2008

# Secure Portable Execution Environments: A Review of Available Technologies

Peter James
*Secure Systems Ltd*

DOI: 10.4225/75/57b55d20b876d
Originally published in the Proceedings of the 6th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 1st to 3rd December 2008.

# Secure Portable Execution Environments: A Review of Available Technologies

Peter James[1]
Secure Systems Ltd
pjames@securesystems.com.au

## Abstract

*Live operating systems and virtualisation allow a known, defined, safe and secure execution environment to be loaded in to a PC's memory and executed with either minimal or possibly no reliance on the PC's internal hard disk drive. The ability to boot a live operating system or load a virtual environment (containing an operating system) from a USB storage device allows a secure portable execution environment to be created. Portable execution environments have typically been used by technologists, for example to recover data from a failing PC internal hard disk drive or to perform forensic analysis. However, with the commercial potential of portable execution environments becoming realised the requirement for such environments to be secure is becoming increasingly important. To be considered truly secure a portable execution environment should require authentication prior to loading the executing environment (from the USB mass storage device) and provide full encryption of the whole mass storage device.*

*This paper discusses the outcomes from building four portable execution environments, using commercially available and/or freeware technologies. An overview is given of the emerging commercial requirement for secure portable USB execution environments, the security threats addressed and research performed in the area. The technologies and products considered in the review are outlined together with rationale behind the selection. The findings from the implementation of the four portable execution environments are discussed including successes, failures and difficulties encountered. A set of security requirements is defined which is used to gauge the effectiveness of each of the four environments.*

**Keywords**
USB boot, Live Operating Systems, Virtualisation, Encryption, Pre-boot Authentication, Post-boot Authentication, U3 technology.

## INTRODUCTION

Live operating systems (OS) are designed to be loaded from a Compact Disk (CD) or a Universal Serial Bus (USB) storage device into a PC's Random Access Memory (RAM) and execute without necessarily having access to the PC's Hard Disk Drive (HDD). A live OS enables a user to rapidly boot an execution environment and execute an application on an available PC without having to install the application on the respective PC HDD; a live OS can provide a platform for a Portable Execution Environment (PEE). Using a USB storage device to hold and load the OS (and applications) provides a number of advantages over a CD based live OS (and applications); advantages include a faster load time, the ability to store data generated during a session, configuring the storage device to provide OS swap space (page file/virtual memory) and a more convenient size and shape to a regular CD. However, a notable disadvantage of a USB storage device over a CD is the inconsistent support provided by the different manufacturer's PC Basic Input Output System (BIOS) to allow the booting of a live OS from a USB storage device.

Virtualisation (often referred to as virtual machines or virtual environments) provides an abstract execution environment separate from the physical PC. There are a number of different types of virtual machine (VM) (CSIRO 2008). The type of virtualisation considered in this paper is a VM that runs within (on top of) the PC operating system; often referred to as a type 2 hosted VM. A "guest" OS is hosted within the type 2 VM and the application is executed within the guest OS. A VM and its guest OS and application can be loaded from a USB storage device into an executing PC to provide a PEE.

The prolific growth in public Internet access centres (e.g. Internet cafes, airport lounges, wireless hot spots, etc) allow an individual to perform sensitive transactions, like Internet banking, on PCs for which no level of trust can be assumed, i.e. these PCs and/or any communication infrastructure may contain malicious software. Also a lack of best practice security (e.g. anti-virus and anti-spyware software and modem/router enabled firewall capabilities) in many homes can result in malicious software residing on PCs unbeknown to the owner/user.

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

Malicious software is capable of capturing and exploiting user credentials (enabling identity theft/fraud to occur) and/or stealing sensitive data when a user is conducting an Internet transaction.

Secure PEEs provide an opportunity for organisations to distribute an Internet application on a USB storage device that can be used safely on PCs which may contain malicious software. In this paper the following business scenario is presented to provide the context for the provision of a secure PEE.

> *A compact and easily transportable secure PEE device is distributed by an organisation to individuals for a specific purpose. The secure PEE contains an application that interacts with a server over the Internet. The individual will use the secure PEE device and its application on PCs for which no level of trust can be assumed, therefore the secure PEE device must provide an environment that can protect against malicious software, i.e. the integrity of the secure PEE must be preserved. It is important that the PEE can be reliably loaded into any PC.*
>
> *The secure PEE device should also allow the application or individual to save data on to the secure PEE device. The data may be sensitive and therefore protecting the confidentiality of the data is important. Stored data should be accessible by the individual (when the secure PEE is not being used) from a PC running Windows XP. Any stored data is likely to be sensitive and therefore protecting the confidentiality of the data is important.*
>
> *The secure PEE device, due to its compact nature, could easily be mislaid and therefore ensuring its contents can not be exploited is paramount.*

To satisfy the above business scenario a secure PEE device would utilise authentication prior to loading the PEE, data encryption, information separation, user/application authentication and device attestation. In this paper a range of available technologies are considered to construct USB based secure PEE devices. The approach adopted is to consider a secure PEE device constructed using a standard thumb drive and available open source technologies and then progressively consider increasingly more sophisticated and expensive technologies that provide additional security capabilities.

Four secure PEEs were constructed that provide the underlying security infrastructure to enable applications to perform secure Internet transactions. The security technologies required by an application to perform Internet transactions are not considered to be within the scope of this paper. The following four PEE devices were constructed:

- Secure PEE Device 1 - Standard Flash Memory Device: A low cost thumb drive with bootable OS, an OS in a VM, partitioning and software based encrypted storage.
- Secure PEE Device 2 - U3 Flash Memory Device: A U3 thumb drive (SanDisk 2008) with bootable OS, an OS in a VM, partitioning and encrypted storage.
- Secure PEE Device 3 - Hardware Encrypted Flash Memory Device: The IronKey thumb drive (IronKey 2008) incorporating hardware based encryption with an OS in a VM.
- Secure PEE Device 4 - Portable Hardware Encrypted Hard Disk Drive: The Pocket Silicon Data Vault (SDV) (SecureSystems 2008) incorporating hardware based encryption with bootable OS, an OS in a VM and cryptographically separated partitions.

