Threat analysis of portable hack tools from USB storage devices and protection solutions

Conference Paper · July 2010

DOI: 10.1109/ICIET.2010.5625728 · Source: IEEE Xplore

4 authors, including:

Dung Vu Pham
University of Melbourne
7 PUBLICATIONS **66** CITATIONS

SEE PROFILE

Ali Syed
Charles Sturt University
20 PUBLICATIONS **195** CITATIONS

SEE PROFILE

Malka N. Halgamuge
University of Melbourne
159 PUBLICATIONS **1,132** CITATIONS

SEE PROFILE

Threat Analysis of Portable Hack Tools from USB Storage Devices and Protection Solutions

Dung V. Pham, Ali Syed, Azeem Mohammad
School of Computing and Mathematics
Charles Sturt University, Study Centre Melbourne
Victoria 3000, Australia
email: dung.pham@smeiss.com

Malka N. Halgamuge. Member, IEEE
Department of Civil and Environmental Engineering,
Department of Electrical and Electronics Engineering,
The University of Melbourne, Victoria 3010, Australia
email: malka.nisha@unimelb.edu.au

Abstract—**Information security risks associated with Universal Serial Bus (USB) devices have been a serious issue in corporate networks after the wide adoption of USB technologies in the computing industry in 2005. Recently, the U3 USB drives have been of great interest for attackers who want to utilize USB drives as their mobile hack tools. However, beside U3 technology, attackers also have another more flexible alternative, portable application or application virtualization, which allows a wide range of hack tools to be compiled into portable format and run from USB storage devices without requiring any USB specific platform such as U3. In this paper, we provide an investigation into hack tools on U3 platform and USB platform free portable hack tools, their working mechanism, and the compilation techniques. We also provide a general description of most dangerous hack tools with their payloads which can be compiled into portable format. Finally, our proposed solution is aimed at providing the most important and concise solutions for enterprise administrators to secure their systems from portable hack tools.**

Index Terms— USB, U3, Portable Application, Hack Tools

I. INTRODUCTION

Universal Serial Bus (USB) is a popular standard which has been widely adopted in the computing industry for the last few years for replacing serial and parallel ports thanks to number of advantages such as high data processing speed, hot swapping, plug-and-play (PnP), and self-power supplying to peripherals. This adoption allows a wide range of different electronic devices to connect to computers including mice, keyboards, PDAs, gamepads and joysticks, scanners, printers, digital cameras, personal media players, and most importantly flash drives and external hard drives via USB interface. However, the popularity of USB interface capable devices has resulted in the increasing risks to information security especially in corporate environments. Regardless of system administrators' efforts in system hardening such as least privilege practices and security enforcement by security tools, USB hack tools will still be a serious threat vector to corporate information security in the near future. In this paper, we will investigate a threat vector from USB storage devices which is commonly known as portable hack tools. Although the concept of USB based portable applications has been associated with U3 technologies since 2004, there does exist another more flexible solution for portable applications which does not require any USB platform to be able executed on. Therefore, an investigation on both U3 portable hacks tools and USB platform free portable hack tools together with their working mechanisms, common payloads, and the compilation techniques will be conducted. We also provide a complete solution for enterprise administrators to mitigate this threat vector.

II. PREVIOUS WORKS

A. USB based Hack Tools

Among attacks on host computers launched from USB storage devices, data theft has been the biggest concern about USB devices in corporate environments since 2005 when the USB 2.0 devices became popular. Data theft is normally conducted using various simple and tiny programs which are capable of silently downloading specific data files from host computers into USB drives [1]–[2]. In following years, 2006 and 2007, there was a substantial increase in the frequency and the level of complexity of USB-based software attacks on computers, especially networked computers. Ad hoc based hack tools automatically launched from USB drives were capable of doing many kinds of data manipulation on the computer systems from changing registry settings, creating backdoors, installing malicious codes, stealing confidential information, to even downloading the system page file from a running computer to USB drives [1]–[3]. Cryptography attacks were also common during the period with the support of USB drives and some small hack tools capable of exploiting operating systems' data encryption keys, Open SSH, and Apache HTTPS servers [4].

After USB 2.0 standard, the U3 revolution becoming popular in 2007 has made U3 drives ultimate hack tools. The applications installed on U3 drives can be executed without having to be installed on host computers. Attackers can simply craft their own U3 ISO images with necessary hack tools to replace the original U3 ISO images on U3 drives, and take the advantages of the technology to launch multi-payload attacks on the target computers [1]–[3].

