Pocket SDV with SDGuardian: A Secure & Forensically Safe Portable Execution Environment

Peter Hannay[1]
Peter James[2]
Secure Systems Limited
Osborne Park
Western Australia.
pjames@securesystems.com.au

Abstract

Storage of sensitive and/or business critical data on portable USB attachable mass storage devices is a common practice. The ability to transport large volumes of data from the standard place of work and then access and process the data on an available PC at a different location provides both convenience and flexibility. However, use of such USB attachable mass storage devices presents two major security risks; the risk of loss of the portable storage device during transport and the risk of data remnants residing on a PC after accessing the data from the USB storage device. The latter risk is due to the way Windows and third party applications store temporary information on the host PC's hard disk. Even if every effort is made to delete temporary information it may be possible to recover this information by using forensic data recovery techniques such as header analysis and magnetic force microscopy.

The Pocket SDV with SDGuardian provides an elegant solution to the aforementioned security risks. The Pocket SDV is a commercially available USB attachable secure hard disk drive. Features of the Pocket SDV include hardware based encryption, strong authentication, differentiated access rights and cryptographically separate partitioning capabilities. Only a user with the correct authentication credentials can gain access to data stored on the Pocket SDV, thus providing assurance if the Pocket SDV is lost. SDGuardian is a proof of concept toolkit that minimises the remnants left on a PC if it is used to process data stored on a Pocket SDV. Forensic examination of the PC, following processing of data held on a Pocket SDV with SDGuardian, should not reveal any remnants of protected data. In this paper an overview of the Pocket SDV is given and its functionality is enumerated. The motivation for SDGuardian is outlined before discussing the design, capabilities and limitations of the Pocket SDV with SDGuardian.

KEYWORDS

Secure Portable Storage, Forensically Safe Portable Execution Environment, Digital Forensics.

INTRODUCTION

The Pocket SDV is a secure portable USB attachable mass storage device. The Pocket SDV enforces correct user authentication before data on the integral hard disk drive (HDD) may be accessed. Once the user has been correctly authenticated, the SDV allows access to the partitions (drives/volumes) on the Pocket SDV integral HDD. The Pocket SDV provides cryptographically enforced access to data contained on the integral HDD according to a previously configured data access profile for each user. The Pocket SDV operates independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from it. If the Pocket SDV is lost or stolen its owner can be assured that no one can gain access to the data due to strong authentication, nor use digital forensic tools to gain access to the data due to strong encryption.

When a Pocket SDV is connected to a PC (with its own internal HDD, operating system and applications) and sensitive data is accessed (from the Portable SDV) then temporary copies of the data may be saved on the PC's internal HDD by the operating system and/or applications. For instance, data accessed using Microsoft Word from a file stored on the Pocket SDV may leave temporary files inside temporary folders on the PC's internal HDD. As a result these folders may contain sensitive/private data of which the user may not necessarily be aware. If the PC is used by other users it may be possible for those users to find data remnants (temporary copies of files created during of processing of sensitive data) may remain on the PC's internal HDD after the user has detached the Pocket SDV. Also, if the PC does not use encryption technology to encrypt everything written to its internal HDD then it may be possible for digital forensic tools to find sensitive data if the HDD were to be obtained by an inappropriate source.

---

[1] Peter Hannay is completing his Honours degree at the School of Computer & Information Science at Edith Cowan University. The research detailed in this paper was performed for Secure Systems Ltd while Peter was performing a Western Australian Government Science & Innovation Scholarship in 2007.

[2] Peter James is registered on a Professional Doctorate programme at the School of Computer & Information Science at Edith Cowan University. Peter is the CEO of Secure Systems Ltd.

SDGuardian (Sensitive Data Guardian) is a proof of concept toolkit aimed at addressing the issue of accessing sensitive data from a USB mass storage device like the Pocket SDV in an untrusted environment, e.g. a PC, not owned by the user, is used to process sensitive data held on a Pocket SDV, then subsequently other people use the PC and are able to find remnants of sensitive data left in temporary files. A variety of technologies are employed in order to address the aforementioned issue. SDGuardian may be commercialised, depending upon market demand, and used with the Pocket SDV (or other portable products offered by Secure Systems).