Each secure PEE device is assessed to gauge the effectiveness of the technologies utilised to counter a set of key threats and satisfy a set of security requirements.

The following terms are used throughout the paper, a definition is therefore given for each:

- **secure PEE device:** the secure platform/infrastructure consisting of a USB mass storage device configured with a secure PEE, secure storage space and possibly hardware based security mechanisms/technologies (e.g. encryption and secure partitions if available).
- **secure PEE:** the trusted OS, trusted application(s), security technologies (e.g. authentication and software encryption) and the appropriately configured hardware security mechanisms (if any) of the secure PEE device.
- **secure PEE OS:** the trusted OS component of the secure PEE.
- **trusted OS:** an OS that has been acknowledged as secure by the supplier and users. To be considered trusted the OS may have a reduced set of hardened functionality and/or been subjected to independent rigorous evaluation and testing.
- **PEE:** a portable execution environment that does not necessarily have any security technology nor has been specifically configured to be secure.

- **portable storage device:** a USB flash device (often know as a thumb drive or pen drive) or a USB HDD packaged in a portable enclosure.

## SECURE PORTABLE EXECUTION ENVIRONMENTS – THE REQUIREMENT

### Emerging Business Requirement

Live OS', VMs and other portable software, executable from CDs and USB storage devices have been available for a number of years; enabling PEEs to be built. Traditionally such PEEs were typically used by technicians to perform PC maintenance activities (e.g. data recovery from a corrupt or failing HDD) or by specialist IT security/forensics specialists as 'tools of the trade'.

The large data and financial losses organisations and individuals are suffering through malicious software exploiting Internet transactions is well known and documented. Organisations are increasingly looking for assurance that Internet based transactions can be performed securely and that user credentials and data will not be compromised. However, when an organisation allows a service to be provided over the Internet it is not able to control the environment from where the service is initiated, and therefore gaining assurance that a transaction has been performed securely is not possible. To combat the data and financial losses organisations are starting to consider the distribution of secure PEE devices to their staff and/or customers to protect against malicious software and enable secure Internet transactions to be performed.

Notable research on the provision of secure portable applications and secure PEEs includes the work conducted by the Commonwealth Scientific & Industrial Research Organisation (CSIRO) and its proof of concept Trust Extension Device (TED) (Chan et al 2007 & Nepal et al 2007). The CSIRO work focussed on trust portability, where a VM is used to encapsulate a trusted computing environment in to a USB storage device that can be plugged in an untrusted PC and used to perform secure Internet transactions. The key requirement of the TED research was to provide an 'on demand' secure PEE where the user and device identity, and the integrity of the execution environment could be confirmed. The research and development focussed on user, application and device attestation to ensure identities could be confirmed before an application performing an Internet transaction was initiated. Similar research into the use of VMs to provide trusted/secure PEEs has been performed at Stanford (Garfinkel et al 2003) and Princeton (Kwan et al 2007).

The CSIRO work was based on requirements gathered from a user forum with members drawn from the finance sector, government agencies and other service organisations all of whom had mobile workers and/or customers. The user forum identified the need for secure PEEs that enable secure Internet transaction to be performed. The CSIRO TED proof of concept device did not implement the secure infrastructure considered (in this paper) as vital underlying security technology in the provision of a secure PEE.

### Threat Environment Addressed

It is proposed in this paper that a secure PEE device provide countermeasures to address the following three key threats:

1. Malicious software (residing on a host PC) capturing user credentials and data.
2. Sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of an Internet transaction; and
3. As a result of loss or theft, unauthorised access is gained to (sensitive) data held on an unsecured USB storage device that was used to store data generated during an Internet transaction.

### Security Requirements

It is proposed in this paper that a secure PEE needs to provide functionality to address the following security requirements:

- The secure PEE shall only allow authorised users to load the resident OS/application(s) and access stored data.
- The secure PEE shall preserve the integrity of the resident OS and application(s) from malicious software and other external attacks.
- The secure PEE shall preserve the confidentiality of any stored data from external attacks including theft of the device.
- The secure PEE shall leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session.

- The secure PEE shall prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen.
- The secure PEE shall confirm both the identity of the user and the device before allowing an Internet transaction to occur.

These security requirements will be used to gauge the security strength of the four constructed secure PEEs.

**Secure PEE – An Overview of the Required Functionality**

To address the security requirements and threats identified above, and to enable the aforementioned scenario to be satisfied it is proposed in this paper that a secure PEE would ideally utilise the following functionality:

- ***Bootable secure PEE OS (pre-boot secure PEE OS):*** Cold booting a (trusted) live OS from the a secure USB PEE device onto a host PC provides a secure platform as the user can have a high degree of assurance that no malicious software will be present, i.e. any malicious software that was active on the host PC would be eliminated by the power-off required prior to the cold boot. However, booting a live OS from a USB device is neither user friendly nor reliable. To boot a live OS from a USB device requires the boot order to be changed. Some new PCs allow the user to select the boot device at power up whilst other PCs require a user to set the boot order in the PC BIOS. Whilst setting the boot order is not a technically complex task the following aspects make the process unfriendly and unreliable:
    - There are many different types of BIOS and there is no standard set of keystrokes to enter the BIOS and change the boot order.
    - Some older PC's do not support booting from a USB device.
    - Older PC's usually only support USB1.1 which can make the booting of a live OS a slow process.
    - Some public access PCs password protect the BIOS which can prevent a user changing the PC boot order.

  Although loading an OS from a cold boot provides a secure platform, given the unfriendly aspects of USB booting means it should not be considered as the only method of loading a secure PEE into a host PC.

- ***Secure PEE in VM (post-boot PEE):*** As booting a secure PEE directly from a USB storage device has some usability problems an alternative approach to providing a secure platform is to load a VM containing a secure PEE. VMs provide security through isolation, however VMs are susceptible to the following vulnerabilities:
    - Keyloggers: The host OS may contain malicious software that can capture and store keystrokes.
    - Screen shot logging: The host OS may contain malicious software that can capture and store screens.
    - Memory Probing/Attacks: The host OS may contain malicious software that can capture the contents of RAM utilised by the VM.