In 2008, a utility was developed to allow attackers to manipulate the information on inserted USB devices stored in Windows registry. It was suggested that such a utility when used in combination with other malicious codes will create an additional protection layer for attackers who employ USB devices as attack tools [5]. Although the idea of manipulating

978-1-4244-8003-6/10/$26.00 ©2010 CROWN

registry by utilities or malware was not new, it did suggest another possibility of software attack using USB drives.

B. Solutions proposed in the previous researches

The solutions for secure use of USB technologies proposed in previous researches come in three categories: data access control, USB port access control, and security policies.

Data access control is probably the most interested, feasible and widely adopted solution of the three because it allows the use of USB devices while maintains definite security levels. The commonly proposed data access control solutions include disabling Autorun, limiting user privileges, encrypting the stored data on both communication ends, restricting access to vital data on critical servers, monitoring access to servers, and limiting the size of data transferable to USB drives [1].

USB port access control involves physically disabling USB ports, and USB port access control by firmware and operating system settings and software utilities. In some organizations, USB ports on computers are physically disabled by glue which is the last recommended solution. Disabling USB ports by Basic Input Output System (BIOS) settings, Windows registry, and Group Policy settings are other options. Many researchers recommend deploying third party utilities to apply USB port access privileges to specific users, user groups, and even USB device classes such as Palm, USB phone, and others [1]–[2].

Acceptable Use Policy (AUP) is commonly referred to as management solutions for USB security issues. AUP are normally implemented with security education and training programs to provide users with essential understanding on secure use of information systems, regulate users' actions and provide procedures for managing security incidents [2]. AUP are generally cost-effective management solutions which can be implemented in any corporate environment.

Generally, these solutions would require additional software license costs as well as additional maintenance costs. Moreover, permanently disabling USB ports may not be a good solution in different contexts. Our proposed solution targets for minimum IT costs while allowing secure use of USB ports.

III. U3 HACK TOOLS

A. U3 Revolution and U3 Hack Tools Revolutions

U3 is an open standard developed by SanDisk and M-Systems aimed at providing users with application mobility through U3 application platform whereby applications can be installed on and run from U3 drives from any host computer without requiring administrative rights. There is a small partition located at the beginning of a U3 drive marked as a CDFS (CD File System) partition. This partition is detected by Windows as a CD rather than a flash drive. Installed tools are self-contained applications run from the CDFS partition without requiring installation on the host computers, modifying the registry, or reserving system resources. Although Autorun functionality is disabled for USB drives on Windows 7, it is enabled still for the CDFS partition. Attackers can create their own ISO images with their hack tools and install the image in the CDFS partitions. When someone plugs in a U3 USB drive, Autoplay feature will trigger malware installation the host computers or attackers can run their own hack tools from U3 Launchpad at will. U3 technology is currently supported on all Windows platforms from Windows 2000 SP4 [6].

B. Working Mechanism

When a U3 drive is inserted, the device presents itself as two separate devices with two separate device class ID created in Windows registry. The first device is a *CDROM* labeled U3 System whose registry entry is located at *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTORCdRom&Ven_Manufacturer&Prod_U3_USBProductName_Micro&Rev_XXX* in which *Manufacturer* and *USBProductName* vary depending on the manufacturer and specific product. The registry key information for the U3 CDROM key is as in Figure 1 below.

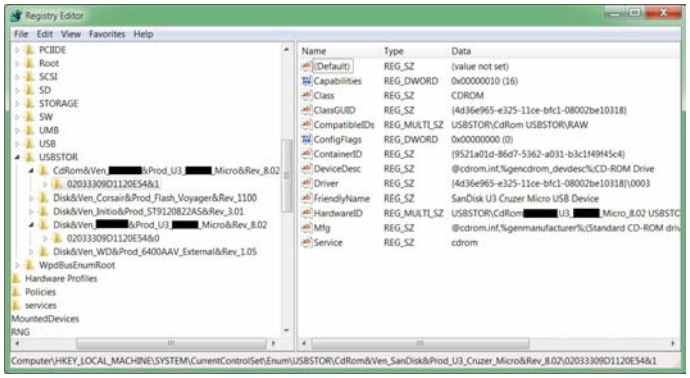


Figure 1. U3 CD-ROM Key

The second device is a *Removable Disk* whose Registry entry is located at *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_Manufacturer&USBProductName&Rev_XXX*. Similarly, *Manufacturer* and *USBProductName* vary depending on the manufacturer and specific product as shown in Figure 2 below.

