Windows Event Forensic Process

Quang Do, Ben Martini, Jonathan Looi, Yu Wang, Kim-Kwang Choo

Chapter 7

WINDOWS EVENT FORENSIC PROCESS

Quang Do, Ben Martini, Jonathan Looi, Yu Wang and Kim-Kwang Choo

Abstract Event logs provide an audit trail that records user events and activities on a computer and are a potential source of evidence in digital forensic investigations. This paper presents a Windows event forensic process (WinEFP) for analyzing Windows operating system event log files. The WinEFP covers a number of relevant events that are encountered in Windows forensics. As such, it provides practitioners with guidance on the use of Windows event logs in digital forensic investigations.

Keywords: Windows event forensic process, Windows event logs

1. Introduction

Microsoft Windows has been the most popular personal computer operating system for many years – as of August 2013, it had more than 90% of the personal computer market share [11]. This suggests that the majority of the personal computers seized in digital forensic investigations run Windows. To fully extract and interpret the wealth of data contained in a Windows environment requires an in-depth understanding of the operating system and, in the context of this work, event logs.

A key step in digital forensics is to gather evidence about an incident involving computer systems and their associated networks. In such circumstances, the expectation is that there has been some accumulation or retention of data on various system components that has to be identified, preserved and analyzed. This process can be documented, defined and used to obtain evidence of a crime or cyber incident [7].

Event log files provide digital forensic practitioners with a wealth of data describing the operations of computer systems. As such, they often contain valuable information that could connect particular user events or activities to specific times.

Windows event logs provide a range of descriptors to allow for the compilation of events into categories such as "informational" and "critical." Individual event IDs indicate specific types of events and recent Windows versions have separate event log files for various applications and services. Despite these filtering options, it can be difficult for digital forensic practitioners to locate events pertinent to their investigations from among the large volume of stored events. The problem is even more acute for practitioners with limited expertise related to event logs. This paper attempts to address this gap by providing a summary of key events that are useful in Windows forensic investigations. The key events summary is used to derive a process that digital forensic practitioners can adapt and follow to obtain useful evidentiary data from Windows event logs.

2. Related Work

Several researchers have noted that Windows event logs contain a large amount of digital evidence. Kent, *et al.* [5] have shown that various types of logs pertaining to the operating system and applications are useful in post-incident response. While Kent and colleagues provide high-level guidance on the use of event logs in digital forensics across a range of operating systems, other researchers focus specifically on Windows event logs. For example, Ibrahim, *et al.* [4] discuss the potential use of Windows event logs as sources of evidence in cyber crime investigations. They found a number of useful data items in their simulated attack, which they assert could be used as evidence in court. In particular, they analyzed login events and network data contained in event logs. Event logs are often a valuable source of data that can link an individual or remote device to a specific event. For example, if a login event is recorded and the system time is changed shortly after the login, this can indicate that the individual who logged in changed the time on the personal computer.

Marrington, *et al.* [6] describe a system for profiling computers to detect activities that are of interest to forensic investigators. While their profiling system uses a number of data sources, event logs feature significantly in their proof-of-concept system. Their study also demonstrates the effectiveness of event logs in augmenting other digital evidence to demonstrate clear usage patterns.

Hashim and Sutherland [3] focus on event logs in their discussion of Windows artifacts. They cover a range of logs with potential forensic value, such as application, security and system event log files, including login, application usage and policy audit logs.

Table 1. Source computer specifications.