  The business scenario calls for a secure PEE that can be reliably loaded on a range of PCs and protect against malicious software. A bootable OS protects against malicious software but cannot be reliably loaded. Conversely VMs are susceptible to malicious software but can be easily and reliably loaded. Therefore secure PEEs were built that have both a bootable OS and a VM with a guest OS; allowing the user to select the secure PEE OS that suits the user and/or operating environment. Note – it should be assumed that the bootable OS and VM guest OS are identical and support the same (trusted) application which executes identically on both bootable OS and VM guest OS.

- ***Authentication:*** A secure PEE device should not be accessible until the user has entered authentication credentials. The following two authentication methods are considered:
    - *Pre-boot:* The most secure way to authenticate a user is before the live OS or host OS is loaded; it is highly unlikely that malicious software (such as keyloggers) can be present as the respective OS will be loaded from a cold boot. In pre-boot authentication an application is loaded into the PC upon power up and the user authenticates with the secure PEE device. However, pre-boot authentication will experience the same set of pre-boot problems identified above for a bootable secure PEE OS, i.e. pre-boot authentication can be secure but unfriendly to initiate.

- *Post-boot:* A more convenient approach to authenticate a secure PEE device is to plug it into a PC that has a booted and executing OS. In post-boot authentication a PC either uploads an authentication application from the secure PEE device or a pre-installed authentication application is resident on the PC. Through the authentication application a user authenticates with the secure PEE device. However, post-boot authentication can be subject to attack by malicious software (e.g. keyloggers) resident within the host PC.

Where possible, secure PEE devices will be built that can support both pre-boot and post-boot authentication.

- ***Device encryption:*** To preserve the confidentiality of the secure PEE OS and data residing on the USB device, on-the-fly encryption is required. On-the fly-encryption can be provided by software or hardware; this paper will consider both software and hardware encryption.
- ***A secure PEE that does not store data on the host PC HDD:*** To prevent data remnants residing on the host PC HDD following the use of a secure PEE, the secure PEE device must be configured to:
  - provide swap space (virtual memory/page file) for the secure PEE OS on the secure PEE device itself; and
  - ensure the secure PEE OS and application(s) write all temporary information to available allocated space on the secure PEE device.
- ***Ability to separate the PEE OS and data:*** To preserve the integrity of the PEE OS and any stored data the secure PEE device should support storage partitioning and role based differentiated access rights to partitions. Such partitioning allows separation and isolation to be achieved.
- ***Protection of PEE OS and data:*** In addition to the provision of a partitioning capability a secure PEE should also allow a partition to be defined as Read-Only; such partitions can be used to protect the integrity of the PEE OS from malicious software. Read-Only partitions can also be used to protect the integrity of 'valued' data.
- ***User, application and device authentication and attestation:*** To enable a secure Internet transaction to occur an application on the secure PEE device needs to mutually authenticate with a remote server providing the service. Assurance is required that the application and secure PEE device are genuine; therefore device and application attestation technology is required. The techniques and technology required to perform application and device mutual authentication and attestation with a remote server are considered beyond the scope of this paper.

## THE TECHNOLOGIES/PRODUCTS SELECTED

The range of technologies that satisfy the functionality requirements and enable secure PEE devices to be constructed is growing rapidly. The technologies considered in this paper represent what the author considers to be amongst the best available at the time of writing (May/June 2008). Both freeware and proprietary technologies were considered. A summary is given below of the technologies and products utilised to build the four secure PEE devices.

### The Live Operating System

A range of live OS' were considered, including Windows MiniPE, Ubuntu, Puppy Linux and Slax. Following an evaluation of a set of live OS' the Linux distribution Slax (Slax 2008) was selected. Slax is a cut down Linux distribution based on Slackware and has been developed primarily as a live OS. Slax was selected for the following reasons:

- The ease with which Slax could be installed on a USB storage device as a bootable OS (PendriveLinux.com-Slax 2008).
- The quality of documentation available (Wielenge 2008).
- The compact yet functionally rich distribution which rapidly loads into a PC's RAM.
- The ability to boot first time every time on a range of PCs and allow Internet access without any configuration of the OS.
- Slax was rated best live OS on "The LiveCD List" (Brand 2008).

The other OS' considered presented a range of problems including inability to boot from a USB device without time consuming configuration, slow to load, would not load into a PC with limited RAM and would not load consistently into a range of different PCs.

For the purposes of this paper Slax is assumed to be secure and is considered to be a trusted OS. In practice the OS selected would be subject to analysis to harden and remove functionality considered unnecessary for a secure PEE. Version 6.0.6 of Slax was used.

**Virtual Environment**

The virtual environment Qemu was selected for the review. Qemu (Bellard 2008) is a freeware type 2 VM. Qemu has gained widespread recognition as an effective VM due to the ease by which it can be used and configured (Hannay & James 2007). Qemu was selected for the following reasons:

- It is an extensively tried and tested open source VM.
- It can be loaded from a USB storage device on to a PC executing Windows XP without requiring any installation or Windows XP administrator privileges; although this approach to using Qemu does result in slower execution times for the guest OS and application(s).
- A Qemu image containing Slax was readily available from www.pendrivelinux.com (PendriveLimux.com-Qemu 2008)

**Software Encryption**

The open source, freeware application Truecrypt (TrueCrypt 2008) was selected to provide software encryption. Truecrypt is a comprehensive application providing encryption for PC HDDs and portable storage devices. Truecrypt has a traveller mode option that allows a number of encrypted volumes to be placed onto a USB storage device and then accessed from any PC. When a USB storage device is configured with traveller mode a Truecrypt autoexec application (known as the 'traveler disk') is placed on to the USB device; the Truecrypt application is obviously in unencrypted form. When the USB device is plugged into a PC the user is given the option to execute Truecrypt where upon, following successful authentication, a Truecrypt volume on the device is mounted and appears as a Windows removable device.