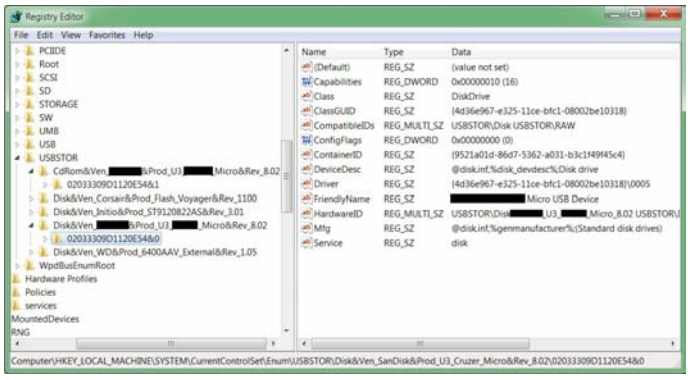


Figure 2. Removable Disk key for U3 USB drives

On successfully launching U3 Launchpad, a number of files are copied to a temporary folders named U3 at *RootDrive:\Users\UserName\AppData\Roaming\U3* on Windows Vista, Windows 7, and Windows 2008, or at *RootDrive:\Documents and Settings\UserName\ Application data\U3* on Windows 2000, XP, and 2003 which are the locations for the temporary working environment for U3 applications. Normally, these files include Cleanup.exe, LaunchPad.exe, Launchpad Removal.exe, U3AccessGrant.exe, U3LauncherSetup.msi, and a number of folders containing the manifest.u3i files. Among these files, manifest.u3i file is basically an XML file which describes the

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

U3 application properties (name, vendor, and functionalities), installation instructions, and action commands. Figure 3 below shows a snapshot of a manifest file for Winrar on a U3 drive.

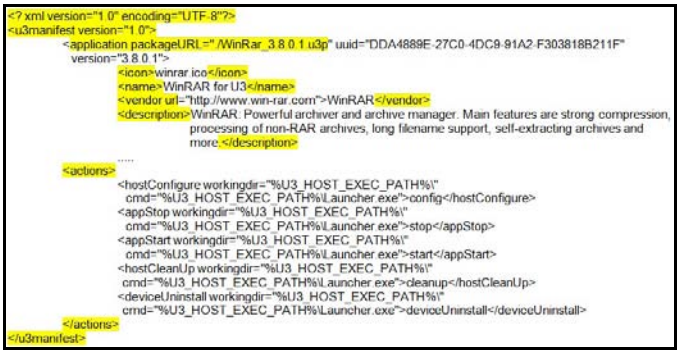


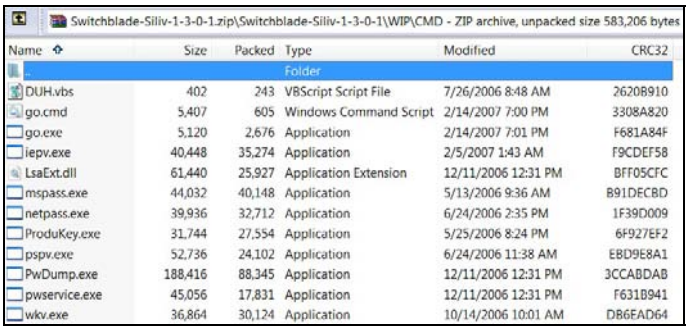
Figure 3. Menifest file sample

Another important file is Cleanup.exe which is used to clean up recorded information about the applications run from U3 drive on the host computer after the applications have been terminated. The Cleanup.exe is normally triggered on closing U3 applications. The basic job of Eleanup.exe is to delete the copy of U3 applications in the temporary directory on the host computer.

C. *The Most Common Payloads*

Attackers of this threat vector have a large range of choice on hack tools which can be deployed on U3 USB drives. The most widely-known hack tools running on U3 platform include USB Switchblade, U3 Incident Response Switchblade, USB Hacksaw, USB Pocket Knife, Nmap, Ethereal, Showtraf, TCPDump, Nemesis and John the Ripper, HTTP RAT, Anonymizer, and Data Recovery.

