once fully flash-frozen power was immediately removed and then quickly reconnected, all while holding a finger on the system's power button so that the instant power was reattached the system would begin booting up. The FreeDOS CD was inserted into the CD-ROM drive and following the system's POST CD was booted. From the FreeDOS command line, the *Memdump* tool was used to image the system's memory and store it to the system partition's root directory.

Upon the successful completion of the memory acquisition the system was powered off by the operator and left in that state for approximately fifteen minutes to allow the cooled memory to equalize with ambient temperature and allow any condensed moisture to evaporate. Then the system was powered back on, the FreeDOS CD removed, and the system was allowed to boot back into Windows XP to carry out the second instance of this experiment.

4.3.6.3 Collected data

4.3.6.3.1 Warm boot

Various information and notes taken by the authors while conducting this portion of the experiments are presented in the following tables:

Table 41: System 6 – Data collected for Experiment 1

Time required to remove power and reattach it	Just over 3 seconds
Operator selected memory dump size	262,064 KB (as determined by FreeDOS *mem.exe*)
Operator specified Memdump hexadecimal memory size	0x0FFEC000
Command used to dump memory	A:\> memdump.exe /DB:0,0x0FFEC000 /F:none /B:C:\memory1.dd
Time required to acquire memory	Less than 1 minute
Name of memory dump file	FreeDOS_warm1.dd
Size of memory dump (as per command *dir c:*)	262,064 KB in size

Table 42: System 6 – Data collected for Experiment 2

Time required to remove power and reattach it	Just over 2 seconds
Operator selected memory dump size	262,064 KB (as determined by FreeDOS *mem.exe*)
Operator specified Memdump hexadecimal memory size	0x0FFEC000
Command used to dump memory	A:\> memdump.exe /DB:0,0x0FFEC000 /F:none /B:C:\memory1.dd
Time required to acquire memory	Less than 1 minute
Name of memory dump file	FreeDOS_warm2.dd
Size of memory dump (as per command *dir c:*)	262,064 K in size

4.3.6.3.2 Cold boot

Various information and notes taken by the authors while conducting this portion of the experiments are presented in the following tables:

Table 43: System 6 – Data collected for Experiment 3

Time required to remove power and reattach it	Just over 2 seconds
Operator selected memory dump size	262,064 KB (as determined by FreeDOS *mem.exe*)
Operator specified Memdump hexadecimal memory size	0x0FFEC000
Command used to dump memory	A:\> memdump.exe /DB:0,0x0FFEC000 /F:none /B:C:\memory1.dd
Time required to acquire memory	Less than 1 minute
Name of memory dump file	FreeDOS_cold1.dd
Size of memory dump (as per command *dir c:*)	262,064 KB in size

Table 44: System 6 – Data collected for Experiment 4

Time required to remove power and reattach it	Just over 3 seconds
Operator selected memory dump size	262,064 KB (as determined by FreeDOS *mem.exe*)
Operator specified Memdump hexadecimal memory size	0x0FFEC000
Command used to dump memory	A:\> memdump.exe /DB:0,0x0FFEC000 /F:none /B:C:\memory1.dd
Time required to acquire memory	Less than 1 minute
Name of memory dump file	FreeDOS_cold2.dd
Size of memory dump (as per command *dir c:*)	262,064 KB in size

4.3.6.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys in memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 45: System 6 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,027,787 / 4,811,637 / 305,630 / 645
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	478,713 / 4,161,137 / 197,619 / 491
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	33 / 32 / 175 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	85 / 87 / 63 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	505 / 443 / 409 / 0
AES keys founds / Unique keys found	None / None

Table 46: System 6 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	1,167,383 / 6,282,115 / 414,001 / 954
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	474,069 / 5,461,762 / 269,121 / 670
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	13 / 12 / 169 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	104 / 104 / 54 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	482 / 446 / 429 / 0
AES keys founds	2 256-bit keys / 2 unique 256-bit keys

Table 47: System 6 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	911,609 / 5,035,785 / 253,541 / 999
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	321,045 / 4,328,076 / 117,188 / 520
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	13 / 13 / 286 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	118 / 118 / 112 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	472 / 467 / 593 / 0
AES keys founds	6 256-bit keys / 4 unique 256-bit keys

Table 48: System 6 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	674,986 / 3,960,945 / 231,813 / 909
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	270,245 / 3,381,150 / 109,890 / 478
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	11 / 11 / 237 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	118 / 116 / 91 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	403 / 386 / 536 / 0
AES keys founds	4 256-bit keys / 4 unique 256-bit keys

4.3.7 Other experiments

4.3.7.1 Objective

One final set of experiments were conducted by the authors. These experiments concentrated on System 2, the Dell Dimension E521 (see Table 62, Annex A.1.1 for more details). Since this was the only computer system other than the Dell CPx laptop (see Table 66, Annex A.1.1 for more details) to exhibit significant data remanence, the authors thought it prudent to verify whether something was unique about its memory modules or whether same-speed modules from other vendors would exhibit similar behaviour. Specifically, the authors wanted to test the memory modules of the Dell 3100 and Dell GX620 (see tables 64 and 65, Annex A.1.1, respectively) against the Dell E521. Since all of these memory modules run at the same speed of 533 MHz it would be straightforward to assess whether the memory used by the Dell E521 is unique in terms of its data remanence or whether there is something specifically unique about the system itself.