Truecrypt was selected for the following reasons:

- It is an extensively tried and tested open source package.
- It provides strong 'on the fly' encryption, allowing selection from a variety of crypto and hashing algorithms.
- It is well documented (TrueCrypt-Foundation 2008).
- It is one of only a few freeware packages that support encrypted volumes on portable storage devices.
- The autoexec/automount feature uses the standard Windows XP popup.
- A Truecrypt volume can be mounted Read-Only.

Some of the limitations of the Truecrypt traveller mode option include:

- the requirement for the host PC to be executing with Windows administrator mode.
- it is not possible to encrypt a whole storage device partition (in traveller mode).
- Truecrypt volumes can be identified on a storage device.
- only one volume can be automounted.
- the autoexec application is easily identifiable on the storage device and therefore susceptible to attack.

**U3**

A U3 flash drive was selected as one of the platforms for the secure PEE device. U3 technology (SanDisk 2008) allows a user to load a set of applications, and launch them from a U3 flash drive whenever the U3 flash drive is plugged into a PC executing Windows XP. The seamless launch of applications is achieved by the U3 hardware (the U3 chip) reserving a small part of the device (containing an ISO image) that is interpreted by Windows XP as a CDROM. When the U3 device is plugged into a PC executing Windows XP an autoexec of the ISO image occurs. The U3 ISO provides an interface that allows applications to be selected and launched.

Figure 1: An example of a U3 USB flash memory device

As can be seen from Figure 1, a U3 flash device looks like a standard thumb drive[2]. U3 flash devices are widely used and some interesting applications of the technology have been achieved (Al-Zarouni 2006). A U3 flash drive was included in the research because:

- it provides a more sophisticated USB flash drive
- given the wide scale adoption of U3 devices it was considered important to determine how effective a U3 device would be when configured and used as a PEE device.

**Hardware Encryption**

Whilst software based encryption provides adequate data protection to preserve the confidentiality for many organisations it is considered insufficient to protect highly sensitive data and applications. The main vulnerability is that the data encryption key for a software encryption package is stored in a PC's RAM (during operation) and therefore potentially subject to capture. Recent research performed by a team from Princeton (J Alex Haldermany et al 2008) has shown that under certain circumstances it is possible to retrieve an encryption key from a PC's RAM even after the PC has been powered down. Hardware based encryption usually stores the encryption key in the hardware crypto engine on the storage device.

It is considered particularly important that a secure PEE device support hardware encryption due to the way secure PEE devices will be distributed and used. Two storage devices that use hardware based encryption were selected; IronKey and the Pocket Silicon Data Vault (SDV). Both devices have a crypto engine implemented on a integrated circuit located within the enclosure or the storage device.

**IronKey**

Since mid 2007 an increasing number of USB flash storage devices (packaged as thumb drives) have become available with full storage encryption performed by an integral hardware crypto engine; IronKey is an example of such a device. An IronKey is slightly larger than a typical thumb drive.


Figure 2: An example of a IronKey flash memory device

[2] Image from U3 website

Figure 2 presents a pictorial image of an IronKey and Figure 3[3] presents a cross sectional image of an IronKey showing the crypto & access control chip and flash memory. The IronKey chip controls all access to the flash memory.

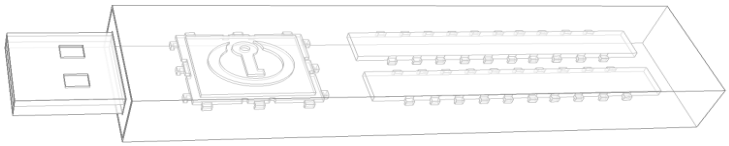

Figure 3: Pictorial cross-section of IronKey

Some of the important features of IronKey include:

- Post-boot authentication – the IronKey has implemented a "U3 like" feature where the IronKey chip reserves a small part of the device that is interpreted by Windows XP as a CDROM. When the IronKey is plugged into the PC an autoexec of an authentication application occurs.
- Strong encryption – the IronKey crypto engine implements the strong 128 bit Advanced Encryption Standard (AES).
- Key escrow (key recovery) – when the IronKey is first initialised a key escrow capability ensures that if the authentication credentials are forgotten they can be recovered from the IronKey website.
- Key destruction – after ten failed authentication attempts the IronKey encryption key(s) are destroyed and the device becomes unusable.

IronKey was selected due to its compact form factor (it is only slightly larger than a standard USB thumb drive (pen drive). The IronKey does however have the following limitations:

- Support for Windows XP/Vista only – an IronKey can only be authenticated with XP/Vista. As Windows only recognises the first partition on a thumb drive the IronKey can only be recognised as a single partitioned device.
- Key escrow limitation – the IronKey key recovery facility requires the user to remember a set of web authentication credentials. Also as the recovery key and authentication credentials are held outside the user's control the assurance afforded to their security can not be guaranteed.
- No pre-boot support - an IronKey can only be authenticated by post-boot authentication.

**Pocket SDV**

The Pocket SDV is a portable 2.5 inch form factor HDD. The Pocket SDV contains an 'SDV chip' which controls access to an interal 1.8 inch HDD. The SDV chip encrypts all data written to the 1.8 inch HDD. 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 system will boot off the attached Portable SDV and the Authentication Application (AA) will be launched from the Portable SDV's on-board flash memory. Once successful authentication has been performed the secure PEE is loaded. Authentication via the post-boot method requires that the Portable Authentication Application (PAA) is installed on the host PC.

Figure 4: An example of a Pocket SDV HDD

Once successful authentication has been achieved the Pocket SDV allows access to data based on pre-defined access rights. The Pocket SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the integral HDD.

[3] Images obtained from IronKey website.

Figure 4 provides a pictorial image of the Pocket SDV[4]. The key functionality and attributes of the Pocket SDV can be summarised as:

- Full disk encryption - all data on the Pocket SDV is encrypted; with encryption performed at the sector level which reduces the possibility that pattern matching can be performed to break the encryption.
- 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; it works with Linux as well as Windows.
- Multiple Partitions - up to 15 partitions (drives/volumes) 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 - a Pocket 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.

The Pocket SDV was selected due to its comprehensive functionality; however it does have the following limitations:

- Form factor – the shape and size of the Pocket SDV is not as convenient as the USB thumb drive.
- Pre-installation of PAA – post-boot authentication can only be performed after the PAA has been installed onto the host PC.