Switchblade provides the attackers with the ability to recover a lot of different important information from Windows systems such as password dumping (SAM, messenger clients, web browsers cache), LSA Secret, service, system information, and port scan. Switchblade comes in with two different versions developed by Hak5 and GonZor [7]. The payload of Switchblade actually comes from the combination payloads of many hack utilities bundled in the Switchblade suit such as *go, mspass, netpass, produkey, and pwdump* as shown in Figure 4.


Switchblade-Siliv-1-3-0-1.zip\Switchblade-Siliv-1-3-0-1\WIP\CMD - ZIP archive, unpacked size 583,206 bytes

Name	Size	Packed	Type	Modified	CRC32
..			Folder		
DUH.vbs	402	243	VBScript Script File	7/26/2006 8:48 AM	26208910
go.cmd	5,407	605	Windows Command Script	2/14/2007 7:00 PM	3308A820
go.exe	5,120	2,676	Application	2/14/2007 7:01 PM	F681A84F
iepv.exe	40,448	35,274	Application	2/5/2007 1:43 AM	F9CDEF58
LsaExt.dll	61,440	25,927	Application Extension	12/11/2006 12:31 PM	BFF05CFC
mspass.exe	44,032	40,148	Application	5/13/2006 9:36 AM	B91DECBD
netpass.exe	39,936	32,712	Application	6/24/2006 2:35 PM	1F39D009
ProduKey.exe	31,744	27,554	Application	5/25/2006 8:24 PM	6F927EF2
pspv.exe	52,736	24,102	Application	6/24/2006 11:38 AM	EBD9E8A1
PwDump.exe	188,416	88,345	Application	12/11/2006 12:31 PM	3CCABDAB
pwservice.exe	45,056	17,831	Application	12/11/2006 12:31 PM	F631B941
wkv.exe	36,864	30,124	Application	10/14/2006 10:01 AM	DB6EAD64

Figure 4. Component of Switchblade tool developed by Hak5.

The later version of Switchblade developed by GonZor is the more powerful version of the two which is equipped with the capability to overwrite the programs on the CD partition of a U3 USB drive. As this partition is read only, antivirus software can detect but cannot delete the installed applications.

In 2009, Hak5 developed a tool called U3 Incident Response Switchblade which is a derivative of Switchblade with the purpose of assisting security incident investigator in evidence gathering. Some of the remarkable features of this tool include account and group information gathering, networking information (IP, DNS cache, ARP table, NetBIOS, routing information, firewall state and rules) and services status retrieving [8].

D. *Tool Compilation*

Applications must be modified to become U3 compliant. Applications that do not use the Registry need only be wrapped in an XML-based U3 header, and simple applications can be changed to avoid the use of the Registry. However, programs that rely on the Registry and COM software components must communicate with U3 functions via U3 Device programming interface. Programmers use the U3 software development kit to develop U3 compliant software. Because U3 development kit is open to public, attackers can also create their own U3 hack tools in many circumstances. Moreover, with the support of U3 compilers such as Package Factory, a number of applications can be compiled to U3 applications and install on U3 drives. Some of the tools which have been successfully compiled to U3 format include disk management tools (Norton Partition Magic, Symantec Ghost), Registry tool (Clean Registry, Registry Mechanic), Anonymous Web surfing (Anonymizer, HTTP RAT), data recover (Data Recovery, Pro Data Recovery, Easy Recovery), Web browser (Firefox, Opera), Torrent client (eMule, FlashGet, Utorrent), Chat client (Pidgin, MSN Messenger, Yahoo Messenger), Password Tool, Script Editing tool (Notepad), Office Document (OpenOffice), ISO Tools (Virtual CD, Ultra ISO, Nero), Data Compression & Encryption (Winrar), and even antivirus software (Avast, Dr Web Cureit).

IV. USB PLATFORM FREE PORTABLE HACK TOOLS

A. *Portable Applications and Portable Hack Tools*

A portable application commonly known as *portable app* is a computer software program that is capable of executing independently without the need of being installed and configured on the system on which it is executed. Portable apps are designed to run on removable storage devices such as compact discs and USB storage devices with the purpose of providing application mobility for travelling users. Portable apps can be executed from specific operating system only where they are particularly designed for, such as Windows XP, Windows Vista or a specific distribution of Linux.

