

# Silicon Pathology ?

A.M.Marshall BSc CEng MBCS FRSA,  
Centre for Internet Computing,  
University of Hull  
&

Eur. Ing. B.C.Tompsett BSc MSc CEng MBCS,  
Dept. of Computer Science,  
University of Hull

June 26, 2003

## **Abstract**

Conventional forensic computing practice concentrates on the retrieval of evidence from suspects' equipment, often without much consideration of how the evidence may have arisen.

The authors show that, given the complexity of modern computing environments, it is possible for "evidence" to be deposited on equipment through a variety of mechanisms. Comparisons are drawn with human diseases and disease propagation mechanisms to show how a computing environment can become host to a variety of unwanted software and data without the owner's or user's intervention.

The model is extended to suggest hypotheses that investigators should consider when "diagnosing" the origins of evidence found on suspect equipment, and to provide prophylactic and curative techniques.

# 1 Introduction & Definitions

## 1.1 Purpose of this exercise

In this paper we propose to examine current techniques in the investigation of computer-related illicit activities and suggest how lessons learnt in “traditional” or “conventional” forensic science disciplines can be transferred to the field of forensic computing. In doing this, we aim to provide guidance for future developments in the discipline itself as well as identifying technical aspects of computing which will require investigation.

## 1.2 Definitions

In considering the approaches to developing new techniques in forensic computing it is appropriate to consider the nature of modern computer systems. As discussed in previous work [1], most modern computer systems do not work in isolation, nor are they necessarily closed systems. Rather, they are a collection of cooperating components interacting with other systems through a range of interfaces and protocols. As a result, we propose to treat the systems as being more akin to biological entities than purely electrical or mechanical systems.

This treatment leads us away from the traditional view of a computer system as primarily a source of evidence, and more towards the proposition that it should be considered as both a victim of crime as well as a scene of crime. Considering the system in this way opens up new areas of investigation which are, perhaps, not as often considered as they deserve when computer-based evidence is being recovered. Hence we present a “pathological” approach to computer-based evidence, where we accept the dictionary definition of pathology as :

“study of variance from normal. (often related to damage from injury or infection)”.

This definition, in particular, is useful to us in that it acts as a reminder that computer systems interact with other systems (human, computer, telephone etc.) and these interactions inherently present a number of vectors which can act as introducers of activity to the system.

Macfarlane, Reid and Callander [2] give a further definition :

“Pathology is the study of disease. It describes the manifestations of the disease, its progress and sequelae and attempts to determine the cause (aetiology) and underlying mechanisms (pathogenesis). It forms a bridge between basic science and clinical practice.

Disease occurs when there are variations of structure outside the normal range.”

By extending and applying these definitions to computer systems we create a situation where the forensic computer scientist must accept that the mere presence of a particular type of evidence is insufficient. Instead, they must begin to enquire how such material could have “infected” (been deposited on) the computer system.

Evidence of computer misuse can be introduced to the system through a range of vectors discussed below.

We propose that consideration of the means of introduction of the evidence should be held to be crucial to a successful investigation, as without this information, inappropriate evidence may be considered or lead to incorrect conclusions.

## 2 Traditional Forensic Computing

The “traditional” approach to forensic computing tends to treat the system as a crime scene with the concomitant assumption that it is a source of evidence. Sammes and Jenkinson [3], quoting Pollit[4], define Forensic Computing as

“the application of science and engineering to the legal problem of digital evidence. It is a synthesis of science and law.”

and go on to quote Pollit’s assertion [5] that

“Rarely is determining that the (paper) document physically exists, or where it came from, a problem. With digital evidence, this is often a problem. What does this binary string represent? Where did it come from ? While these questions, to the computer literate may seem obvious at first glance, they are neither obvious nor understandable to the layman. These problems then require a substantial foundation being laid prior to their admission into evidence at trial.”

Given these broad-ranging statements, the field of forensic computing seems to naturally fall into two types of activity - data recovery and evidence interpretation.

Recovery is, generally speaking, a straightforward task involving the treatment of the computer system as a crime scene and making use of appropriate search and extraction tools to examine the storage media present for any and all traces of relevant data. However, as volumes of data increase and software complexity rises, it is becoming increasingly difficult for “non-specialists” to carry out a thorough interpretation of the evidence recovered. This is particularly so, where attention is given only to the nature of the evidence recovered without consideration of when and how it originated. Temporal information can often be determined by appropriate examination of meta-data on the storage media, in which the system will have recorded dates and times of last modification etc., but the mechanisms of deposition are less easy to find. However, as we shall discuss later, the determination of the vector of introduction is increasingly important.