Computer	Operating System	CPU (Intel)	Primary Memory	Secondary Memory
1. Desktop	Win 7	Core i7 2600k	8 GB	120 GB SSD
2. Laptop	Win 7	Core i5-430UM	4 GB	60 GB SSD
3. Desktop	Win 7	Core i5 2500k	4 GB	1 TB HDD
4. Laptop	Win 7	Core i5-430UM	4 GB	500 GB HDD
5. Desktop	Win 7	Core i5 2500k	4 GB	1 TB HDD
6. Desktop	Win 7	Core i5 2500k	4 GB	1 TB HDD
7. Laptop	Win 7	Core i7-3470QM	16 GB	2 TB HDD
8. Desktop	Win 7/Win 8	Core i5 2500k	8 GB	2 TB HDD
9. Desktop	Win 7	Core i5 2500k	4 GB	500 GB HDD
10. VM	Server 2008 R2	Core 2 Quad Q9400	4 GB	40 GB HDD
11. VM	Win 7	Core 2 Quad Q9400	1.5 GB	60 GB HDD

Murphey [10] conducted technical research into the format of NT5 Windows event logs (used in Windows XP and Windows Server 2003 systems) and devised an automated method for forensic extraction (including recovery and repair) of the event logs. Schuster [12] provides technical insights into the newer Windows XML event log format introduced with Windows Vista. Unfortunately, aside from this work, there have been relatively few papers published that discuss specific aspects of Windows event logs for the purpose of forensic analysis.

The research efforts mentioned above provide invaluable knowledge and guidance on the use of event logs and log data in forensic investigations. However, because of the large quantities of data contained in the logs, it can be challenging and time consuming for a forensic practitioner to discover exactly where the evidence resides. To ensure the most effective use of event logs in forensic investigations, practitioners need a well-defined process to follow, more so when they have limited technical expertise.

3. Experimental Setup

Our experiments sampled event logs from several source computers; the specifications of the computers are listed in Table 1. The computers ranged from systems with low processing power used for recreation and simple office work (e.g., Internet browsing, social media browsing and low-intensity document and spreadsheet editing) to systems used for video gaming and other resource-intensive tasks (e.g., hobbyist programming).

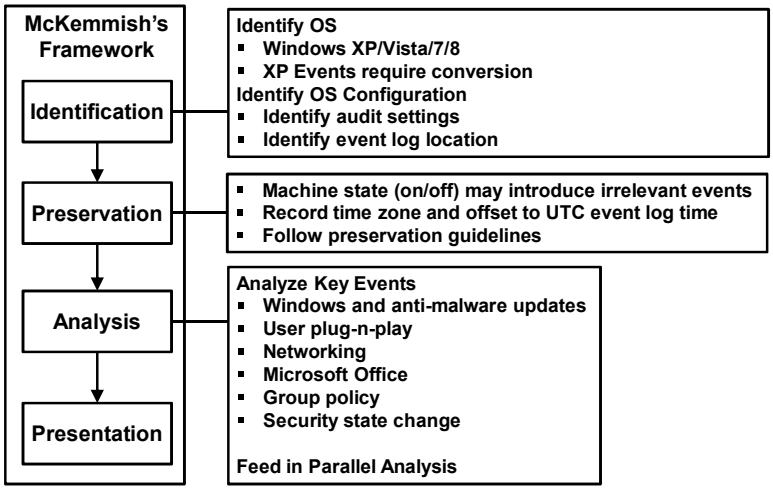

Figure 1. Proposed Windows event forensic process.

The laptops used in our experiments had wired and wireless network connections. The desktop computers only had wired connections. Virtual machines (VMs) were also provisioned to investigate auditing and group policies. Event log files collected over a two-month period – from November 1, 2012 to December 31, 2012 – were analyzed. This enabled commonly-occurring events to be discovered without having to sort through massive amounts of log data.

4. Proposed Windows Event Forensic Process

Current digital forensic frameworks do not discuss the analysis of event logs in detail. For example, the McKemmish framework [7] focuses on physical media such as hard drives and DVDs; there is little emphasis on operating system and configuration specifications, which are crucial to a Windows event log forensic process. Kent, *et al.* [5] have provided guidelines related to operating system event logs, but they do not discuss specific operating systems (which have different events and, depending on the configuration of an operating system, different information would be present in the logs). Thus, we present the Windows event log forensic process (WinEFP) to specifically support forensic examinations of event logs and log data stored in Windows systems.