B. *Working Mechanism*

Portable apps are generally designed to be bundled with its configuration information and data files on the same portable storage devices where they locate and thus removing the need of being installed and configured on the host computers where they are executed. This means that on execution, portable apps do not need to update information on Windows registry or other INF files and all they do is to access their pre-configured configuration files bundled with them in the same portable media. In order to ensure the success of configuration and program file access on different computers, these portable files are designed to access their own files using relative naming convention paths. However, this is not always done by all portable apps and many of them utilize the same strategy as U3 applications which copy some of their configuration files to a

temporary location on the host computers and execute the program with configuration and data files from the temporary folders. On application shutdown, these files are cleaned from the host computer using the same technique as the U3 applications.

Another solution utilized by portable apps is called application virtualization which encapsulates the applications from the operating systems and other applications. This solution allows application to be complied into .MSI or .EXE format which have registry keys, DLLs, third-party libraries, and frameworks bundled with them in a single package and can be used to run anywhere. The process of compiling application into portable format involves the intercept of application execution on the runtime layer to obtain the registry call and file system calls before saving such configuration into a single package with necessary configuration files and supporting system files. This approach therefore does not create any modification on the application but make the application portable with all the necessary supporting files and pre-configured settings. Because each portable application is bundled with all necessary supporting files and settings, they can run independently from operating system and the other application on the host computer. In addition, because these applications' components are independent from system files and settings, they do not require system administrative right and privileges to be able successfully executed.

C. Tool Compilation

The most popular portable application compilers include VMware ThinApp, Landesk Application Virtualization, Ceedo, InstallFree, and Xenocode. The basic mechanism of these compilers is to observe the system state (check point) before and after an application installation and thus obtaining necessary information about system changes made by the application's installer. Based on the obtained information, the virtual registry and configuration files are created and bundled with the executable files of the application into a single package allowing application to be executed from any computer running similar operating system version.

Figure 5 and 6 show the compilation process for Internet Explorer Password Recovery Master which is done on a Windows XP machine by VMware ThinApp. The prescan process (Figure 5) capture the state of the computer including files on hard drives and Windows registry before the tool is installed to use as the system baseline which can be used to compare with the system state after the tool is installed.

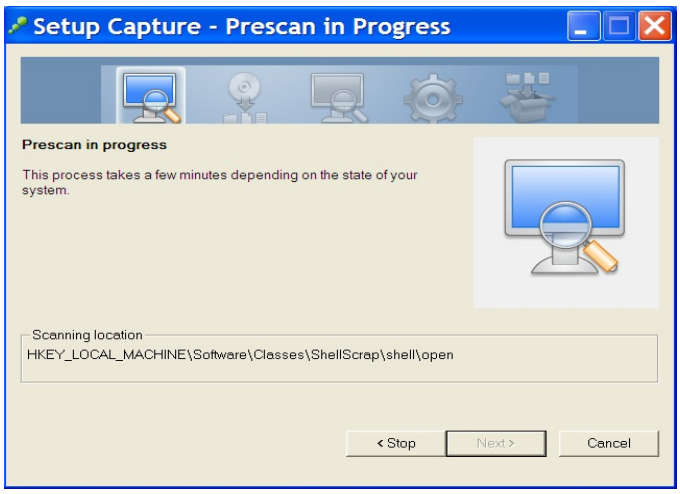


Figure 5. Prescan process capturing Windows registry state

The setup capture process (Figure 6) looks for changes made on the local computer after the installation of the tool. This includes new file and directory creation, update and modification made to Windows registry.

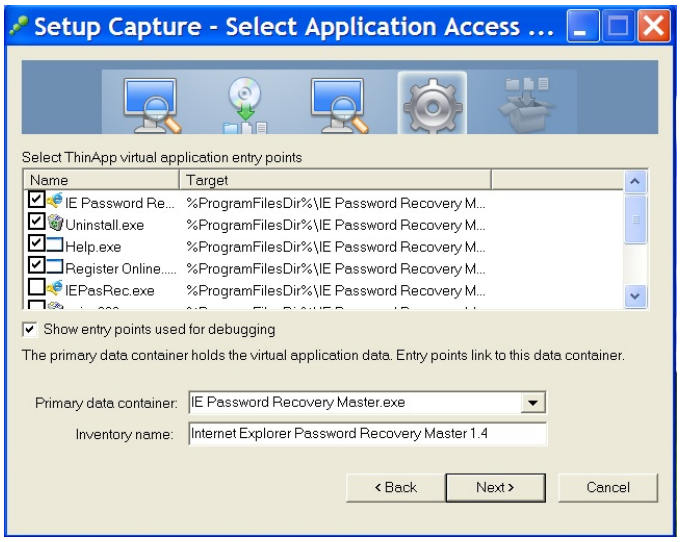


Figure 6. Setup capture detects changes made to the computer after the tool has been installed

Based on the obtained information, VMware ThinApp allows users to build up the working environment for the tools with all necessary supporting files and configured parameters allowing the tool to be able execute on other Windows XP machine without the need of installing the tool on those computer. Similarly, if the tool is installed on Windows Vista or Windows 7, after the compilation processes, the portable tools will be able to execute on Windows Vista or Windows 7 machine without the need of being installed again on the host computers.