To assist with the discovery of deposition mechanisms, the approach proposed calls for the treatment of the system to be extended so that it can be considered as not only a scene, but also a victim with inherent properties and behaviours which may assist in the determination of the sources and transmission mechanisms of evidence recovered.

## 3 Organic Computing?

In a discussion of biological pathology techniques, it is usually assumed that the audience is familiar with principles of the appropriate biological systems being considered. In this discussion, however, it is thought appropriate to include some discussion of the “silicon equivalent” to a biological system. The authors apologise to the knowledgeable reader, but hope that the information contained in this section will serve as a useful contextual reference for the discussions which follow.

### 3.1 Initial Computer Software Origins

#### 3.1.1 A “new born” system

When a computer is manufactured it contains a minimum amount of software, usually contained in the boot PROMs<sup>1</sup> for the motherboard, graphics card, network card and other similar devices. This code is usually, but not exclusively unchangeable. This basic software (colloquially known as the BIOS<sup>2</sup>) is used to introduce other software into the computer system by the process we know as “booting” or “bootstrapping”. This then introduces to the empty memory software from devices such as the CDROM, floppy disk and hard disk.

A computer, in manufacture, is provided with an empty persistent storage device (such as a disk), which needs to be properly formatted, and can then be given its initial contents for the operation of the computer “Operating System” (OS). This would, typically, be obtained from the “Installation” package provided by a software manufacturer.

At this point it might seem that the corpus of data associated with such a computer can be readily identified and its origins and purpose delineated. This, of course, will only be true if the provenance of the software can be established. That is, for instance, to be able to demonstrate that the software has been acquired from a known, trusted source and is itself a known and trusted copy.

Although this question is not often asked, in a serious investigation of a computer system and its contents, one should even verify the origins and contents of the initial software package to eliminate the possibility that the system contained suspect material from that initial point.

After the initial operating system installation, packages and applications are usually installed, and again their provenance may or not be bona-fide.

#### 3.1.2 Upgrades - Growth and development

The initial software of almost any computer requires further updates and amendments. These are obtained by the owner/operator from many and varied sources, such as network downloads, magazine cover disks, “borrowed from friends”, as well as apparently bona fide sources such as manufacturer originated CDROMs. The nature of these “upgrades” is such that we cannot always be convinced that they “improve” the operation of the resulting software combination.

---

<sup>1</sup>Programmable Read-Only Memory

<sup>2</sup>Basic Input-Output System

### 3.1.3 Later Software Installation

The areas of most interest and challenge to a forensic investigator, are often the later amendments to the body of software on a computer system.

We assume that the initial software installation and planned upgrades to that software are performed knowingly by the owner/operator, although the question of provenance may be raised. After this point we must consider the matrix of possibilities that might arise:

- Installation by the Authorised User
  - Knowingly
  - Unknowingly
- Installation by Unauthorised Users
  - Knowingly
  - Unknowingly

### 3.1.4 Knowing Installation by Authorised User

To this point we have used the terms “User”, “Owner”, “Operator” somewhat interchangeably, but for the purposes of a forensic study it may be necessary to be more precise.

In a business context, for example, the “Owner” of the system might be the proprietor of the company. IT specialists may be employed to maintain and install computer systems. These specialists can often be classified as the “Operators”<sup>3</sup>. The systems may be utilised by a number of people who we can call the “Users”.

A side-effect of this is that, although the “Owner” of the system may appear to have certain rights and privileges above all other persons connected to the system, they may in fact be no more than a “User” except when it comes to legal responsibilities. Indeed, the situation has parallels with the legal situation pertaining to motor vehicles where the owner and the driver are recognised as distinct entities with different responsibilities.

The origin of software on the system may be the result of actions from any one of these categories of people.

When material of a questionable nature is discovered during an investigation, the default assumption often made is that it is behaviour by someone in the “User” category, who has acted knowingly, which has caused the material to be present.