The proposed WinEFP (Figure 1) fits within the McKemmish framework [7]. The four key elements of the framework are: (i) identification; (ii) preservation; (iii) analysis; and (iv) presentation of digital evidence.

Using the McKemmish framework renders the event log forensic process straightforward for even an inexperienced practitioner to follow.

4.1 Identification

The goal of the identification step in the McKemmish framework is to determine the digital evidence that is present along with its type and format. The location and format of the evidence are crucial because they determine the recovery methods. The WinEFP identification step focuses on identifying and understanding relevant event data from Windows machines.

Depending on the version of Windows installed on the system under investigation, the number and types of events will differ. In fact, the events logged by a Windows XP machine may be incompatible with an event log analysis tool designed for Windows 8. For example, Event ID 551 on a Windows XP machine refers to a logoff event; the Windows Vista/7/8 equivalent is Event ID 4647. Windows XP events can be converted to Vista events by adding 4096 to the Event ID [1]. Windows versions since Vista include a number of new events that are not logged by Windows XP systems. Windows Server editions have larger numbers and types of events. Thus, the exact version of the Windows system must be considered very carefully when developing a digital forensic process centered on event logs.

By default, a Windows system is set to log a limited number of events, but it can be modified to include actions such as file deletions and changes. Windows event logs are stored in a binary XML format that is unreadable by a text editor. However, the included Windows Event Viewer is able to read the logs and convert them to plaintext XML. The default location of Windows event logs is typically `C:\Windows\System32\winevt\Logs`. This can be changed by a user by modifying the File value of the following registry keys in `HKEY_LOCAL_MACHINE (HKLM)` on the local machine:

- **Application Events:** `SYSTEM\CurrentControlSet\services\eventlog\Application`
- **Hardware Events:** `SYSTEM\CurrentControlSet\services\eventlog\HardwareEvents`
- **Security Events:** `SYSTEM\CurrentControlSet\services\eventlog\Security`
- **System Events:** `SYSTEM\CurrentControlSet\services\eventlog\System`

We found that other event "channels" only generate File keys when their paths are changed from the default. When a custom path is used, a key is generated at the registry location: `HKLN\Microsoft\Windows\CurrentVersion\WINEVT\Channels\[logname]` (e.g., `Microsoft-Windows-Audio\CaptureMonitor`).

4.2 Preservation

The preservation step seeks to keep alterations to the digital evidence to a minimum. The event logs on a standard Windows machine are generally contained in just a few files, so it is generally not necessary to alter any data. However, because the event logging tools that come with Windows systems can remove and even completely clear data, care must be taken when retrieving event logs.

A running Windows machine logs events almost continuously. Therefore, if a forensic practitioner were to use the machine, new events would be added to the event log that could contaminate the digital evidence. Because the event log files are non-volatile [5], the computer should be powered down as soon as possible (after obtaining all the volatile data) to reduce evidence contamination. All the actions taken by the practitioner upon accessing the machine should be recorded thoroughly.

Digital forensic practitioners often make use of timestamps in investigations. Our research shows that timestamps in event logs are recorded in Coordinated Universal Time or UTC. This means that the current time zone of the machine must be used to compute the local time on the computer when the event occurred.

If a Windows machine is in a powered down state, it is possible to retrieve the Windows event log files without adding additional events by collecting the log files from the hard drive using an external machine. The log files typically reside in the `C:\Windows\System32\winevt\Logs` directory. However, as mentioned above, the Windows registry can be modified to change the event log locations.

4.3 Analysis

The analysis step in the McKemmish framework contributes specific knowledge about Windows system events that are of interest to a digital forensic practitioner.

Key Events. Locating digital evidence in a Windows event log file requires an understanding of Windows events and knowing what to look for. Our WinEFP attempts to describe key Windows event logs (`.evtx` in Windows Vista onwards) and their associated Event IDs.