AN OVERVIEW OF THE POCKET SDV: DESIGN, METHODS OF USE & CONCEPT OF OPERATION

Overview of Design

The Pocket SDV is one of a range of SDV products; the product range also includes the Laptop SDV, the SDV Duo and SDV Plus. The primary objective of the Pocket SDV is to provide strong security for data at rest[3]. The Pocket SDV is a cryptographic hardware device (James et al 2004) that asserts total control over its integral HDD at start-up and enforces correct user authentication before data on the Pocket SDV is accessible.

The encryption processes utilised by the Pocket SDV are implemented in the hardware. The hardware implementation of cryptographic functions avoids many of the inherent insecurities of a software-based approach, for example the hardware based approach ensures that keys are not present within the PC RAM; in addition the hardware implementation results in security enforcement that is transparent to the user and not dependant on the resources of the host PC.

Once successful authentication has been achieved the Pocket SDV allows access to data based on pre-defined access rights. The implementation of the Pocket SDV's security mechanisms in hardware coupled with independence from the PC's operating system ensures that successful direct attacks and/or exploitation of operating system vulnerabilities are minimised. Figure 1 provides a pictorial image of the Pocket SDB.

Figure 1: Image of Pocket SDV[4]

The Pocket SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the integral HDD. The Pocket SDV operates independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the integral HDD; ensuring the data stored on the hard disk drive is cryptographically secured at rest. A conceptual model of a Laptop SDV topology is given in Figure 2 below.
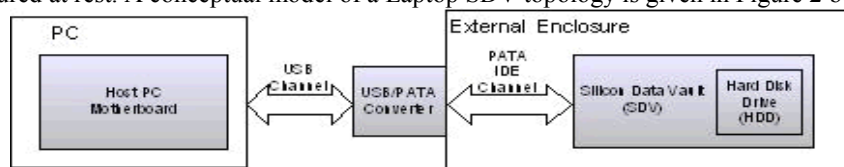

Figure 2 – Conceptual Model of Pocket SDV Topology

There are two modes of authentication supported by the Portable SDV; pre-boot and post-boot authentication. When authenticating using the pre-boot method the host PC will boot off the attached Portable SDV and the Authentication

---

3   Data at rest is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory.

4   Image of Pocket SDV made available by Secure Systems Limited.

Application (AA) will be launched from the Portable SDV's on-board flash memory. Once successful authentication has been performed the operating system on the PC's internal HDD is loaded. Authentication via the post-boot method requires that the Portable Authentication Application (PAA) is installed on the host PC. When the Pocket SDV is attached the user will be prompted for authentication details by PAA.

The authentication credentials of a Portable SDV user are tied to a specific set of access rights for each partition on the Portable SDV. These rights can be no access, read only or read/write. These user profiles could be used by different individuals or by the same individual, e.g. one user profile could be used for work and one for home.The key functionality and attributes of the Pocket SDV can be summarised as:

- *Pre-boot authentication:* The Pocket SDV achieves a high level of portability by performing authentication before the operating system has loaded. The only requirement is that the host PC provides the capability to allow a USB device to be the first boot device. Pre-boot authentication ensures no hostile software or operating system vulnerabilities can be exploited to obtain the Pocket SDV's authentication credentials.

- *Post-boot authentication:* A Pocket SDV can be authenticated to a PC running an operating system using the Portable Authentication Application (PAA).

- *Full disk encryption:* All data on the Pocket SDV is encrypted. With no data in plain text the opportunities to gain a 'starting point' to break the encryption are eliminated.

- *Sector level encryption:* Encrypting at the lowest level of formatted storage reduces the possibility that pattern matching can be performed to break the encryption.

- *Control of data channel:* Physically positioning the SDV technology between the PC USB controller and Pocket SDV integral HDD ensures all writes are encrypted. Also access control to parts of the HDD can be enforced.

- *Totally independent of PC Operating System:* The Pocket SDV behaves like a standard USB mass storage device and has no dependencies upon the PC operating system to which it is attached.

- *Security functionality implemented in hardware:* Implementing the SDV technology in an Integrated Circuit is recognised as a superior trusted platform; exploiting and attacking hardware is extremely difficult.