### 3.1.5 Unknowing Installation by Authorised Users

The level of sophistication of modern software packages exceeds the level of understanding of an average computer user. It is often the case that, during the installation of one software product, a user unknowingly installs further software on their machine. Some software packages require this additional software to be installed, but if questioned the user would deny their involvement in its installation. An example might be the installation of driver systems like DirectX[6]

---

<sup>3</sup>to borrow a term from mainframe computing

included in a game installation, or Acrobat Reader[7] installed by a laser printer driver program to allow the printer manuals to read on-screen.

These unknowing installations are not always innocent, as some software may have been tampered with, in distribution or manufacture, and the user may be installing unwelcome or malicious software agents, such as trojan horses, or a virus. This was, in practice, the only propagation vector of unwanted or malicious software before the advent of the ubiquitous Internet as a distribution mechanism.

Correct use of the privilege levels provided by some operating systems can mitigate against this problem, by removing the ability of the more vulnerable user to install code unknowingly or without permission.

### **3.1.6 Knowing Installation by Unauthorised Users**

Many computer systems may be accessed by persons not normally authorised for their use. The actions of these persons, if their existence is neither suspected nor hypothesised, would be attributed to the proper users of the system.

Unauthorised users can introduce software changes to a system, often at a time greatly removed from the investigation of the system, but their existence and involvement can cloud and obfuscate the later collection of evidence. Examples of unauthorised users can be visitors to the premises whose use of the computer is unrecorded, colleagues who “borrow” a system temporarily, or other people such as janitorial staff accessing the system, as an informal “perk of the job”, outwith normal working hours.

These users can often knowingly install software, to facilitate their use of the system, but as their use is unknown and undocumented, their software changes may be regarded as unauthorised, or even more suspicious than they actually are.

### **3.1.7 Unknowing software installation by unauthorised users**

Having established that any system may be accessed by cohorts of unauthorised persons, it is not difficult to envisage a situation where, during their unauthorised use, further software is installed, without the unauthorised user’s knowledge, to change the overall configuration of that machine, in the same manner as could occur with an authorised user.

## **3.2 Data Files**

To this point the discussion has only examined the nature and origin of the software artefacts that populate the computer system under investigation. All these software artefacts will use and interact with data that has also been stored within the computer.

The same levels of analysis need to be applied to this data, and how its existence can be explained as we applied to the software.

We often consider, when discussing data files, that we may be referring to data files created explicitly by a user using some application program and saved to their storage by some explicit and knowing action.

We separate the categories of data and software, because data is only stored within a computer system under the control of a piece of software. It can never

be the case that data arrives on some form of storage without the involvement of software which itself came to the machine by the multifarious methods under discussion. Further, many assume, that the presence of certain data artefacts within the computer, must have originated from some user action. It is evident, from further detailed analysis, that software can create and store data without user knowledge, permission or intervention.

### **3.2.1 Implicit data**

The first category of data created within a computer, would be made by the system software itself, and is part of the normal operation of any system so configured. Each one of the application programs will also store and manipulate sets of files in persistent storage. These will also be implicit in the operation of that application and be comparable for any similar system running the same application.

We have already established that a system may contain software whose origins may be unauthorised or unknown. This software may also create, implicitly, its own sets of data that can be discovered or discerned later.

### **3.2.2 Network Downloads**

To this point we have not mentioned the effect of connection of a computer system to a network, such as the internet. One of the major sources of data files, in a modern computer system, is via internet delivery in some form or other. There are many applications that people use to get data from the internet, some of them operated explicitly by a user, such as a web browser, and some operating autonomously, such as an auto-update facility[8, 9].

As users interact with the internet they cause data to be downloaded from the network to be stored locally on the machine. It is the normal operation of web browsers and other tools, that a complete copy of internet documents and all their components is downloaded and stored locally for display and re-use.

### **3.2.3 Authorising Downloads**

The normal operation of web applications presents a problem when considering the question of whether a user authorised the downloading of a document from the internet to their computer.

Hyperlinks within documents usually contain a simple description of the target page, inviting the user to select the linked text and cause a document to be downloaded. Often the descriptions are limited to vague terms such as "Next Page".

The question of informed consent thus arises. Is a user in this situation being given proper notification of the nature of the material they are likely to view ?

Having followed a vague link, the user may decide, that they do not wish to be associated with some pages, and cancel the browsing activity before the page has become completely visible on their screen. However, in most cases it will be too late, as material will have been downloaded and stored on their computer, and will remain there for some considerable time thereafter.

It can be questioned, therefore, to what degree the user has consented to this downloading and storage of material they neither wanted or knew about.