Windows and Anti-Malware Update Events. When enabled, a Windows 7 system records details of updates applied by the Windows update service and every update to the Microsoft Security Essentials anti-malware software. Each event in this log provides details of the contents of an update and the time when the event was generated. This information and the conclusions derived from it are potentially interesting from a forensic standpoint. For example, the event timestamp and the details of the installed update can help a forensic practitioner determine if the system in question was secure or vulnerable to specific security threats during a particular period of time.

The events covered in this section were sourced from the System Event Log. Events that provide information regarding Windows updates have the "Windows Update Client" as their source and the events that indicate Microsoft Security Essentials updates have "Microsoft Anti-Malware" as their source.

Event 19 is the most common event that is used to ascertain installed updates on Windows 7 machines. This event is generated when an update is applied to the system; the event description provides details about the update file that generated the event. Often, Windows will download multiple update files in a batch; this generates an instance of Event 18. The event provides a description that lists all the update files that have been downloaded and are ready for installation. Therefore, the relationship between Event 18 and Event 19 is one to many: each update file mentioned in the Event 18 description generates an instance of Event 19 when it is installed. Event 22 is generated when the installation of a Windows update requires the system to be restarted.

Two events are of interest when determining the update status of Microsoft Security Essentials, namely Event 2000 and Event 2002. Event 2000 is logged when the Microsoft anti-malware signature database has been updated; the event provides the version numbers of the new and previous signatures. When coupled with the event timestamp, it is possible to determine periods of vulnerability where the installed anti-malware was not updated to a particular signature version. Event 2000 also provides the version number of the anti-malware engine in operation at the time of the signature update.

Event 2002 is generated when the Microsoft anti-malware engine is updated. We discovered that this occurs less frequently than signature updates – when reviewing logs collected over a two-month period, there were 42 times more Event 2000 instances on the average compared with Event 2002 instances. Much like Event 2000, the description of Event 2002 provides the version numbers of the previous security engine and the updated engine.

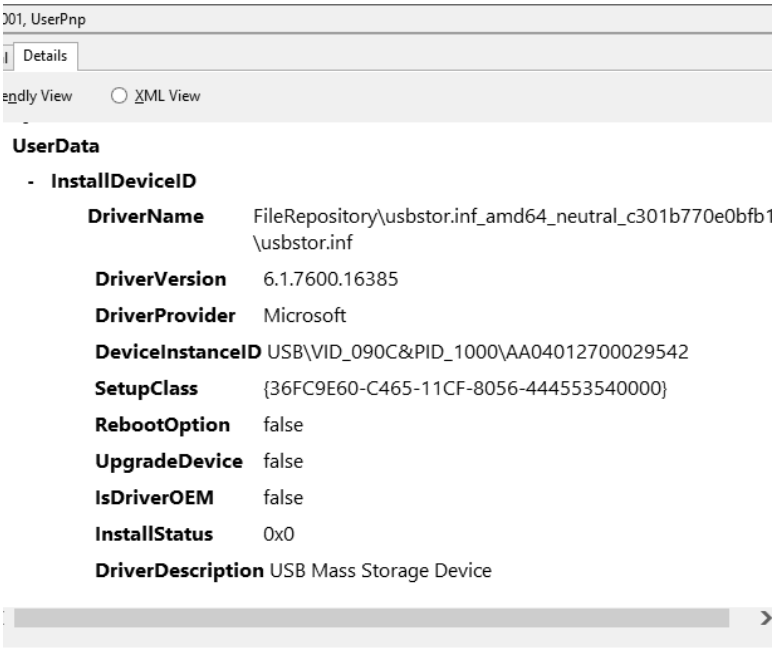

Figure 2. Example of Event 20001 (involving a USB device).

User Plug-n-Play Events. The User Plug-n-Play Device Events located in the System Event Log show USB/PCI connections to the computer. An event is triggered when a driver is installed or updated. Events that provide information about a hardware installation and driver have UserPnp as their source. The device installation and update event is 20001 and the service installation and update event is 20003.