More specifically, the authors wished to assess if the data remanence that had already been observed in the Dell E521 using the 4x1024MB memory modules, both under warm and cold boot attack conditions, would occur given the use of altogether different memory modules while only conducting warm boot attack experiments. In only using the warm boot attack, it becomes possible to separate memory modules with a higher propensity for data remanence as opposed to those that may only exhibit data remanence when flash-frozen. It was further hoped that in performing warm boot experiments only it would be possible to determine if the underlying computer system, the Dell E521, plays any key role in determining the data remanence characteristics of a given set of memory modules.

Recall that both the Dell 3100 and Dell GX620 exhibited no data remanence. However, if their memory modules used on the Dell E521 and should significant data remanence occur then this is a strong indication that the memory modules are not directly responsible for data remanence. Instead, the authors contend that in fact another phenomenon is instead at play here, specifically motherboard residual capacitance.

The current memory modules used by the Dell E521 are 4x1024MB 240-pin Patriot Signature PC4200 DDR2 533 MHz memory modules. However, before the 4x1024MB memory modules were in use the system was using 2x512MB 240-pin Samsung PC4200 DDR2 533 MHz memory modules, which are going to be used in these experiments.

The same warm boot experiments were conducted against the Dell E521 with three different pairs of 512MB memory modules (the ones in the 3100, the one in the GX620 and the older ones from the E521).

4.3.7.2 Background

Two sets of memory experiments were conducted against each specific set of memory modules. The first pair of memory module experiments were conducted against the Micron 2x512MB memory modules from the Dell 3100. The second set of experiments was conducted against the Micron 2x512MB memory modules from the Dell GX620. The final set of experiments was

conducted against the Dell E521's older Samsung 2x512MB memory modules. Each specific experiment was run twice to rule out random results.

Specifically, Windows XP Professional 32-bit was installed onto this system, a Dell Dimension E521 desktop (see Table 62, Annex A.1.1 for more details). The operating system was then upgraded to Service Pack 3 and *TrueCrypt 6.0a* was installed. The only partition on the system disk was an NTFS partition that stored the Windows XP operating system and other various applications. This partition spanned the entire disk and was set as active.

From the BIOS, the internal hard disk drive was set as the primary boot device and the secondary boot device was set to the internal CD/DVD drive. The BIOS was configured to automatically power up the instant power was reattached in the event power had been interrupted. To boot from another device (e.g. USB flash drive) at boot time it sufficed to press the F12 key at the system's POST in order to access the system boot menu.

A previously created *TrueCrypt* volume *vol.tc* (without the use of a keyfile) was copied to the system partition prior to commencing the experimentation. The hash and encryption algorithms used were Whirlpool and AES, respectively. The encrypted volume was 10 MB in size and formatted as FAT16.

This computer system is USB-bootable thereby enabling USB-based memory acquisition. For the experiments conducted against this specific computer system, a 16 GB USB flash drive was used (see Table 68, Annex A.1.2 for more details). A 32-bit version of the *bios_memimage* software package *scraper* program was compiled and copied to sector 0 of the aforementioned flash drive.

Upon completion of each specific experiment, the USB flash drive was connected to a Linux workstation for post-mortem memory image extraction and analysis. Once a given memory image was successfully recovered the USB flash drive was zero-fill wiped to ensure there would be no potential contamination for future experiments. However, since the memory to be captured is approximately only 1 GB in size the 32-bit versions of the *scraper* and *usbdump* programs were used.

Therefore, four experiments were conducted against this system as follows:

- Experiment 1: Warm boot attack using Dell 3100 Micron 2x512MB memory
- Experiment 2: Warm boot attack using Dell 3100 Micron 2x512MB memory
- Experiment 3: Warm boot attack using Dell GX620 Micron 2x512MB memory
- Experiment 4: Warm boot attack using Dell GX620 Micron 2x512MB memory
- Experiment 5: Warm boot attack using Dell E521 Samsung 2x512MB memory
- Experiment 6: Warm boot attack using Dell E521 Samsung 2x512MB memory

The data from the computer system (data) and the analysis of the memory images (analyses) are examined further on in this section.

4.3.7.3 Data

Various information and notes taken by the authors while conducting this portion of the experiments are presented in the following tables:

Table 49: System 2 – Data collected for Experiment 1

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory1_3100.dd

Table 50: System 2 – Data collected for Experiment 2

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory2_3100.dd

Table 51: System 2 – Data collected for Experiment 3

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory1_GX620.dd

Table 52: System 2 – Data collected for Experiment 4

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory2_GX620.dd

Table 53: System 2 – Data collected for Experiment 5

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory1_Samsung.dd

Table 54: System 2 – Data collected for Experiment 6

Time required to remove power and reattach it	Less than 1 second
Scraper total detected memory (in bytes)	937,947,136
Scraper detected memory ranges (in bytes)	Range 0: 651,264 Range 1: 937,295,872
Scraper detected USB disk size (in bytes)	3,246,390,784
Time required to acquire memory	Approximately 2 minutes
Recovered memory image size (in bytes)	937,947,136
Name given to extracted memory image	Memory2_Samsung.dd

4.3.7.4 Analyses

Various analyses were conducted against the four acquired memory dumps. The first of these analyses was determining the total number of strings contained in the various memory images as determined by the UNIX *strings* command. Then the number of unique strings for each memory image was determined using the same command. Finally, since each memory image was based

on a running instance of the Windows XP operating system with *TrueCrypt 6.0a* running it should be possible to extract various Windows and application-related strings and potentially AES-based encryption keys from memory. Of course, this is assuming that the memory contents have not degraded much (or at all). Encryption key extraction was carried out using the tool *aeskeyfind*. The data collected from these analyses are presented in the following tables below:

Table 55: System 2 – Analyses of Experiment 1

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	707,022 / 21,717,801 / 208,914 / 7
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	459,745 / 2,704,676 / 87,058 / 7
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 56: System 2 – Analyses of Experiment 2

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	551,604 / 10,933,892 / 136,782 / 11
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	419,832 / 3,001,454 / 60,621 / 11
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 57: System 2 – Analyses of Experiment 3

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	5,316,041 / 38,512,659 / 2,142,697 / 87
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	3,090,884 / 13,003,470 / 676,740 / 68
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 58: System 2 – Analyses of Experiment 4

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	463,235 / 3,457,139 / 76,363 / 9
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	97,953 / 825,624 / 58,653 / 9
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	0 / 0 / 0 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	0 / 0 / 0 / 0
AES keys founds / Unique keys found	None / None

Table 59: System 2 – Analyses of Experiment 5

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	7,280,219 / 28,224,913 / 785,724 / 6,799
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	3,078,387 / 11,580,686 / 306,910 / 2,786
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	55 / 54 / 861 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	367 / 367 / 309 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	778 / 778 / 601 / 0
AES keys founds	5 256-bit keys / 4 unique 256-bit keys

Table 60: System 2 – Analyses of Experiment 6

Analysis	Results
Number of 7-bit / 8-bit / 16-bit / 32-bit strings	7,339,597 / 28,710,290 / 917,826 / 6,568
Number of unique 7-bit / 8-bit / 16-bit / 32-bit strings	3,252,002 / 11,973,366 / 401,014 / 3,042
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Windows XP"	54 / 53 / 817 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "Service Pack"	347 / 347 / 344 / 0
Number of 7-bit / 8-bit / 16-bit / 32-bit strings containing "TrueCrypt"	764 / 764 / 587 / 0
AES keys founds	5 256-bit keys / 4 unique 256-bit keys

4.3.8 End of experiments

All the experiments carried out herein have at this point been successfully completed and conducted. Surprisingly, all experiments succeeded as intended such that no aberrant system behaviour was noticed such as the inability for a system to conduct a POST or boot from a *scraper*-enabled USB flash drive or hard disk drive. However, it was surprising that most of the computer systems experimented upon herein exhibited little to no memory remanence, at least according to the various tests conducted herein against the *scraper*-dumped memory.

References

[1] Halderman, J. Alex, Schoen, Seth D., Heninger, Nadia, et al. Lest We Remember: Cold Boot Attacks on Encryption Keys. Research paper. February 2008. Published in Proceedings 2008 USENIX Security Symposium. Princeton University. http://citp.princeton.edu/pub/coldboot.pdf

[2] Skorobogatov, Sergei. Low temperature data remanence in static RAM. June 2002. Technical report, UCAM-CL-TR-536 / ISSN 1476-2986. Report No: 536. University of Cambridge. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.pdf

[3] Skorobogatov, Sergei. Data Remanence in Flash Memory Devices. Research paper. Unknown date. University of Cambridge. http://www.cl.cam.ac.uk/~sps32/DataRem_CHES2005.pdf

[4] Briody, Jason T. Data to RAM: "I Will Survive." Research paper. October 2008. http://c3di.champlain.edu/TR/New_Developments_in_Volatile_Memory_Forensics.pdf

[5] Han, Ellick M, Carlyle, Jeffrey C., David, Francis M., et al. BootJacker: Compromising Computers using Forced Restarts. Research paper. October 2008. University of Illinois at Urbana-Champaign. http://srgsec.cs.illinois.edu/bootjacker.pdf

[6] Wikipedia. Data remanence. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Data_remanence

[7] Wikipedia. Cold boot attack. Online encyclopaedia. June 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Cold_boot_attack

[8] Wikipedia. Dynamic random access memory. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Dynamic_random_access_memory

[9] Wikipedia. Static random access memory. Online encyclopaedia. January 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Static_random_access_memory

[10] Wikipedia. Regenerative capacitor memory. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Regenerative_capacitor_memory

[11] Wikipedia. Capacitance. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Capacitance

[12] Guttmann, Peter. Secure Deletion of Data from Magnetic and Solid-State Memory. Research paper. July 1996. Published in Sixth USENIX Security Symposium Proceedings. University of Auckland. http://www.cs.cornell.edu/people/clarkson/secdg/papers.sp06/secure_deletion.pdf

[13] Karabatsos, Chris. Apparatus and method for improving computer memory speed and capacity. Online filed patent. September 1999. U.S. Patent No: 5,953,215. Free Patents Online.com. http://www.freepatentsonline.com/5953215.html

[14] Karabatsos, Chris. Apparatus and method for terminating a computer memory bus. Online filed patent. July 2001. U.S. Patent No: 6,266,252. Free Patents Online.com. http://www.freepatentsonline.com/6266252.html

[15] Washburn, Robert D. and McClanahan, Robert F. Control signal interface circuit for computer memory modules. Online filed patent. August 2007. U.S. Patent No: 7,260,000. Patent Storm.us. http://www.patentstorm.us/patents/7260000/fulltext.html

[16] Wyns, Philippe and Anderson, Richard L. Low-Temperature Operation of Silicon Dynamic Random-Access Memories. Research paper. August 1989. Published in IEEE Transactions on Electron Devices, Vol. 36. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00030954

[17] Wikipedia. Physical Address Extension. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Physical_Address_Extension

[18] Wikipedia. Gas duster. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Gas_duster

[19] Carbone, Richard. Forensic software memory acquisition of various operating systems. Research paper (draft). 2010. Defence Research Development Canada.