This simple case is but the tip of a much larger iceberg. Many internet documents can be complex and include executable code in their make up which can download, in the background, further documents and data, perhaps to streamline or enhance some later presentation. The user, in this case, may not even select the options that trigger the display of the offending material, and be even less aware of the presence of these files.

Again the issue of consent or authorisation arises. We would argue that, in this case, the user has given no consent, as the action was dictated by the author of the document and not by informed actions on the part of the user.

### 3.2.4 Unauthorised Download

Whatever position one takes in the previous argument, there are internet mechanisms that deliver data to a user's computer that most would consider unauthorised. In particular the delivery of malicious code, often known as a virus, worm or trojan horse, can be placed in this category.

Interaction with the internet is now fraught with danger, in that the reading of any web page or email message can result in unwelcome code being executed (e.g. through HTML<sup>4</sup>-formatted e-mail being automatically displayed and processed in the "preview pane" of Outlook and similar e-mail clients [10] on our computer and unwelcome documents being downloaded without our knowledge or permission.

## 4 Outside influences

### 4.1 Pervasive Sharing Cultures

One aspect of the technology involved in the internet and some users attitude to the network as a resource that may be relevant to this discussion is the pervasive nature of the sharing of resources. This sharing may be performed knowingly, and with consent (e.g. through peer to peer file sharing networks such as Kazaa[11] and Freenet [12]), but it is oftenthe case that it happens implicitly and without consent (e.g. via the creation of administrative file shares in Windows NT etc.).

Some computer systems, are by their design, accidentally, or deliberately, open for use by others. This can be considered a feature or a flaw depending on your perspective. Others on the network then exploit this sharing ability of other computer systems. Organised Crime, in particular, seems to have become adept in exploiting this facility.

There are several well-known examples of this situation. For example the music sharing service, Napster[13], relied upon each user donating their computer and its music files into a common pool. Other similar systems have followed. Some have even gone much further, such as Freenet [12] (Fig: 1) which permits a computer user to donate some of his storage to hold, in encrypted form, unknown material generated by third parties, explicitly to avoid the sanctions of law enforcement.

Many email or web systems are, in their default state, available for sharing, and other parties have written software tools that allow the routing of informa-

---

<sup>4</sup>HyperText Markup Language - see <http://www.w3c.org/>

“Users contribute to the network by giving bandwidth and a portion of their hard drive (called the ”data store”) for storing files. Unlike other peer-to-peer file sharing networks, Freenet does not let the user control what is stored in the data store. Instead, files are kept or deleted depending on how popular they are, with the least popular being discarded to make way for newer or more popular content. Files in the data store are encrypted to reduce the likelihood of prosecution by persons wishing to censor Freenet content.”

Figure 1: Freenet Philosophy

tion, without explicit permission, through the computers of other third parties. It is known that illegal material is distributed on the internet in this manner. The presence of these “open relays” is often caused by acceptance of default installation options during software configuration and hence the presence of unnecessary legitimate software on a system.

The dependence of certain criminal behaviour on parasitism of other persons’ computers is so endemic, that they are now seen to react against the proper securing of systems, by the malicious introduction of viral vectors to re-open a closed system and make it sharable again [14].

## 5 Silicon pathology

If we accept the definition of pathology as an investigation of deviation from the norm in an attempt to explain disease and/or death, then we can draw parallels between “traditional” pathology and forensic investigation of alleged computer crime.

We have already seen that a computer system, when “born” conforms to a well-defined and known state, defined by the vendor or manufacturer. Shortly thereafter, however, it begins interacting with software and systems to which it has never previously been exposed. Although the majority of these interactions will be relatively harmless, simply developing the capabilities of the system, some will engender exposure to “disease” vectors as described above.

Thus, in conducting a correct examination of a compromised or suspect computer system, we must first establish what the norm for this system is, and further consider which of the many interactions in which it has participated, may have been responsible for the vector which introduced the evidence under investigation.

### 5.1 Establishing a Reference Model

Currently, the nearest things to a reference model establishment tool are the NSRL (National Software Reference Library)[15] which consists of a database of hash<sup>5</sup> values for known good copies of files from standard applications and

---

<sup>5</sup>hash value=numerical “signature”

operating systems and tools such as the Tripwire IDS (Intrusion Detection System) [16], which creates and checks hash signatures for files specified by the system manager.