## BUILDING SECURE PORTABLE EXECUTION ENVIRONMENTS

### Host PC Technology Constraints

The following constraints in the host PC technology influence how secure PEEs can be constructed:

1. Unfriendly PC BIOS' and a lack of user experience changing a PC boot order can result in problems when assigning a secure PEE device as the first boot device.
2. Windows XP only supports FAT (File Allocation Table) file systems for USB flash devices.
3. Windows XP only supports access to the first FAT partition (volume/drive) on a USB flash drive[5].

It is again worth reviewing both the BIOS boot and VM vulnerability issues as building secure PEEs that address these issues is an important proposition in this paper. Changing a PC BIOS setting can be difficult, inconvenient and in some instances impossible to change (i.e. the BIOS is locked). Building a secure PEE based purely on a bootable OS is impractical. As noted above, to provide both a high level of security and convenience the secure PEEs constructed will contain both a bootable (trusted) OS & application(s) and a loadable VM containing a (trusted) OS & application. The bootable OS provides the most secure execution environment but it is an unfriendly activity to enable a secure PEE device to be the first boot device. Alternatively loading a VM with a guest OS is a user friendly action but is vulnerable to attacks from malicious software that may be resident on the host PC. By providing a secure PEE with multiple partitions with a bootable OS in one partition and a VM with a guest OS in another partition, allows the user to select either execution environment based upon:

- user skill level.
- allowed access to the PC BIOS.
- access to PC power on/off switch.
- the level of trust that can be placed in the host PC.
- user convenience.

[4] Image obtained from Secure Systems web site.
[5] This is not the case for a USB portable HDD where Windows drivers will mount all FAT partitions on the HDD.

The Windows limitation of only supporting FAT file systems on a flash drive and also only recognising the first partition identified on a flash device limits the opportunity to provide an elegant secure PEE solution using a single partition. A single partition could be used to hold a bootable OS, a VM (with OS) and user generated data. However, a single partition solution would not readily support encryption and data separation and therefore the aforementioned secure PEE requirements for confidentiality, integrity and preventing data remnants would be difficult (probably impossible) to satisfy. One solution to overcome this Windows limitation and satisfy the secure PEE requirements is to use multiple partitions (a mixture of FAT and Linux file systems) as described below.

**Secure PEE Configuration Decisions**

Secure PEE devices were built using multiple partitions; this approach enables certain secure PEE requirements and aspects of the business scenario to be achieved for secure PEE devices built using USB flash drives.

It is important the first partition on a secure PEE device has a FAT file system so that it is recognised by Windows XP, if the first partition were to be a Linux partition Windows would not be able mount it and as only the first partition is recognised by a USB flash device no other partitions on the device would be mounted. Whilst the selected OS (Slax) can be installed as a bootable OS on a FAT partition it must however be the first partition. If the bootable OS is in the first partition then it will not be possible to have a VM in a second partition that can be loaded into a PC executing Windows XP. Therefore the first partition on a secure PEE device must be a FAT file system containing the VM and guest (trusted) OS and application(s).

A Linux ext3 partition was required as the second partition to hold a bootable copy of Slax. The boot loader 'lilo' was used to configure the Slax Master Boot Record (MBR) so that Slax would boot from the second partition.

An outcome from the use of a Linux partition is that when the user is accessing the secure PEE device from Windows XP the Linux partition is not mounted and therefore the possibility of the bootable OS being attacked or corrupted is reduced.

**Secure PEE Configuration Model**

Based on the above configuration decisions the configuration model presented in Figure 5 defines a target configuration for a secure PEE device. For each secure PEE constructed an attempt was made to implement the target configuration model. The configuration model presents four partitions. The first is a Windows (FAT 16/32) partition containing a VM hosting an OS with additional space for operational data. The second partition is a Linux partition with a bootable OS. Ideally the first and second partitions will be Read-Only partitions with the Read-Only mechanism enforced by the secure PEE device infrastructure. The third (Linux) partition will be swap space for both the VM guest OS and the bootable OS. Finally the fourth partition (with a FAT file system) will provide storage space for user generated data; whilst the fourth partition is a target requirement in practice it is only possible for a secure PEE implemented on a USB portable HDD. All partitions will be primary partitions and ideally fully encrypted.

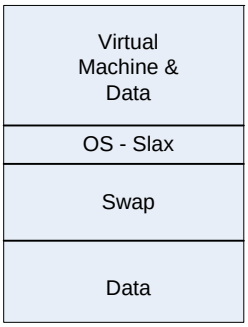


Figure 5: Target Configuration Model

**Secure PEE Device 1 - Implementation on a Standard Flash Memory Device**

The flash memory device used was a Buffalo 2GB high speed thumb drive. The target configuration model could not be fully achieved as only one FAT file system can be mounted; therefore the first partition was used for the VM and user created data. Figure 6 models the implemented configuration. Truecrypt was used to protect the VM and user generated data; two truecrypt volumes (containers) were generated in the first partition, one for the VM (containing Slax and an application) and another for user generated data – the truecrypt volumes provided separation. The second and third partitions were Linux partitions; the second an ext 3 (journaling) file system for a bootable Slax and the third an ext2 file system for OS swap space – the swap partition was used by both the VM running Slax or the bootable Slax.