Generally, this method of creating portable application allows many different tools to be compiled to portable format and run from USB storage devices or compact discs. There is practically no limit on the number of hack tools and payloads that attackers can choose to compile into portable format and run from USB drives. Moreover, the portability also makes the administrative rights on computer become less meaningful.

V. SOLUTION

The most easily recognized solution for these portable hack tools are malware scanners. Hack tools are normally detected by many malware scanners including commercial products on the market such as Symantec, Kaspersky, and AVG or free anti-malware solutions including Windows Defender, and Microsoft Security Essentials. However, there are many cases in which malware scanners failed to detect such tools.

The only radical solution for these portable hack tools requires the implementation of software restriction policies and the enforcement of trusted executable files. A trusted executable files must be a valid executable file signed with a non-expired digital signature by a trusted publisher or a reputable certificate authority such as VeriSign. Viruses and worms are often deployed with social engineering technique to trick users into triggering them. Therefore, allowing only trusted executable files on USB drive to be executed will help blocking any potentially dangerous code from executing even by users. Technically on Windows platform, this can be done

via either software restriction policies or AppLocker feature. While software restriction policies feature is first introduced in Windows XP and 2003, AppLocker is a new advanced feature first introduced in Windows 7 Ultimate/Enterprise, and Windows 2008 R2. The implementation of software restriction policies for executable files on USB drives are handled by certificates rules specifying a code-signing, software publisher certificate and path rules specifying a fully qualified path to the USB drives using wildcards to address all executable and script files. AppLocker is the more flexible option where we can specify only files belonging to trusted publishers can be executed from USB drives on specific user or users groups [6].

VI. CONCLUSION

In this paper, we have analyzed the working mechanisms of U3 portable applications and USB platform free portable applications. We have also described the compilation process and techniques for creating U3 and USB platform free portable applications which can be utilized by attackers to create their portable hack tools to execute from USB storage devices. The portability of these hack tools does not only make the administrative right become less meaningful but also make the forensic investigation process become more difficult as there are not many traces left on victim computers as these tools are executed with the support of files on USB drives. On the other hand, the only radical solution for such portable tools is software restriction policies with the enforcement of software policies or AppLocker with trusted executable files bundled with valid digital signature from trusted publishers.

REFERENCES

[1] M. Alzarouni, "The reality of risks from consented use of USB devices," in Proceedings of the 4th Australian Information Security Conference, 2006.

[2] M. Fabian, "Endpoint Security: Managing USB-based Removable Devices with the Advent of Portable Applications," in Information Security Curriculum Development Conference, 2007.

[3] S. Lee, A. Savoldi, S. Lee and J. Lim, "Password Recovery Using an Evidence Collection Tool and Countermeasures," in *Intelligent Information Hiding and Multimedia Signal Processing, Third International Conference, Volume 2,* 2007.

[4] K. Harrison and S. Xu, "Protecting Cryptographic Keys from Memory Disclosure Attacks," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks,* 2007.

[5] P. Thomas and A. Morris, "An Investigation into the Development of an Anti-forensic Tool to Obscure USB Flash Drive Device Information on a Windows XP Platform," in *Digital Forensics and Incident Analysis, Third International Annual Workshop*, 2008, pp.60 – 66.

[6] D. V. Pham, M. N. Halgamuge and A. Syed, P. Mendis, "Optimising Windows security features to prevent USB based software attacks", *The 28th PIERS*, Cambridge, USA, 5-8 July 2010 (accepted).

[7] Hak5, *USB Switchblade*. Hak5.org. <http://wiki.hak5.org/wiki/USB_ Switchblade>, 2008 (accessed November 10, 2009).

[8] Hak5, *U3 Incident Response Switchblade*. Hak5.org. <http://wiki.hak5. org/wiki/U3_Incident_Response_Switchblade>, 2009 (accessed November 10, 2009).