[20] Wikipedia. Cryogenics. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Cryogenics

[21] Wikipedia. JEDEC. Online encyclopaedia. May 2010. Wikimedia Foundation. http://en.wikihttp://en.wikipedia.org/wiki/JEDEC

[22] Martin, Antonio. FireWire memory Dump of a Windows XP Computer: A Forensic Approach. Research paper. 2007. http://www.friendsglobal.com/papers/FireWire%20Memory%20Dump%20of%20Windows%20XP.pdf

[23] Dornseif, Maximillian. 0wned by an iPod. Electronic presentation. Presented at PacSec2004. University of Technology Aachen. http://md.hudora.de/presentations/firewire/PacSec2004.pdf

[24] Boileau, Adam. Hit by a bus: Physical Access Attacks with Firewire. Electronic presentation. Presented at Ruxcon 2006. Security-assessment.com. http://www.storm.net.nz/static/files/ab_firewire_rux2k6-final.pdf

[25] Gray, Patrick. Security conference to debut Windows firewire crack. Online article. September 2006. The Sydney Morning Herald. http://www.smh.com.au/news/security/security-conference-to-debut-windows-firewire-crack/2006/09/18/1158431640614.html#

[26] Schuster, Andreas. Acquisition (5): FireWire. Online information resource. February 2008. computer.forensikblog.de. http://computer.forensikblog.de/en/2008/02/acquisition_5_firewire.html

[27] Carrier, Brian D., and Grand, Joe. A Hardware-Based Memory Acquisition Procedure for Digital Investigations. Research paper. Digital-evidence.org and Grand Idea Studio Inc. http://www.digital-evidence.org/papers/tribble-preprint.pdf

[28] Rutkowska, Janet. Beyond the CPU: Defeating Hardware Based RAM Acquisition (part I: AMD case). Electronic presentation. February 2007. Presented at Blackhat 2007. COSEINC Advanced Malware Labs. http://i.i.com.com/cnwk.1d/i/z/200701/bh-dc-07-Rutkowska-ppt.pdf

[29] Petterson, Torbjörn. Cryptographic key recovery from Linux memory dumps. Research paper. Aprtil 2007. Torbjörn Petterson. http://events.ccc.de/camp/2007/Fahrplan/attachments/1300-Cryptokey_forensics_A.pdf

[30] Walters, Aaron, and Petroni Jr., Nick L. Volatools: Integrating Volatile Memory forensics into the Digital Investigation Process. Research paper. Unknown date. Komoku Inc. http://www.blackhat.com/presentations/bh-dc-07/Walters/Paper/bh-dc-07-Walters-WP.pdf

[31] Kaplan, Brian. RAM is Key: Extracting Disk Encryption Keys From Volatile Memory. Thesis report. May 2007. Carnegie Mellon University. http://www.contrib.andrew.cmu.edu/~bfkaplan/KaplanRAMisKeyThesis.pdf

[32] Schouwenaar, Michael. Analysing the Cold Boot Attack. Research paper. February 2009. http://security1.win.tue.nl/~bskoric/seminar/papers1/Schouwenaar_coldboot.pdf

[33] Davidoff, Sherri. Cleartext Passwords in Linux Memory. Research paper. July 2008. Massachusets Institute of Technology. http://philosecurity.org/pubs/davidoff-clearmem-linux.pdf

[34] Casey, Eoghan. Practical Approaches to Recovering Encrypted Digital Evidence. Research paper. Fall 2002. International Journal of Digital Evidence, Volume 1, Issue 3. http://www.utica.edu/academic/institutes/ecii/publications/articles/A04AF2FB-BD97-C28C-7F9F4349043FD3A9.pdf

[35] Klein, Tobias. All your private keys are belong to us: Extracting RSA private keys and certificates out of the process memory. Research paper. May 2006. Trapkit.de. http://www.trapkit.de/research/sslkeyfinder/keyfinder_v1.0_20060205.pdf

[36] Garfinkel, Simson, and Shelat, Abhi. Remembrance of Data Passed: A Study of Disk Sanitization Practices. Research paper. 2003. IEEE Security and Privacy, No: 1540-7993/03$17.00. Institute_of_Electrical_and_Electronics_Engineers. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1176992

[37] National Institute for Occupational Safety and Health. 1,1,1,2-Tetrafluoroethane. International Chemical Safety Card. 2001. National Institute for Occupational Safety and Health. http://www.setonresourcecenter.com/msdshazcom/htdocs/IPCSNENG/neng1281.htm

[38] Air Liquide. Safety Data Sheet: Difluoroethane (R152a). Chemical safety data sheet. July 2005. Air Liquide. http://www.setonresourcecenter.com/msdshazcom/htdocs/MSDS/A/Air_Liquide/045_AL_EN.pdf#search=%22difluoroethane%22

[39] Air Liquide. Safety Data Sheet: Trifluoroethane (R143a). Chemical safety data sheet. July 2005. Air Liquide. http://www.msdshazcom.com/MSDS/A/Air_Liquide/118_AL_EN.pdf

[40] Wikipedia. Liquid nitrogen. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Liquid_nitrogen

[41] Wasson, Scott. Intel's Core i7 processors: Nehalem arrives with a splash. Online article. November 2008. The Tech Report.com. http://techreport.com/articles.x/15818

[42] Wikipedia. Electrical resistance. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Electrical_resistance

[43] Wikipedia. Electrical conductor. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Electrical_conductor