- *Multiple Partitions:* Up to 15 partitions can be defined for a Pocket SDV with each partition cryptographically separated from the other partitions by its own cryptographic key.

- *Differentiated Access Rights & User Profiles:* The Portable SDV allows user profiles (roles) to be defined with different authentication credentials and access rights allowing different parts of the Pocket SDV integral HDD to be accessed according to the selected user profile.

- *Audit Log:* Security related events are written to an audit log only accessible by the Pocket SDV administrator role. This log can be used for forensic purposes.

Methods of Use

The rich functionality of the Pocket SDV allows the device to be configured and used in a number of ways; three configurations are summarised below:

- *Highly Portable Secure Storage Device:* The ability to authenticate via pre-boot authentication results in a highly portable device that can be accessed via any PC capable of booting a USB device. The PAA provides the convenience of accessing data on the Pocket SDV on a fully booted system. Whilst the Pocket SDV provides Defence5 level security for data at rest, it like all other USB mass storage devices cannot prevent data remnants remaining on a host PC's internal HDD.

- *Highly Portable Secure Storage Device with SDGuardian:* As this paper will show, the SDGuardian toolkit enables a user to attach a Pocket SDV to an untrusted or semi-trusted PC with the assurance that sensitive data remnants be minimised upon completion of data processing.

- *Highly Portable Secure Storage Device with USB Bootable Operating System:* A forensically safe alternative to using the tools and techniques of the SDGuardian is to use a Pocket SDV with a USB bootable operating system, e.g. a version of Linux or Windows PE. For a specific application the bootable operating system approach provides an ideal solution. A disadvantage of this approach is that USB bootable operating systems do not contain the functionality and look and feel of Microsoft Windows XP or Vista, resulting in a disincentive for many users. Current research at Secure Systems includes the development of a business toolkit as part of a bootable operating system with a Pocket SDV.

---

[5] The SDV product range has successfully passed rigorous Australian, USA and International cryptographic and security evaluation standards.

A stepwise summary of the Pocket SDV pre and post boot authentication is given below to enable a concept of operation to be acquired.

Concept of Operation: Pre-boot Authentication

The PC must be configured to boot from a USB device at power up with a Pocket SDV attached; operation then proceeds as follows:

- The PC loads a Master Boot Record from the Pocket SDV, which in turn loads an Authentication Application (AA) stored in the Pocket SDV flash memory. N.B. While the AA is running, the user has no access to the Pocket SDV's integral HDD.

- The user is prompted to authenticate.

- The AA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the AA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PC must be powered down and restarted to continue the user pre-boot authentication process.

- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.

- The user is then prompted to select one of the following:

    o Boot an operating system from the PC internal HDD.

    o Boot an operating system held on the Pocket PC.

    o Select to authenticate another SDV.

- If the user selects to boot an operating system from the PC's internal HDD the Pocket SDV loads a Master Boot Record for the operating system on the PC's internal HDD.

- The boot process continues and loads the operating system from the PC's HDD.

- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

Concept of Operation: Post-boot Authentication

The PC must have an operating system fully booted and the PAA installed and its underlaying Windows service running; operation then proceeds as follows:

- When the Pocket SDV is attached to a USB port it is detected and a pop up authentication window presented to the user. N.B. The PAA can also be invoked to authenticate a Pocket SDV previously attached. While the PAA is running, the user has no access to the Pocket SDV's integral HDD.

- The user enters the authentication credentials and the PAA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the PAA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PAA must be restarted.

- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.

- If another portable SDV is detected the user is given the opportunity to authenticate the device.

- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

The Pocket SDV makes use of proven encryption standards and strong authentication to provide strong hardware based security for data at rest. The 'set and forget' nature of the device with all encryption handled in hardware results in a secure solution that is transparent to the end user.

SDGUARDIAN DEVELOPMENT

Motivation for Development

The objective of the research was to ensure no data remnants remain on the internal HDD of a PC used to process sensitive data retrieved from a Pocket SDV. Initial investigations considered how a utility (known as SDCleaner) could remove data

remnants, which had been written to the PC's internal HDD following the completion of data processing, i.e. a reactive approach was considered.