Data provided by Event 20001 includes the DeviceInstanceID, driver information and setup class. The DeviceInstanceID is a unique identifier for each device; it is a slash-delimited string. Figure 2 shows an event recorded after the insertion of a USB storage device. A unique identifier is recorded in the DeviceInstanceID. The first field (USB in this example) represents the type of connection. VID represents the vendor ID (USB vendor IDs are recorded at `www.usb.org`) and PID represents the physical interface device class number (see `www.usb.org/developers/devclass_docs/pid1_01.pdf`). This is followed by the device identifier (`AA04012700029542` in the example).

Networking Events. An attempt to connect to a wireless network results in Event 8000 being logged in WLAN-AutoConfig, which also stores the SSID of the wireless connection. Event 8001 is logged on a

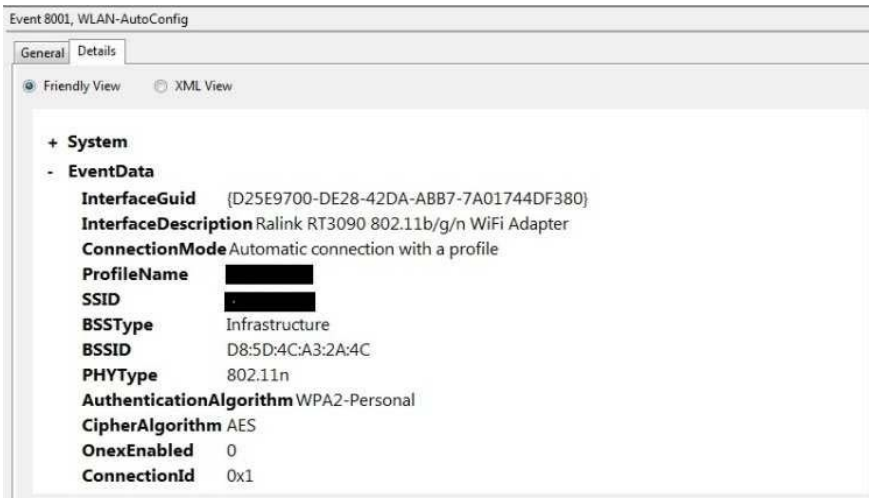

Figure 3. Example of Event 8001.

connection to the wireless network (Figure 3). Event 8001 contains information about the network interface, connection SSID, connection BSSID (which uniquely identifies the access point), wireless connection type (802.11a/b/g/n/ac) as well as the authentication type (WEP, WPA, WPA2, etc.) and cipher type (AES, TKIP, etc.). Event 8003 is logged on a disconnection from a wireless connection. In the case of laptops and mobile devices, interactions with wireless access points could be used by a practitioner to estimate the locations of the devices [2].

Because all events are timestamped, both the location and time are known. Event 11001 is logged when a wired connection succeeds; it provides the unique MAC address of the network interface and the MAC address of the peer. When a wired connection requires user authentication, Event 12012 is also logged. This event saves user login information, including the domain, username and computer name.

Microsoft Office Events. When a Microsoft Office application is installed, the Windows event log files contain a separate entry for Microsoft Office Alerts. The entry contains the events that are generated when an Office program displays a dialog box informing the user of an alert.

From a digital forensic standpoint, the events in the Microsoft Office Alerts log are useful for determining when documents were created, modified and opened, and their file names and types. For example, an event is logged when an Office application displays a dialog box that prompts the user to save a Microsoft Office file. This event records the details of the save prompt, including the file name.

Figure 4. Microsoft Office Alert event generated by a prompt to save a modified file.

All events in the Microsoft Office Alerts log have Event ID 300 and the same event source of "Microsoft Office 14 Alerts" (for Microsoft Office 2010). These events only differ in their descriptions. This leads to increased difficulty in finding specific types of events in the Microsoft Office Alerts log. For example, if a digital forensic practitioner is interested in finding all the events generated when a user was prompted to save a modified Office file, it is not possible to filter the list of events by searching for a unique combination of event ID and source.

All event descriptions in the Microsoft Office Alert log have the same general structure. The first line of the event description is the name of the Office application that generated the event. The second line is the literal of the dialog box that generated the event. The final four lines contain four parameters.