[44] Wikipedia. Capacitor. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Capacitor

[45] Pabel, Jürgen. Frozen Cache. Online information resource. January 2009. Blogger.com. http://frozencache.blogspot.com/

[46] Leyden, John. Security boffins attempt to freeze out cold boot crypto attack: Cache from chaos. Online article. January 2009. The Register .co.uk. http://www.theregister.co.uk/2009/01/19/cold_boot_countermeasures/

[47] Wikipedia. Trusted Platform Module. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Trusted_Platform_Module

[48] Wikipedia. Truecrypt. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Truecrypt

[49] Wikipedia. Hibernation (computing). Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Hibernate_(OS_feature)

[50] Wikipedia. Advanced Configuration and Power Interface. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface

[51] Maartmann-Moe, Carsten, Thorkildsen, Steffen E., and Arnes, Andre. The persistence of memory: Forensic identification and extraction of cryptographic keys. Research paper. 2009. Norwegian University of Science and Technology and the Gjøvik University College. http://www.dfrws.org/2009/proceedings/p132-moe.pdf

[52] Truecrypt team. Hibernation File. Technical documentation. Unknown date. Truecrypt.org. http://www.truecrypt.org/docs/?s=hibernation-file

[53] Unknown author. Low-Temperature Data Retention in Nonvolatile SRAM. Application Note. Note No: 4289. September 2008. Maxim Integrated Products. http://pdfserv.maxim-ic.com/en/an/AN4289.pdf

[54] Wikipedia. CONFIG.SYS. Online encyclopaedia. June 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/CONFIG.SYS

[55] Wikipedia. AUTOEXEC.BAT. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/AUTOEXEC.BAT

[56] Wikipedia. Comparison of file systems. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Comparison_of_file_systems

[57] Wikipedia. File Allocation Table. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/File_Allocation_Table

[58] Carbone, Richard. Operating system hardware reconfiguration: A case study for Linux. Technical memorandum. January 2007. Defence Research Development Canada. http://cradpdf.drdc.gc.ca/PDFS/unc56/p527008.pdf

[59] Wikipedia. Fermi level. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Fermi_level

[60] Wikipedia. Semiconductor. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Semiconductors

This page intentionally left blank.

A.1 Computer systems and other hardware used in cold boot memory acquisition experimentation

A.1.1 Computer system details

This appendix provides the details of the various computer system experimented upon by the authors. Tables 61 through 66 provide a highly detailed hardware list of said computer systems.

Table 61: System 1 – Dell Precision 690 Workstation

Computer model	Dell Precision 690 Workstation (64-bit capable)
Service Tag	XXXXNB1
Desktop/Laptop	Desktop (specifically workstation)
USB Bootable	Yes
BIOS	Precision Workstation 690 Revision A01
Processors	Dual Xeon 3.20 GHz w/Hyper-Threading (8 logical processors)
Physical RAM	16.00 GB 533 MHz ECC RAM; 2x4GB; 8x1GB
CD/DVD Drive	1) Hitachi LG CD-RW 48X 2) Hitachi LG DVD+/-RW 16X
Hard drives	1) Hitachi 500 GB SATA 7,200 RPM (system disk) 2) Western 300 GB Digital VelociRaptor SATA 10,000 RPM
Sound card	Sigmatel High Definition audio card (motherboard integrated)
Network cards	1) Broadcom NetXtreme 57XX Ethernet (motherboard integrated) 2) 1394 Net Adapter Ethernet (PCI Express Card)
Graphics adapter	NVidia NVS 285 PCI Express Graphics Card
Monitors	BenQ 19" LCD and Dell 19" LCD
Floppy	1.44 MB floppy drive
USB Ports	8 USB ports in system
Keyboard	Dell USB 101 US English keyboard
Mouse	Dell USB 3-button optical mouse
FireWire Ports	2 FireWire ports
Swap	Set to 16 GB (on non-system disk 300 GB Western Digital)
Operating System	Windows XP Professional 64-bit Service Pack 2
Virtualization	VMware Workstation 7.0.0 installed but no virtual machine currently

Software	running
Connected Peripherals	None

Table 62: System 2 – Dell Dimension E521

Computer model	Dell Dimension E521 (64-bit capable)
Service Tag	XXXXBC1
Desktop/Laptop	Desktop
USB Bootable	Yes
BIOS	Dell BIOS Revision 1.1.11
Processors	AMD Athlon X2 3800+ 2.0 GHz (2 logical processors)
Physical RAM	4.00 GB 533 MHz non-ECC RAM; 4x1GB
CD/DVD Drive	1 DVD+/-RW Sony AW-Q160S 16X
Hard drives	Seagate 500 GB ST3500630AS SATA 7,200 RPM (system disk)
Sound card	Sigmatel High Definition audio card (motherboard integrated)
Network cards	Broadcom BCM4401-BO 440X Ethernet (motherboard integrated)
Graphics adapter	NVidia C51 GeForce 6150 LE (motherboard integrated/PCI Express)
Monitors	Dell 19" LCD
Floppy	None
USB Ports	5 USB ports in system
Keyboard	Dell USB 101 US English keyboard
Mouse	Dell USB 3-button optical mouse
FireWire Ports	None
Operating System	Windows XP Professional Service Pack 3
Swap	Set to 2,048 MB (on system disk)
Virtualization Software	None
Connected Peripherals	USB connected HP OfficeJet 4125xi all-in-one printer