Research into file system structures identified that with a traditional file system (such as FAT32, NTFS and ext3) there is an area of the file system that provides a table of contents (the TOC). The TOC provides a list of all files located on the file system and where they are located logically (logical locations can be file paths such as C:\Program Files\). In addition to this logical file structure the TOC provides a list of physical locations for each logical item, these physical locations can then be used to read and write data. The key issue for any SDCleaner utility is that when files are deleted the reference in the TOC is simply removed, with the physical data remaining elsewhere on the file system.

Another issue arises when a user launches a program that is used to access sensitive data on the Pocket SDV, this data would then be copied to a temporary location on the host PC's internal HDD. Upon exiting the application it is possible that this temporary data would then be deleted in an insecure fashion (i.e. physical data remains, logical construct removed). This series of events leads to a situation in which an SDCleaner utility would have no means of locating the physical data and is therefore unable to erase said data.

It was therefore decided to adopt a proactive approach and prevent the creation of data remnants on a PC's internal HDD as a result of retrieving and processing sensitive data from a Pocket SDV. The research project was redefined as the SDGuardian. The SDGuardian would act primarily as a preventative measure, with the aim of avoiding the situation where any sensitive information reached the host PC's internal HDD in the first place. Additional measures would also be employed to securely delete sensitive data in situations where prevention is not possible.

Implementation of the SDGuardian Toolkit - Design, Capabilities and Limitations

Implementation Scenarios: The research considered a number of different scenarios where a Pocket SDV could be attached to a PC:

- *Scenario 1:* Data processing is to be performed on a "semi-trusted" PC where access to all installed applications is allowed and operation in Windows Administrator mode is permitted. Performance is also a requirement in this scenario, i.e. data processing needs to be performed at close to standard PC processing time.

- *Scenario 2:* Data processing is to be performed on an "untrusted" PC where the installed applications cannot be trusted. However, operation in Windows Administrator mode is permitted.

- *Scenario 3:* As per Scenario 2 but operation must be performed in Windows user mode, i.e. non administrator privileges are available.

To satisfy the requirements of the three scenarios SDGuardian was developed using Junction Points, Secure Deletion and Virtualisation to provide a toolkit for the proactive prevention of sensitive data from a Pocket SDV remaining on a PC's internal HDD. Table 1 shows the tools used to satisfy each scenario.

Scenario	Tool/Technique
1	*Junction points* are used to prevent specific temporary files from being written to the host PC's HDD. In addition to junction points, secure deletion is used to securely erase the Windows page file after use.
2	*Virtualisation technology running in privileged Administrator mode* is used to provide a virtualised environment where work can be performed without requiring the use of the PC's installed applications.
3	*Virtualisation technology running in non-privileged user mode* is used to provide a virtualised environment where work can be performed without requiring the use of the PC's installed applications.

Table 1: Tools/Techniques used in SDGuardian to Meet the Requirements for Each Scenario

Junction Points - File System Manipulation

The NTFS file system supports 'junction points' which are similar to symlinks under 'unix like' operating systems. These junction points allow for an empty folder on the file system to be mapped to a different physical location on the disk. The result of this is that two or more logical folders can reference the same physical data. Junction points can reference folders on a different volume or physical storage device.

An initial investigation into the currently available tools for the creation and manipulation of junction points was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source availability
Junction(Russinovich, 2006)	A command line utility that allows for the manipulation of NTFS junction points	Proprietary	Available
Junction Link Magic (Rekenwonder, 2006)	A GUI based utility that allows for the manipulation of NTFS junction points	Proprietary	Not available

Table 2: A comparison of existing junction point manipulation software

Both Junction and Junction Link Magic were used to evaluate the premise that junction points could satisfy the requirement to remap operating system and application specific directories.

There are a number of shortcomings associated with the use of NTFS junction points. The first of these is that when performing a delete operation on a junction point removal of the associated data from the disk occurs even if that physical data is referenced logically elsewhere in the file system. The window GUI however does not reflect this shortcoming and this deletion is unlikely to be noticed until the user next attempts to access this data. It is due to these factors that care was taken to ensure that junction points were not deleted with the standard tools provided by Windows; instead a custom utility was developed for this purpose.

SDGuardian used junction points to remap common temporary directories onto a partition of the Pocket SDV. This mapping was performed prior to the user accessing sensitive data located on the Pocket SDV. The result of the junction point would be that specific temporary data would never be written to the disk of the host PC, radically reducing the complexities associated with the standard methods of secure erasure.