The most common event in the Microsoft Office Alerts log is typically the modification of an Office file. This event provides information about the name of the Office application file that a user opened and modified, along with the event timestamp. From among the event logs sampled in this research, it is the only event whose description provides the name of the file that the user was editing.

Every Office application generates its own version of this event. The descriptions differ in the first line (i.e., application name) and in the first parameter (i.e., unique numerical value based on the event description and application). In the case of Microsoft Word 2010, the unique first parameter value is 200054 (see Figure 4), for Microsoft Excel 2010 it is 100216 and for Microsoft PowerPoint 2010 it is 400072.

Group Policy and Auditing Events. Windows 7 Professional edition and higher and Windows Server editions allow the use of a group policy to control and administer user accounts in corporate environments [9]. Depending on the network group policy, it may be important in a digital forensic investigation to check the configuration, e.g., when the group policy is configured so that certain events are forwarded to another Windows device. Thus, if the event logs on a given machine are wiped, the logs may be obtained from the other Windows device.

Table 2. Windows Client editions vs. Windows Server editions.

Configuration(s)	Client Windows	Server Windows
Audit Account Login Events		
Credential Validation; Kerberos Authentication Service; Other Account Login Events	No Auditing	Success
Kerberos Service Ticket Operations	No Auditing	No Auditing
Audit Account Management		
Application Group Management; Distribution Group Management	No Auditing	No Auditing
Computer Account Management	No Auditing	Success
Security Group Management; User Account Management	Success	Success
Audit Directory Service Access		
Detailed Directory Service Replication; Directory Service Change; Directory Service Replication	No Auditing	No Auditing
Directory Service Access	No Auditing	Success
Audit Login Events		
Account Lockout; Logoff; Special Login	Success	Success
IPSec Extended/Main/Quick Mode; Other Login/Logoff Events	No Auditing	No Auditing
Login	Success	Success, Failure
Network Policy Server	Success, Failure	Success, Failure
Audit Object Access		
Application Generation; Certification Services; Detailed File Share; File Share; File System; Filtering Platform Connection/Packet Drop; Handle Manipulation; Kernel Object; Registry; SAM; Other Object Access Events	No Auditing	No Auditing
Audit Policy Change		
Audit/Authentication Policy Change	Success	Success
Authorization Policy Change; Filtering Platform Policy Change; MPSSVC Rule-Level Policy Change; Other Policy Change Events	No Auditing	No Auditing
Audit Privilege Use		
Non-Sensitive Privilege Use; Other Privilege Use Events; Sensitive Privilege Use	No Auditing	No Auditing
Audit Process Tracking		
DPAPI Activity; Process Creation; Process Termination; RPC Events	No Auditing	No Auditing
Audit System Events		
IPSec Driver; Security System Extension	No Auditing	No Auditing
Other System Events; System Integrity	Success, Failure	Success, Failure
Security State Change	Success	Success

Another example is the audit policy setting in a group policy object, which specifies if certain types of events should be logged as success or failure, or both. In most cases, the default setting for audit policies is "No Auditing." Table 2 lists each audit policy category and the

events audited in each category for Windows Client editions and Windows Server editions (default configurations).

Security State Change Events. Windows Vista and newer editions have an event category known as Security State Change events. Event 4608 occurs when a system starts up and Event 4616 occurs when the system time is changed, either by the user or when Windows communicates with a time synchronization server.

Event 4609, the final event in this category, occurs when the system is shut down [8]. This event category is listed under "Audit System Events" in the group policy settings.

