available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

Applying a forensic approach to incident response, network investigation and system administration using Digital Evidence Bags

*Philip Turner**

QinetiQ, Digital Investigation Services, Trusted Information Management Department, Malvern Technology Centre, G Building, Room 311, St. Andrews Road, Malvern, Worcestershire WR14 3PS, UK

ARTICLE INFO

Article history:
Received 22 December 2006
Accepted 7 January 2007

Keywords:
Digital forensics
Digital Evidence Bags
Digital investigation
Network investigation
System administration
Incident response

ABSTRACT

This paper questions the current approach to forensic incident response and network investigations. Although claiming to be 'forensic' in nature it shows that the basic processes and mechanisms used in traditional computer forensics are rarely applied in the live incident investigation arena. This paper demonstrates how the newly proposed Digital Evidence Bag (DEB) storage format can be applied to a dynamic environment. A DEB is a universal container for digital evidence from any source. It allows the provenance to be recorded and continuity to be maintained throughout the life of the investigation. With a small amount of forethought a forensically rigorous approach can be applied to incident response, network investigations and system administration with minimal overhead.

1. Introduction

Many organisations and companies have their own computer incident response teams and network investigators. These teams are often assembled at very short notice when an incident occurs or anomalous system behaviour is detected. The problem with this is that within a very short time period, personnel with the necessary technical skills have to start working together to diagnose the behaviour, problem or attack.

At the onset of the incident all that may be known is that something on a system or network is not working as expected and the whole aim is usually to restore the service, capability or system to its normal level. In this first instance the System Administrator (SA) would probably be either the first person to detect a problem or the first person to start examining the system as the result of a helpdesk call from a user. What the eventual outcome may be is usually far from the thoughts of the SA or examiner.

In most cases the problem may be benign or user error but in the instance where a more serious problem is discovered a more comprehensive and rigorous examination or investigation may follow. This would especially be true if a system had failed completely, or if it was discovered that information had been lost or stolen.

The problem with this approach is that the SA may already have inadvertently modified a system or lost the opportunity to accurately record the system state that was discovered. This puts them in a vulnerable position should the finger of blame be pointed at them at a later date. In addition, it may

* *Tel.*: +44 01684 895777; fax: +44 01684 894365.
E-mail address: pbturner@qinetiq.com
1742-2876/$ – see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.diin.2007.01.002

be the case that the SA cannot get repeatable results when they realise that there is a serious problem.

The aim of this paper is to raise awareness of the potential problems that could be encountered and unite a thorough forensic approach with the incident response procedure or network investigation procedure. The two procedures are not mutually exclusive and it is not possible to 'bolt on' a forensically rigorous approach as an after thought. The forensic aspect should be an integral part of any system administration role that is responsible for operating any business critical or business operational system.

In this paper the term 'System Administrator' (SA) is used to cover the processes, tasks or operations performed by the system administrator, system operator, incident investigator, security administrator, network investigator… anyone who may have any role in determining the problem, cause or effect of any abnormal or unusual system behaviour.

2. Common tools

The SA usually has a vast range of tools available at their disposal to monitor system performance, determine system configuration or to fault find system problems. The majority of these tools and utilities are very focussed in the function they perform and the information they provide. Many of these tools are console applications and run as command line utilities. Furthermore, they are often packaged with the operating system.

The problem with these tools is they are designed to provide information, but are not designed to provide any form of integrity assurance or record when those utilities were executed. From a forensic perspective they provide no audit record about the timestamp or actions taken, or results returned from running those utilities.

Admittedly the SA, if they had the forethought, could choose to pipe the output to a log file but this still has no mechanism to assure the integrity of the data output. Furthermore, the SA would rarely keep hand written notes of the actions taken, or results obtained whilst trying to diagnose and examine a system.

This typical scenario results in the SA being unable to justify or even demonstrate to colleagues the system state as they found it, never mind being able to assure that they were not the cause or a contributor to the problem or incident.

There are many tools that may be used to diagnose system behaviour (Sorenson, 2003a,b; Carvey, 2001; Microsoft Windows XP). In addition to this there are toolkits available with many of these applications already compiled into a compendium (Fense tool; Foundstone tools).

What is really required here is a way of uniting the rigorous approach required from the forensic dimension with the flexibility for each SA to be able to execute their favourite tools and utilities. The proposed solution is the Digital Evidence Bag (DEB) (Turner, 2005; Turner, 2005–2006).

3. Digital Evidence Bags

A DEB is a universal container format for digital information from any source. It allows the provenance of digital information to be recorded and continuity to be maintained throughout the life of the investigation.

The main components of a DEB are the tag, index and bag files (Fig. 1). The index and bag files together are known as an Evidence Unit (EU). It is the EU coupled with the customisable index definition that provides the tremendous flexibility afforded by the DEB framework.

DEBs were originally conceived to be used in the traditional role of static digital forensic investigations. In this role they permit more advanced data capture techniques to be supported, for example selective and intelligent imaging methodologies (Turner, 2006). However, they can also be used to store forensic images of command line utility output, digital media, memory dumps, network packet captures and all associated meta-data.

This paper demonstrates how the DEB framework can also be used in a more dynamic environment, i.e. that of incident response, system administration and network forensics. The data captured being in a compatible format with that obtained in the static environment and thus permitting traditional forensic analysis tools and techniques to be utilised.

When an investigation, or enquiry is commenced then each investigator would use the DEB forensic incident response tool. When they commence their examination of the system a DEB is created which automatically logs the current date and time. The DEB forensic incident response tool allows files on the system to be captured into a single or multiple EU. For example, each application's log files could be captured to separate EUs and configuration or registry files could be captured to yet another EU.

In addition to this the DEB forensic incident response tool allows command line applications to be executed from a special dialogue box. When the command is executed the output from each command is captured in the DEB together with an integrity hash of the data and a timestamp of when the command commenced and completed. When all the relevant system information has been captured the DEB is closed and sealed.

The following example (Fig. 2) shows the information recorded in a DEB tag, index and bag files when acquiring information from a forensic incident response tool. When a DEB is created a tag file is generated that is similar in content to that used as a physical evidence tag.

The DEB tag file shown is an example of that created once the evidence acquisition process is completed and the DEB is closed. The tag file is a plain text file comprising four main sections:

[DEB Header];
[Evidence Units];
[DEB Footer]; and
[TCB].

The DEB header contains information such as investigating officer, timestamp of when the DEB was created, and description of what, where and when evidence was captured. Within the DEB header the 'Index Format' line specifies the default content sequence of the DEBs' index files, and it may be specified per EU or apply to the DEB as a whole. The index file format is defined by a sequence of meta tags. This allows each EU to be customisable within the DEB thus enabling a DEB to store information from a wide range of devices.