SDGuardian removes these junction points and recreates the empty temporary directories after the user has finished working with any sensitive material. The temporary data stored on the Pocket SDV is then erased.

As a minimum junction points are used to ensure the security of the Windows temporary directories and the temporary internet directories present on a Windows system.

Secure Deletion

Secure deletion tools allow for the forensically sound erasure of data from a hard disk or other storage device, this is achieved by overwriting the data in question several times with different sets of data. Typically the data being written will be all zeros, all ones or the output of a pseudo random number generator (Gutmann, 1996). This is often accomplished by using the Windows disk defragmentation API, this API allows a logical file location to be resolved into a physical location (MSDN, 2007). Once the physical location of the data is known it is possible to overwrite this data as needed.

An initial investigation into the some of the most common tools for secure erasure was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source availability
Sdelete (Russinovich, 1999)	Command line secure erasure utility	Proprietary	Available
Eraser (Tolvanen, 1997)	Graphical secure erasure utility	GPL	Available

Table 3: A comparison of existing secure deletion software

Both Sdelete and Eraser were used to determine the best strategy to adopt for the implementation of a secure deletion solution in the SDGuardian tookit.

Virtualisation - Application Sandboxing

Application sandboxing attempts to isolate running processes from performing modifications to the host system on which they are being executed. The type of isolation depends heavily on each sandboxing application's specific implementation. There are two main types of application sandboxes, the first attempts to create multiple isolated environments on a system, while the second attempts to limit or prevent specific processes from making changes to the host environment.

There are two main approaches to implementing sandboxing, the first of these is the use of a full virtualised environment. This environment has its own operating system that runs on top of the native operating system. The second approach to implementation involves the use of kernel hooks to isolate a specific application or set of applications from accessing specific system resources (Gibson, 2006).

The use of application sandboxing utilities was investigated. The aim of such utilities is to create an environment which runs on the Pocket SDV to prevent any data from being written to the internal HDD of the host PC, the virtualisation software directs all writes to a specific partition on the Pocket SDV. Unfortunately virtualisation in itself cannot be used as a complete solution due to the nature of Windows virtual memory.

Windows makes use of a 'page file' which acts as virtual memory when adequate physical memory is unavailable. This 'page file' is located on the host PC's internal HDD (Mallery, 2006). The issue arises when a user accesses data stored on the Pocket SDV, the sandbox application can prevent all user level hard disk writes, however the Windows memory management system operates at a kernel level and as such it is not possible to prevent Windows from storing sensitive data in the Windows page file.

The use of a full virtualised environment such as those provided by VMWare or Qemu would allow a user to create a complete operating system environment that would run on top of the host machine's operating system. The advantage of this is that the majority of file system writes would be contained within the virtual machines disk image file, this image would be stored on the Pocket SDV. An advantage of this implementation is that the user would have the ability to install software such as a word document viewer within the disk image, thus negating the need for this software to be present on the host system.

A range of application sandboxing/virtualisation solutions can be found in the table below.

Software name	Description	License type	Source availability
Sandboxie (Tzur, 2006)	An application sandbox utility capable of redirecting file system writes to a specified location.	Proprietary	Not available
Vmware (Vmware, 2007)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Paralells (Paralells, 2006)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Mojopac (MojoPac, 2006)	A sandboxing utility capable of creating an isolated enviroment on a host computer. MojoPac is intended to be installed on a portable storage device.	Proprietary	Not available
Qemu (Bellard, 2006)	A full virtualization application capable of emulating a host computer completely in software, as such administrator rights are not needed on the host computer.	GPL	Available

Table 4: A comparison of existing sandboxing / virtualisation software

The full range of application sandboxing/virtualisation software specified in Table 4 were tested. Qemu was selected for SDGuardian due to its ability to execute in both Windows Administrator and User modes.