Table 63: System 3 – Dell Latitude E6500

Computer model	Dell Latitude E6500 (64-bit capable)
Service Tag	XXXX3J1
Desktop/Laptop	Laptop
USB Bootable	Yes
BIOS	Dell Latitude E6500 Series BIOS Revision A11
Processors	Intel Centrino 2 2.80 GHz (2 logical processors)
Physical RAM	4.00 GB 800 MHz non-ECC RAM; 2x2GB
CD/DVD Drive	Toshiba/Samsung DVD+/-RW 8X
Hard drives	None
Sound card	Intel 82801I HD audio card (motherboard integrated)
Network cards	1) Intel 82567LM Gigabit Ethernet (motherboard integrated) 2) Intel PRO/Wireless 5300 AGN (disabled) (motherboard integrated)
Graphics adapter	NVidia Quadro NVS 160M 256 MB (motherboard integrated/PCI Express)
Monitors	Samsung 15.4" integrated WUXGA+ LCD
Floppy	None
USB Ports	3 USB ports in system
E-SATA Ports	3 E-SATA port in system
Keyboard	Integrated keyboard
Mouse	Integrated touch mouse pad
FireWire Ports	1 FireWire port in system
Other Integrated Peripherals	1) 1 smartcard reader (integrated) 2) 1 integrated fax/voice modem
Operating System	Ubuntu 9.04 64-bit Live CD kernel 2.6.28-11 generic #42-Ubuntu SMP x86_64
Swap	None
Virtualization Software	None
Connected Peripherals	None

Table 64: System 4 – Dell Dimension 3100

Computer model	Dell Dimension 3100 (64-bit capable)
Service Tag	XXXXQB1
Desktop/Laptop	Desktop
USB Bootable	Yes
BIOS	Dell DV051 Series Revision A04
Processors	Intel Pentium 4 w/Hyper-Threading 3.06 GHz
Physical RAM	1,024 MB 533 MHz non-ECC RAM; 2x512 MB
CD/DVD Drive	1) Philips DVD+/-RW 16X 2) Hitachi/LG DVD 16X
Hard drives	Samsung 160 GB HD160JJ SATA 7,200 RPM (system disk)
Sound card	Sigmatel High Definition audio card (motherboard integrated)
Network cards	Intel Pro 100/VE Ethernet (motherboard integrated)
Graphics adapter	Intel 82195G PCI Express (motherboard integrated)
Monitors	Philips 19" 190B LCD
Floppy	TEAC Integrated USB Flash Card Multimedia Reader
USB Ports	6 USB ports in system
Keyboard	Dell USB 101 US English keyboard
Mouse	Dell USB 3-button optical mouse
FireWire Ports	None
Operating System	Windows XP Professional Service Pack 3
Swap	Set to 2,048 MB (on system disk)
Virtualization Software	None
Connected Peripherals	USB connected Primera Bravo II Disc Publisher (50 disc capacity)

Table 65: System 5 – Dell OptiPlex GX620

Computer model	Dell OptiPlex GX620 (64-bit capable)
Service Tag	XXXX7B1
Desktop/Laptop	Desktop
USB Bootable	Yes
BIOS	Dell OptiPlex GX620 Revision A07
Processors	Intel Pentium 4 w/Hyper-Threading 3.59 GHz
Physical RAM	1,024 MB 533 MHz non-ECC RAM; 2x512 MB
CD/DVD Drive	Sony DVD/CD+-RW CRX310EE
Hard drives	Seagate 160 GB ST31608AS SATA 7,200 RPM (system disk)
Sound card	Analog Devices ADI 198x HD Audio (motherboard integrated)
Network cards	Broadcom NetXtreme Ethernet 75XX (motherboard integrated)
Graphics adapter	Intel 945G PCI Express
Monitors	Dell 19" E196FPF LCD
Floppy	1.44 MB floppy disk drive
USB Ports	8 USB ports in system
Keyboard	Dell USB 101 US English keyboard
Mouse	Dell USB 3-button optical mouse
FireWire Ports	None
Operating System	Windows XP Professional Service Pack 3
Swap	Set to 2,048 MB (on system disk)
Virtualization Software	None
Connected Peripherals	None

Table 66: System 6 – Dell Latitude CPx

Computer model	Dell Latitude CPx (32-bit system only)
Service Tag	XXX-YYY-63
Desktop/Laptop	Laptop
USB Bootable	No
BIOS	Dell Latitude CPx H500GT Revision A07
Processors	Intel Pentium 3 500 MHz
Physical RAM	256 MB 100 MHz non-ECC RAM; 2x128 MB
CD/DVD Drive	TEAC CD-224E CD-ROM 24X
Hard drives	Seagate 40 GB ST94811A IDE 5,400 RPM (system disk)
Sound card	ESS Maestro 2E (motherboard integrated)
Network cards	PCMCIA Xircom CE3-10/100 Ethernet (motherboard integrated)
Graphics adapter	ATI Rage Mobility P/M AGP 2x (motherboard integrated)
Monitors	Integrated 14.1" WXGA LCD
Floppy	None
USB Ports	1 USB ports in system
Keyboard	Integrated keyboard
Mouse	Dell USB 3-button optical mouse
FireWire Ports	None
Operating System	Windows XP Professional Service Pack 3
Swap	Set to 2,048 MB (on system disk)
Virtualization Software	None
Connected Peripherals	None

A.1.2 USB memory acquisition devices

Tables 67 and 68 provide the details for the two various forms of USB-based media used throughout the experiments herein.