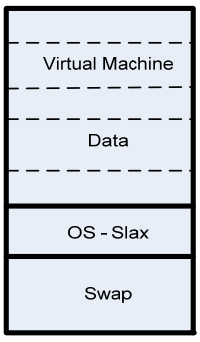


Figure 6: Configuration Model for Secure PEE Device 1

The following observations were made during the construction and testing of secure PEE device 1:

- The Slax distribution included the script "liloinst.sh" which allowed the second partition on the secure PEE device to be easily configured as a bootable partition.
- Booting Slax from the second partition was tested on a range of PCs and only once (on a Dell 5400 PC) did it fail to boot.
- Ubuntu (on a live CD) with fdisk provided a simple system to configure the device's partition table, file systems and bootable version of Slax.
- The truecrypt volumes were easy to create and use, they could however be identified on the secure PEE device.
- When truecrypt was configured for autoexec and the secure PEE device was plugged into the host PC a standard Windows 'removable device' popup appeared, from which the truecrypt option could be selected resulting in the truecrypt authentication screen appearing. Upon successful authentication the first partition on the secure PEE device is opened as a 'removable device' with a truecrypt label.
- When truecrypt was configured for 'automount upon authentication' the truecrypt volume containing the VM was automatically mounted. Only one truecrypt volume could be automounted. The automounted truecrypt volume is presented as a 'removable device' by Windows XP.
- The VM and its guest OS executed from a truecrypt volume without problems, however when the truecrypt automount Read-Only feature was used for the truecrypt volume containing the VM, the VM encountered problems booting Slax (the guest OS). Therefore it may not be possible to protect the integrity of the VM & its guest OS by using the truecrypt Read-Only mechanism.
- When the truecrypt volume used to store and protect user generated data was mounted it was mounted as a 'local disk' rather than a 'removable device', which could confuse a user.
- Limited testing showed that both the bootable OS and VM guest OS appeared to use the swap partition successfully, albeit with some degradation of performance.

**Secure PEE Device 2 - Implementation on a U3 Flash Memory Device**

The U3 device used was a SanDisk 2GB flash drive. Like the secure PEE device 1, the target configuration model could not be fully achieved as only one FAT file system can be mounted. Therefore the first partition was used for the VM and user generated data. Figure 7 models the implemented configuration. Also, like secure PEE device 1, truecrypt was used to protect the VM and user generated data. The second and third partitions were Linux partitions and configured exactly like secure PEE device 1.

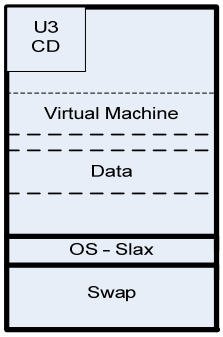


Figure 7: Configuration Model for Secure PEE Device 2

The following observations were made during the construction and testing of secure PEE device 2:

- Most of the observations made above with respect to secure PEE device 1 were found to apply to the U3 based secure PEE device 2. The notable differences occurred when the U3 authentication feature was enabled, as highlighted below.
- The truecrypt automount works with U3. The U3 red cruzer icon appears in addition to the windows popup enabling truecrypt to be selected.
- The U3 (post-boot) Windows authentication feature prevents any upload from the U3 device until successfully authentication. Therefore the truecrypt autoexec and automount features could not be utilised if the U3 authentication feature was enabled.

- Although a post-boot authentication mechanism, if the U3 authentication is enabled the secure PEE bootable OS was blocked from loading. Similarly if the U3 secure PEE device was plugged into a host PC running Linux no access to any partitions was possible, i.e. the U3 authentication feature appears to block all access to the device until successful authentication.
- When the two truecrypt volumes (in the first partition) were mounted each volume was labelled as a 'local disk', whilst the U3 secure PEE device first partition is labelled as a 'removable disk'; this approach to labelling truecrypt volumes differs from secure PEE device 1. As each truecrypt volume 'local disk' label only differs by the drive letter it may be difficult for a user to distinguish the VM volume and the data volume.

**Secure PEE Device 3 - Implementation on a Hardware Encrypted Flash Memory Device (IronKey)**

The IronKey could not be configured to achieve the target configuration model due to its Windows only based authentication feature. The IronKey authentication feature prevents any access to the IronKey until successful authentication; therefore neither could a bootable OS be installed nor was it possible to use Ubuntu to set up separate partitions for the VM, user generated data and swap space. Figure 8 models the IronKey secure PEE device configuration; essentially it is the standard IronKey containing a VM and separate directory structure for user generated data.

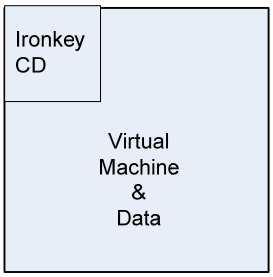


Figure 8: Configuration Model for Secure PEE Device 3

The following observations were made during the construction and testing of secure PEE device 3:

- Establishing the encryption keys (via the creation of the authentication credentials) both for the IronKey and the key recover capability (via the IronKey web site) was a user friendly seamless set of actions.
- The IronKey was tested on numerous host PCs running Windows XP and successfully authenticated (based on the insertion of the correct credentials).
- As would be expected the secure PEE VM and guest OS loaded from the IronKey correctly.

**Secure PEE Device 4 - Implementation on a Hardware Encrypted Hard Disk Drive (Pocket SDV)**

The Pocket SDV was able to be configured to fully satisfy the target configuration model. As the Pocket SDV is a USB HDD, Windows is able to detect and mount all FAT partitions on the HDD; therefore both the first and fourth partitions are accessible on the Pocket SDV secure PEE device. The following configuration could be achieved for the secure PEE device 4:

- The first and second partitions were set as Read-Only, to protect the integrity of the VM and bootable OS.
- A separate Read-Write data partition was created to separate and protect user generated data as required by the reference target configuration.
- As a further integrity protection measure, different user profiles were created with different access rights to partitions; a pre-boot profile denied access to partition 1 and a post-boot profile denied access to partition 2.

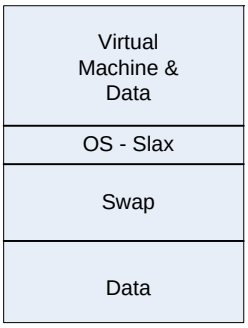


Figure 9: Configuration Model for Secure PEE Device 4

The following observations were made during the construction and testing of secure PEE device 4:

- The Pocket SDV System Administration Utility (SAU) allowed partitions, user profiles and access rights to be created. Although the SAU was an intuitive application it was extremely important to spend some time reading the detailed System Administration manual before configuration commenced.
- Testing of pre-boot authentication and subsequent booting of the secure PEE OS worked on all PCs tested.

- The post-boot PAA had to be installed on the host PC and also required Windows XP service pack 2 and .net 2.0 to be installed, proving less flexible than the other three secure PEE devices.
- Once installed the PAA consistently allowed authentication and loading of the VM.