Implementation Life Cycle

The proposed solution went through a phase of requirements specification and development over the period of several weeks. The SDGuardian was written primarily in the C# programming language. The following features were implemented:

- Junction Points
- Secure Deletion
- Virtualisation

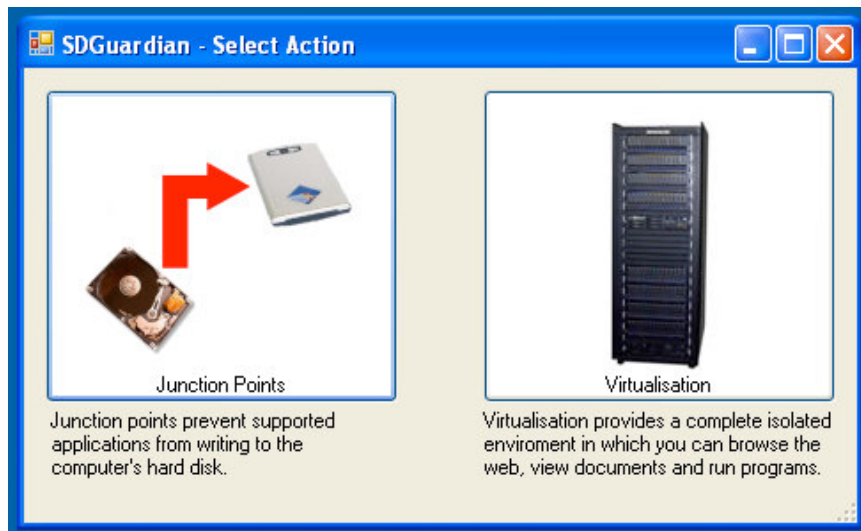A series of screenshots of the SDGuardian application itself are provided below:
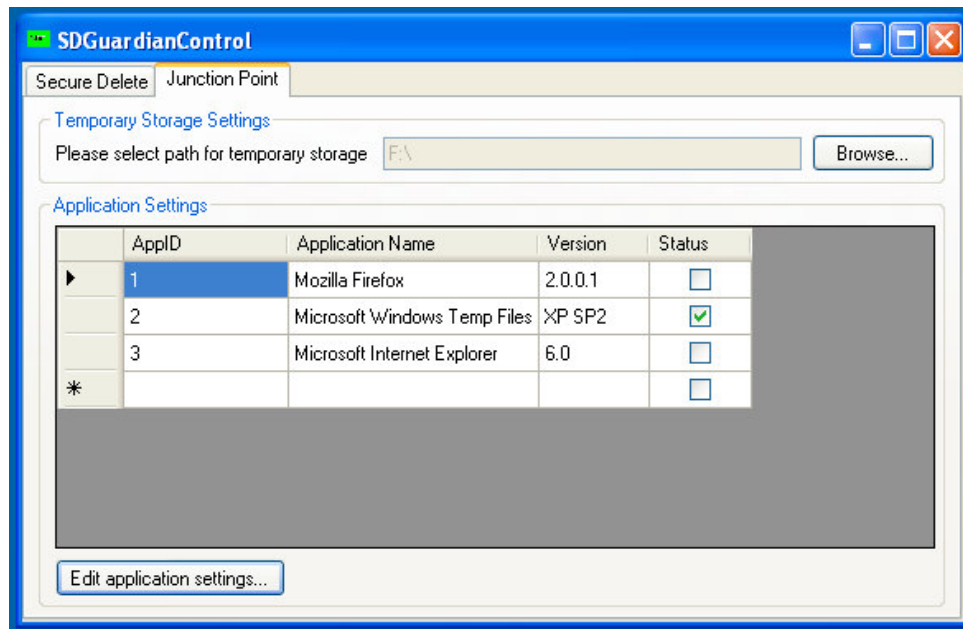
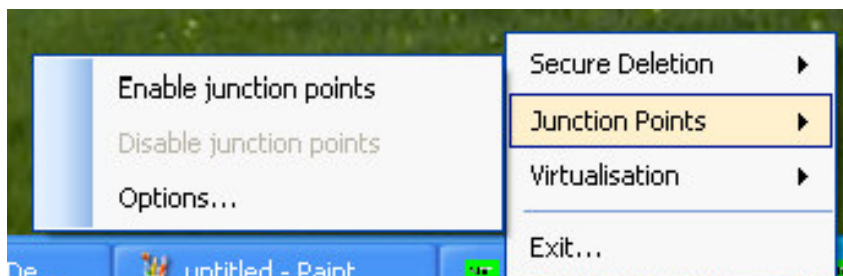*Figure 3: Main screen*