Event Summary. The following is a summary of the key events discussed above:

- **Windows and Anti-Malware Events:**
  - Event ID 19 is logged when an update is applied to the system.
  - Event ID 18 is logged when a batch Windows update is downloaded.
  - Event ID 22 is logged when a restart is required for an update installation.
  - Event ID 2000 is logged when Microsoft anti-malware signatures are updated.
  - Event ID 2002 is logged when the Microsoft anti-malware engine is updated.
- **User Plug-n-Play Events:**
  - Event ID 20001 is logged when a driver is installed/updated.
  - Event ID 20003 is logged when a service is installed/updated.
- **Networking Events:**
  - Event ID 8000 is logged when a wireless network connection is attempted.
  - Event ID 8001 is logged on a successful wireless network connection.
  - Event ID 8003 is logged on a disconnection from a wireless network.
  - Event ID 11001 is logged on a successful wired network connection.
  - Event ID 12012 is logged when a wired connection requires user authentication.
- **Microsoft Office Events:**
  - Event ID 300 is logged when an Office application generates a dialog box; the contents of this event differ based on the Office application that generates it.
- **Auditing Events:**
  - Refer to Table 2.

- **Security State Change Events:**
  - Event ID 4608 is logged when Windows starts up.
  - Event ID 4609 is logged when Windows shuts down.
  - Event ID 4616 is logged when the system time is modified.

4.4 Presentation

The final presentation step in the McKemmish framework involves presenting the evidence in a court of law [7]. In the case of Windows event logs, presenting textual information may not be ideal because demonstrating computer usage over time may involve many pages of text. It is advisable to present the information in the form of graphs, tables and other graphical visualization schemes.

5. Conclusions

Windows is the dominant operating system in consumer and corporate computing environments. Unless a Windows user has manually disabled the event logging service, the proposed Windows event forensic process (WinEFP) can be applied to practically every forensic investigation involving a Windows personal computer. In developing WinEFP, a number of forensically-relevant Windows event log entries were identified and cataloged. Our future research will extend the process to address other Windows versions as well as other operating systems.

References

[1] Dorian Software Blog, 4094 Security Events Lane (`eventlogs.blogspot.com.au/2007/04/4096-security-events-lane.html`), April 13, 2007.

[2] X. Fu, N. Zhang, A. Pingley, W. Yu, J. Wang and W. Zhao, The digital marauder's map: A WiFi forensic positioning tool, *IEEE Transactions on Mobile Computing*, vol. 11(3), pp. 377–389, 2012.

[3] N. Hashim and I. Sutherland, An architecture for the forensic analysis of Windows system artifacts, *Proceedings of the Second International ICST Conference on Digital Forensics and Cyber Crime*, pp. 120–128, 2011.

[4] N. Ibrahim, A. Al-Nemrat, H. Jahankhani and R. Bashroush, Sufficiency of Windows event log as evidence in digital forensics, *Proceedings of the Seventh International Conference on Global Security, Safety and Sustainability and the Fourth Conference on e-Democracy*, pp. 253–262, 2011.

[5] K. Kent, S. Chevalier, T. Grance and H. Dang, Guide to Integrating Forensic Techniques into Incident Response, NIST Special Publication 800-86, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[6] A. Marrington, G. Mohay, A. Clark and H. Morarji, Event-based computer profiling for the forensic reconstruction of computer activity, *Proceedings of the AusCERT Asia Pacific Information Technology Security Conference*, pp. 71–87, 2007.

[7] R. McKemmish, What is forensic computing? *Trends and Issues in Crime and Criminal Justice*, vol. 118, pp. 1–6, 1999.

[8] Microsoft, Audit Security State Change, Microsoft, Redmond, Washington (`technet.microsoft.com/en-us/library/dn311493.aspx`), 2013.

[9] Microsoft, Windows 7 Product Guide, Microsoft, Redmond, Washington (`www.microsoft.com/en-us/download/details.aspx?id=4984`), 2014.

[10] R. Murphey, Automated Windows event log forensics, *Digital Investigation*, vol. 4(S), pp. S92–S100, 2007.

[11] J. Newman, Windows 8 grabs more market share, but so do older versions, *PC World*, August 1, 2013.

[12] A. Schuster, Introducing the Microsoft Vista event log file format, *Digital Investigation*, vol. 4(S), pp. S65–S72, 2007.