Table 67: 16 GB USB flash drive (disk size specifics based on information from Linux fdisk)

Manufacturer (I/O case)	Kingston
Model	DataTraveller 16GB
Size (GB = 1024^3 bytes)	15.02
Size (MB = 1024^2 bytes)	15,382.55
Heads	255
Sectors/Track	63
Cylinders	1,961

Table 68: 250 GB USB hard disk drive (disk size specifics based on information from Linux fdisk)

Manufacturer (I/O case)	Vantec
Model	NexStar 3 USB/eSATA
Size (GB = 1024^3 bytes)	232.88
Size (MB = 1024^2 bytes)	238,472.69 MB
Heads	255
Sectors/Track	63
Cylinders	30,401

This page intentionally left blank.

Bibliography

3saul, jamie, and fatrabbit. dd issues on Ubuntu 6.10. Online information resource. February 2007. http://www.forensicfocus.com/index.php?name=Forums&file=viewtopic&t=1453

Dornier, Pascal. Adaptive DRAM timing set according to sum of capacitance valves retrieved from tables based on memory bank size. United States patent. U.S. Patent No: 5,505,877. April 1996. Elonex.co.uk. http://www.elonex.co.uk/imgpdf/05504877.pdf

Guttmann, Peter. Data Remanence in Semiconductor Devices. Research paper. Unknown date. IBM. http://www.cypherpunks.to/~peter/usenix01.pdf

Heald, Ray. How Cosmic Rays Cause Computer Downtime. Presentation. March 2005. Sun Microsystems. http://www.ewh.ieee.org/r6/scv/rl/articles/ser-050323-talk-ref.pdf

JEDEC. Member List. Information online resource – list of member organizations. 2010. JEDEC.org. http://www.jedec.org/service_members/New_Members/memberco.cfm

Juang, Philo, Diodato, Phil, et al. Implementing Decay Techniques using Quasi-Static Memory Cells. Research paper. Unknown date. Princeton University, Agere Systems, and University of Virginia. http://www.cs.virginia.edu/papers/caletters02.pdf

O'Neill, Ian. Intel to Protect Microchips from Cosmic Rays. Online article. April 2008. Universe Today.com. http://www.universetoday.com/2008/04/08/intel-to-protect-microchips-from-cosmic-rays/

Osborn, Neal A. Predictive temperature compensation for memory devices systems and method. Online filed patent – Description. U.S. Patent No: 6,557,072. April 2003. Patent Storm.us. http://www.patentstorm.us/patents/6557072/description.html

PC Tech Guide. L1 cache. Online information resource. Unknown date. PC Tech Guide.com. http://www.pctechguide.com/14Memory_L1_cache.htm

Simonite, Tom. Should every computer chip have a cosmic ray detector? Online article. March 2008. New Scientist Magazine. http://www.newscientist.com/blog/technology/2008/03/do-we-need-cosmic-ray-alerts-for.html

Smith, Sean W. and Weingart, Steve. Building a High-Performance, Programmable Secure Coprocessor. Research paper. October 1998. IBM. http://www.research.ibm.com/secure_systems_department/projects/scop/papers/arch.pdf

Sun Microsystems. Sun Enterprise 6500/5500/4500 Systems Reference Manual. Technical manual. August 2001. http://dlc.sun.com/pdf/805-2632-11/805-2632-11.pdf

Sygus. La mémoire sous Linux : analyse de /dev/mem. Online information resource. June 2009. Sygus.net. http://www.sygus.net/dotclear/index.php?post/2009/06/03/La-m%C3%A9moire-sous-Linux-%3A-analyse-de-/dev/mem

Unknown author. Attack on computer memory reveals vulnerability of widely-used security systems. Online information resource. February 2008. Physorg.com. http://www.physorg.com/news122820185.html

Unknown author. Understanding CCD Read Noise. Online information resource. Unknown date. BintelShop.com.au. https://www.bintelshop.com.au/HTML/ccd_noise_measure.html

Walle, Bernhard. Turn CONFIG_STRICT_DEVMEM in sysctl dev.mem.restricted. Online information resource. November 2008. LWN.net. http://lwn.net/Articles/307396

Washburn, Robert D. and McClanahan, Robert F. Control signal interface circuit for computer memory module. Online filed patent – Description. U.S. Patent No: 7,260,000. August 2007. FreePatentsOnline.com. http://www.freepatentsonline.com/7260000.html

Wickman, Cliff, and Walle, Bernhard. CONFIG_STRICT_DEVMEM. Online mail information resource archive. November 2008. Linux Archive.org. http://www.linux-archive.org/crash-utility/192275-config_strict_devmem.html

Wikipedia. Computer forensics. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Computer_forensics

Wikipedia. DDR SDRAM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/DDR_SDRAM

Wikipedia. DDR2 SDRAM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/DDR2_SDRAM

Wikipedia. DDR3 SDRAM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/DDR3_SDRAM

Wikipedia. DDR4 SDRAM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/DDR4_SDRAM

Wikipedia. DIMM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/DIMM

Wikipedia. Dynamic random access memory. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Dynamic_random_access_memory#Extended_Data_Out_.28EDO.29_DRAM

Wikipedia. JEDEC memory standards. Online encyclopaedia. August 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/JEDEC_memory_standards

Wikipedia. Random-access memory. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Random_access_memory

Wikipedia. SIMM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/SIMM

Wikipedia. SIMM. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/SIMM

Wikipedia. Stationary state. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/Stationary_state

Wikipedia. Synchronous dynamic random access memory. Online encyclopaedia. September 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/SDRAM