## A QUALATATIVE ASSESSMENT OF THE SECURE PEE DEVICES

The required features and functionality of the four secure PEE devices was driven primarily by the capabilities required to satisfy the business scenario and the countermeasures required to address the identified threats. A set of security requirements were defined to allow a comparative analysis of the four secure PEE devices to be performed; the comparative analysis is given below. As would be expected the security requirements overlap with security aspects of the business scenario and threat countermeasures. Before performing the comparative analysis, the capabilities of the four secure PEE devices are summarised against the requirements of the business scenario and countermeasures required for the threats.

**Satisfying the Business Scenario**

The key aspects of the scenario are identified together with a summary of how the four secure PEEs provide appropriate functionality to address the scenario.

***A compact and easily transportable secure PEE device***: Three of the four secure PEEs were implemented on thumb drives. The Pocket SDV whilst compact is in a less transportable form.

***The individual will use the secure PEE device and its application on PCs for which no level of trust can be assumed, therefore the secure PEE device must provide an environment that can protect against malicious software***: Two secure PEE OS environments are provided. Where a user is concerned about the trustworthiness of a PC the bootable OS can be used. Read-Only partitions were configured (where possible) to protect the secure PEE against corruption by malicious software.

***It is important that the PEE can be reliably loaded into any PC***: The less secure, but easy to load VM based secure PEE provides a high level of reliability.

***The secure PEE device should also allow the application or individual to save data on to the secure PEE device***: Three of the four secure PEEs were able to be configured to provide a separate secure volume/partition for user generated data.

***Protecting the confidentiality of the data is important***: All of the secure PEEs enforced user authentication to deny access to unauthorised users coupled with storage encryption to protect the confidentiality of data.

***Stored data should be accessible by the individual (when the secure PEE is not being used) from a PC running Windows XP***: All four secure PEEs were configured to enable user generated data to be easily stored and retrieved.

***The secure PEE device, due to its compact nature, could easily be mislaid and therefore ensuring its contents can not be exploited is paramount***: User authentication and encryption protect the contents of the secure PEE device.

**Counter Measures to Address Threats**

***Threat - Malicious software (residing on a host PC) capturing user credentials and data***: Countermeasure - Using the bootable OS capability of the secure PEE device will ensure no malicious software is present to capture user credentials and data. If the VM capability of the secure PEE is used it is more difficult to prevent the capture of user credentials and data, therefore it is assumed a user will not use the VM if the host PC is considered to be likely to have been compromised by malicious software.

***Threat - Sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of an Internet transaction***: Countermeasure - The secure PEE device swap partition will prevent data remnants in the form of page files being written to the host PC HDD. It is assumed that a secure PEE OS and Internet based application which are considered to be trusted will have been hardened to prevent any temporary data being written to the host PC HDD.

***Threat - As a result of loss or theft, unauthorised access is gained to (sensitive) data held on an unsecured USB storage device that was used to store data generated during an Internet transaction***: Countermeasure - User authentication and encryption will prevent access by unauthorised personnel to user generated data following the loss or theft of a secure PEE device.

**Comparative Analysis Security Strength of Secure PEE**

Table 1 provides a comparative analysis of the security strength of the secure PEEs by presenting for each security requirement a statement of compliance for each secure PEE; with the strongest secure PEE highlighted.

| Requirements | Secure PEE Features that Address the Requirements | | | |
|---|---|---|---|---|
| | **Device 1: Std Flash** | **Device 2: U3** | **Device 3: IronKey** | **Device 4: SDV** |
| Only allow authorised users to load the resident OS/application(s) and access stored data | Post-boot authentication via truecrypt password. Only authorised users can gain access to the 2 truecrypt volumes; however access to non-encrypted storage space is possible. No pre-boot authentication therefore unauthorised users can gain access to bootable OS. | Post-boot authentication via truecrypt password. Only authorised users can gain access to the 2 truecrypt volumes; however access to non-encrypted storage space is possible, unless U3 password used. No pre-boot authentication therefore unauthorised users can gain access to bootable OS. | Post-boot authentication via IronKey password. No access can be gained to the device until successful authentication. No pre-boot authentication, however no bootable OS is available. | **Both strong pre-boot and post- boot authentication. No access can be gained to the device until successful authentication.** |
| Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks | Partitioning provides separation and isolation. Although truecrypt can mount a volume Read-Only, running a VM and OS inside a truecrypt Read-Only volume failed. Further analysis and testing of the Read-Only feature is required. | Partitioning provides separation and isolation. Although truecrypt can mount a volume Read-Only, running a VM and OS inside a truecrypt Read-Only volume failed. Further analysis and testing of the Read-Only feature is required. | No partitioning and no Read-Only mechanism. Once successful authentication has been achieved the device is open to any read and write access. | **Partitioning provides separation and isolation. Integrity can be preserved by the Read-Only capability. Read-Only and No-Access permissions can be set per partition per user profile, i.e. different profiles can have different access rights to the same partition.** |
| Preserve the confidentiality of any stored data from external attacks including theft of the device | User generated data is stored in a truecrypt encrypted volume. | User generated data is stored in a truecrypt encrypted volume. | User generated data is protected by fully hardware based encryption. | **User generated data is protected by fully hardware based encryption. Each partition has its own encryption key, if one is broken then only one partition at most is exposed.** |
| Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session | **By utilising the swap partition on the device no page files are written to the host PC.** | **By utilising the swap partition on the device no page files are written to the host PC.** | The IronKey can not be configured with a swap partition. | **By utilising the swap partition on the device no page files are written to the host PC.** |
| Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen | 256 bit AES truecrypt volumes will prevent acquisition; however software encryption is not as strong as hardware encryption[6]. | 256 bit AES truecrypt volumes will prevent acquisition; however software encryption is not as strong as hardware encryption. | **128 bit AES hardware based encryption will stop even the most sophisticated highly resourced forensic investigation.** | **128 bit AES hardware based encryption will stop even the most sophisticated highly resourced forensic investigation.** |
| Confirm both the identity of the user and the device before allowing an Internet transaction to occur | This requirement is considered out of scope in this paper. | | | |

Table 1

[6] Software encryption keys reside in the host PC RAM and therefore could be captured by a determined, highly skilled and well resourced forensic analyst (J Alex Haldermany et al 2008)

## CONCLUSION

Four secure PEEs were built and tested using a range of freeware and commercial off the shelf technologies. A brief summary of each secure PEE device is given together with areas that could be subject to further investigation.