Figure 4: Junction points options dialog
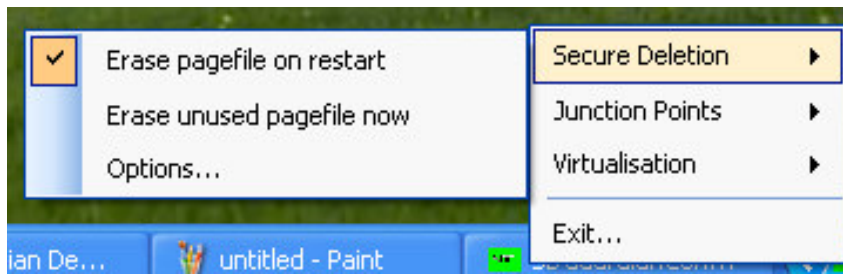

Figure 5: Junction points menu
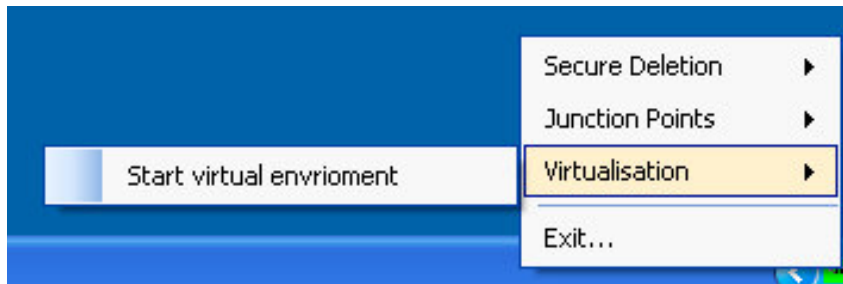
Figure 6: Secure deletion menu

Figure 7: Virtualisation menu

CONCLUSION & FUTURE DEVELOPMENT

The research and development of the SDGuardian was successfully achieved. A proof of concept implementing junction points, secure deletion and virtualisation was developed which meets the original goals of the project. Plans have been made for the continued development of the SDGuardian. These plans include improvements to the virtualisation system employed, additional focus on portability and an in depth forensic evaluation of the software.

REFERENCES

Bellard, F. (2006). "QEmu." Retrieved January 11, 2007, from http: fabrice.bellard.free.fr/qemu/about.html. MojoPac. (2006).

James, P. & Wynne, M 2004, Securing Data at Rest, 2nd Australian Information Security Conference, Edith Cowan University, Perth November 2004.

Gibson, S. (2006, Oct 26, 2006). "Security Now - Transcript of Episode #63." Retrieved 11 January, 2007, from http://www.grc.com/sn/SN-063.htm.

Gutmann, P. (1996). Secure Deletion of Data from Magnetic and Solid-State Memory. Sixth USENIX Security Symposium, San Jose, California.

Mallery, J. R. (2001, December 6, 2006). "Secure File Deletion: Fact or Fiction?" Retrieved January 11, 2007, from http://www.cybercrimelaw.org/documents secure_delete.pdf.

MojoPac. (2006). "What is MojoPac?" Retrieved January 11, 2007, from http: www.mojopac.com/portal/content/what/.

MSDN. (2007). Defragmenting Files.   Retrieved 8th of January, 2007, from http://msdn2.microsoft.com/en-us/library/aa363911.aspx

Parallels. (2007). "Parallels Workstation." Retrieved January 11, 2007, from http: www.parallels.com/en/products/workstation/.

Rekenwonder Software. (2007). Junction Link Magic, Rekenwonder Software.

Russinovich, M. (1999). SDelete - Secure Delete, Systems Internals.

Russinovich, M. (2006). Junction, Systems Internals.

Tolvanen, S. (1997). Eraser, Heidi Computers Limited.

Tzur, R. (2006, 14 December 2006). "Sandboxie." Retrieved January 11, 2007, from http://www.sandboxie.com/.

VMware. (2007). "VMware: Virtualization, Virtual Machine & Virtual Server Consolidation." Retrieved January 11, 2007, from http://www.

COPYRIGHT