Wikipedia. X86-64. Online encyclopaedia. October 2010. Wikimedia Foundation. http://en.wikipedia.org/wiki/X86-64

List of symbols/abbreviations/acronyms/initialisms

ACPI	Advanced Configuration and Power Interface
BIOS	Basic Input/Output System
BSD	Berkeley Software Distribution
CCD	Charged Coupled Device
CD	Compact Disc
CD-ROM	Compact Disc-Read Only Memory
CISC	Complex Instruction Set Computer
CPU	Central Processing Unit
DDR	Double Data Rate
DEC	Digital Equipment Corporation
DIMM	Dual Inline Memory Module
DMA	Direct Memory Access
DND	Department of National Defence
DOS	Disk Operating System
DRAM	Dynamic RAM
DRDC	Defence Research & Development Canada
DVD	Digital Video Disc or Digital Versatile Disc
e-SATA	External Serial Advanced Technology Attachment
EB	Exabyte
ECC	Error Correcting Code
EEPROM	Electrically Erasable Programmable Read-Only Memory
FAT16	File Allocation Table 16-bit
FAT32	File Allocation Table 32-bit
FreeDOS	Free Disk Operating System
GB	Gigabyte
GHz	Gigahertz
GRUB	GRand Unified Bootloader
HP	Hewlett Packard
HP PA-RISC	Hewlett Packard Precision Architecture Reduced Instruction Set Computer
IBM	International Business Machine

IDE	Integrated Drive Electronics
IEEE	Institute of Electrical and Electronics Engineers
ISP	Internet Service Provider
JEDEC	Joint Electron Device Engineering Council
KB	Kilobyte
LCD	Liquid Crystal Display
MHz	Megahertz
MIPS	Microprocessor without Interlocked Pipeline Stages
MS	Microsoft
MS-DOS	Microsoft-Disk Operating System
NAND	(logical) "Not AND"
NTFS	New Technology File System
NVRAM	Non-Volatile Random Access Memory
PAE	Physical Address Extension
PC	Personal Computer
PCI	Peripheral Component Interconnect
POST	Power On Self-Test
PROM	Programmable Read-Only Memory
PXE	Preboot eXecution Environment
R&D	Research & Development
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RPM	Revolutions per Minute
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SDRAM	Synchronous Dynamic Random Access Memory
SGI	Silicon Graphics International
SI	Système international d'unités (international system of units)
SIMM	Single Inline Memory Module
SP[1-6]	Service Pack [1 to 6]
SRAM	Static Random Access Memory

SSH	Secure SHell
TB	Terabyte
TFTP	Trivial File Transfer Protocol
TPM	Trusted Platform Module
USB	Universal Serial Bus
VAX	Virtual Address eXtension
VNC	Virtual Network Computing
Windows ME	Windows Millennium
Windows NT	Windows New Technology
Windows XP	Windows eXPerience

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED
NON-CONTROLLED GOODS
DMC A
REVIEW: GCEC JUNE 2010

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

An in-depth analysis of the cold boot attack: Can it be used for sound forensic memory acquisition?

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Carbone, R; Bean, C., and Martin S.

5. DATE OF PUBLICATION (Month and year of publication of document.)

January 2011

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

114

6b. NO. OF REFS (Total cited in document.)

60

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

31XF20 « MOU RCMP "Live Forensics" »

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Valcartier TM 2010-296

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) The purpose of this technical memorandum is to examine the technical characteristics behind the cold boot attack technique and to understand when and how this technique should be applied to the field of computer forensic investigations. Upon thorough examination of the technique, the authors highlight its advantages, drawbacks, applicability and appropriateness for use in the acquisition of computer memory contents. The original cold boot attack paper, as conducted by a team of students and researchers in 2008, demonstrated the usefulness of computer memory remanence and how this phenomenon could be used to defeat popular disk encryptions tools and other data hiding techniques necessary for the safe storage of secret data and information. However, the technique is not a panacea and has many drawbacks dictated by the laws of physics, which cannot be overcome by the technique. The authors believe that a thorough understanding of this phenomenon will empower computer forensic investigators to take advantage of it when appropriate but also aim at dispelling various distortions surrounding it.

(U) Le but de cet article est d'examiner les caractéristiques techniques de l'attaque par démarrage à froid et de comprendre quand et comment cette technique devrait être appliquée au domaine des enquêtes en informatique judiciaire. À la suite de l'examen approfondi de la technique, les auteurs font ressortir ses avantages et inconvénients, son applicabilité et sa pertinence pour l'acquisition du contenu de la mémoire d'ordinateur. L'article original sur l'attaque par démarrage à froid, telle que menée par une équipe d'étudiants et de chercheurs en 2008, a démontré l'utilité de la rémanence de la mémoire d'ordinateur et comment ce phénomène peut être utilisé pour percer les outils populaires de chiffrement de disque et autres techniques de dissimulation de données nécessaires au stockage sécurisé de données et d'information secrètes. Cependant, cette technique ne constitue pas une panacée et comporte plusieurs inconvénients découlant des lois de la physique que la technique ne peut contrecarrer. Les auteurs croient qu'une compréhension approfondie de ce phénomène habilitera les enquêteurs en informatique judiciaire à en tirer profit lorsque requis mais dissipera également certaines des idées fausses qui l'entourent.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Cold boot attack; Cold ghosting attack; Computer forensics; Flash freeze; Gaseous refrigerant; Hardware memory acquisition; Iceman attack; Inverted spray can; Liquid refrigerant; Memory acquisition; Platform reset attack; Propellant; Refrigerant; Software memory acquisition