*Secure PEE Device 1 - Implementation on a Standard Flash Memory Device:* Secure PEE 1 was the cheapest solution to construct, it provides a reasonable capability that addresses many of the security requirements, counters the threats and could be used under certain circumstances within the given business scenario. It would not be suitable as a turnkey solution that could be supplied to an organisation's customers to perform secure transactions. It is conceivable however that such a secure PEE device could be distributed to a 'controlled audience' for a specific application. For instance an organisation could distribute the secure PEE device to a certain group of employees with fixed operational instructions.

*Secure PEE Device 2 - Implementation on a U3 Flash Memory Device:* The secure PEE features and functionality of this device are identical to secure PEE device 1, but instead uses a U3 thumb drive as its platform which provides a strong titanium enclosure, the 'U3 chip' and a post-boot authentication mechanism. As per secure PEE device 1, the device is probably not suitable for a turnkey application; it would however provide a more robust platform than secure PEE device 1 for a 'controlled audience' application.

*Secure PEE Device 3 - Implementation on a Hardware Encrypted Flash Memory Device (IronKey):* The secure PEE built using the Ironkey had fewer features than the other secure PEE devices. It did provide a strong platform with hardware based encryption and authentication. Secure PEE device 3 did not provide the best platform for use in the business scenario presented.

*Secure PEE Device 4 - Implementation on a Hardware Encrypted Hard Disk Drive (Pocket SDV):* Secure PEE device 4 provided the best functionality to satisfy the security requirements and counter the threats. It would be the best secure PEE to use as a solution for the business scenario presented in this paper. The Pocket SDV does have a number of disadvantages, including its size, shape and weight compared to the other secure PEE devices considered, and it is also not as robust as platforms like the IronKey and U3.

The work presented in this paper was defined to be a contained and complete piece of research. However, a number of areas for further investigation arose during the research including:

- Determine if a VM and guest OS can be configured to execute within a true crypt Read-Only volume.
- Determine if truecrypt can be launched from the U3 CD partition as other applications have been configured to do (Al-Zarouni & Al-Harji 2007).

## REFERENCES:


Al-Zarouni, M. (2006). The Reality of Risks from Consented use of USB Devices. 5th Australian Digital Forensics Conference, Perth, Edith Cowan University.

Al-Zarouni, M. Al-Harji. H. (2007). A Proof of Concept Project for Utilising U3 Technology In Incident Response. 6th Australian Digital Forensics Conference, Perth, Edith Cowan University.

Bellard, F. (2008). "Qemu Open Source Processor Emulator." Retrieved May, 2008, from http://bellard.org/qemu/.

Brand, N. (2008). "The LiveCD List sponsored by FrozenTech." Retrieved April, 2008, from http://www.frozentech.com/content/livecd.php.

Chan, J. N., S. Moreland, D. Hon Hwang Shiping Chen Zic, J. CSIRO ICT Centre, Marsfield (2007). User-Controlled Collaborations in the Context of Trust Extended Environments. WETICE 2007. 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. .

CSIRO (2008). "Virtual Machines: An Initial Analysis of Threats and Remedial Actions."

Garfinkel T, P. B., Chow J, Rosenblum M, Boneh D (2003). Terra: A Virtual Machine Based Platform for Trusted Computing. Proc. 9th ACM Symposium on Operating Systems Principles SOSO'03.

Hannay, P., James, P. (2007). Pocket SDV with SDGuardian: A Secure & Forensically Safe Portable Execution Environment . 5th Australian Digital Forensics Conference, Perth, Edith Cowan University.

IronKey. (2008). "IronKey Technology." Retrieved May, 2008, from https://www.ironkey.com/technology.

J. Alex Haldermany, S. D. S., Nadia Heningery, William Clarksony, William Paulx, and A. J. F. Joseph A. Calandrinoy, Jacob Appelbaum, and Edward W. Felten (2008). Lest We Remember: Cold Boot Attacks on Encryption Keys. Proc. 2008 USENIX Security Symposium.

Kwan P, D. G. (2007). Practical Uses of Virtual Machines for Protection of Sensitive Data. Proc. 3rd Information Security Practice and Experience Conference (ISPEC).

PendriveLinux.com-Qemu (2008). "Qemu Persistent SLAX Linux tutorial." Retrieved May, 2008, from http://www.pendrivelinux.com/2007/04/02/qemu-persistent-slax-linux/.

PendriveLinux.com-Slax (2008). "SLAX USB flash drive installation using Windows." Retrieved April, 2008, from http://www.pendrivelinux.com/2006/09/20/all-in-one-usb-slaxzip/.

SanDisk. (2008). "What is a U3 Smart Drive." Retrieved May, 2008, from http://www.u3.com/smart/default.aspx.

SecureSystems. (2008, May 2008). "Portable SDVs." Retrieved May, 2008, from http://www.securesystems.com.au/pages/04_news/brochure_pdf/Portable-Specs.pdf.

Slax. (2008). "Slax - your pocket operating system." Retrieved May, 2008, from http://www.slax.org/.

Nepal S, Hon Hwang and David Moreland (2007). Trust Extension Device: Providing Mobility and Portability of Trust in Cooperative Information Systems Springer Berlin / Heidelberg

TrueCrypt. (2008). "TrueCrypt - Free Open Source On-The-Fly Encryption." Retrieved May, 2008, from http://www.truecrypt.org/.

TrueCrypt-Foundation. (2008). "TrueCrypt Users Guide." Version 5.1a. Retrieved May, 2008, from http://www.truecrypt.org/docs/.

Wielenge, D. (2008). "Slax Guide." Retrieved May, 2008, from http://www.geocities.com/slaxfansite.

## COPYRIGHT