The fundamental flaws in this approach to establishing a “norm” for a system, however, are that it inherently lags behind releases of official files and it takes no account of the interactions between programs and data. (e.g. it is impossible to produce a NSRL reference value for a Windows registry file, other than on a clean installation for a fixed configuration, Tripwire on the other hand must be updated every time a system configuration is changed).

i.e. the NSRL is primarily intended to allow investigators to check the integrity of individual files only and eliminate them from further inspection<sup>6</sup>. Tripwire, on the other hand, is designed to alert a system manager only when a protected file is modified without permission.

Given the huge volumes of data found on modern systems, the desire and need to reduce information under investigation in this way is understandable, but may lead to problems when giving evidence. Information sources such as NSRL could be used in other ways, to aid the investigator in producing a more complete and comprehensive evaluation of the system.

It is the authors’ contention, that certain combinations of apparently valid software on a system can produce unwanted behaviours, leading to “infection” or be a signature of an infection itself. (e.g. missing or modified system files) In addition to the “unknown” files investigated after “good” files have been eliminated from suspicion using a reference list, checks on combinations of “good” files should be performed to allow consideration of issues raised by interactions between these files. In order to perform this, a secondary reference database describing file combinations and issues will need to be constructed. This will allow a check similar in nature, but more complex than, a virus scan to be performed. For example, in the “open relay” case, above, the presence of insecure server software, which forms a normal part of the operating system, could easily be overlooked by an inexperienced investigator, particularly where a database such as NSRL lists this software as normal and, by implication, trustworthy.

## 5.2 What is normal?

There are so many variations in computer systems, their standard states and mechanisms for introducing infections to those systems, that it becomes unclear what a “normal” system is. In human populations, sufficient analysis has been performed that the state of a typical population member is known along with well-defined limits of variance from the norm. Using this information, it is possible to determine information such as age, gender and population group from often small remains. This information is known from the many studies of demographics and attributes of human bodies. In populations of computer systems, we can find no evidence that this work has been done to any significantly useful level.

No studies have really been performed on typical, non-crime related computer systems to determine the probability that they will be hosting particular infections or certain system attributes. If we do not know what is normal, how can we say, and identify what is noteworthy or unusual, and further how can

---

<sup>6</sup>a technique employed by forensic data recovery tools such as Encase and Autopsy

we use that information as an indicator to support evidence in the prosecution against a crime?

It might be useful to gather such information, and from evidence provided by existing internet traffic, it appears that the probability that a system contains infected and variant content is quite high.

Any plan to perform a census of existing systems would need to be done on an ongoing and progressive basis, as the technology of the computer systems themselves is evolving so rapidly [17]<sup>7</sup>.

### 5.2.1 Computer Medicine and Silicon DNA

If an accurate profile of a “normal” system can be generated, we propose to consider it as loosely equivalent to a DNA profile for the system.

One approach to the generation of such a profile could be the creation of a list of the NSRL hash values for each operating system and application program file on the machine. This list would reflect the base state of the system as it should appear during normal operation. Only the program files are considered, as these constitute the active elements in the computer and therefore the list of hash values would constitute its profile. We contend that this constitutes a valid parallel to a genetic profile in that each program will have a lesser or greater effect on the overall development of the system depending on how often it is used, similar to the effects on environmental factors on the expression of individual genes in biological systems.

By examining these “silicon genomes” for systems we would be able, based on past experience, to identify those systems which, because of their particular combinations of software, are pre-disposed to particular forms of “disease” (i.e. express recessive traits).

Additionally, periodic use of systems such as Tripwire provide similar facilities to regular medical “check-up” regimes, where the state of a system can be compared with previous states, and the appearance of abnormalities specific to this system (as opposed to the generalised norm) can be clearly identified.

Virus scanners can be equated to standard screening techniques for diseases such as cancers and HIV. Anti-viral software is more akin to inoculation, whereby the immune system is strengthened by exposure to weakened versions of the original threat. It should be noted, of course, that anti-viral software also forms part of the “genetic make-up” of the systems we are considering, thus an update of such software (or any other program in the system) constitutes a mutation which must be re-evaluated.

Combination of these profiling and check-up approaches, coupled with examination of data files might present us with the equivalent of “anti-bodies”, i.e. evidence of past “infection”, which has subsequently been removed or inoculated against.

---

<sup>7</sup>Unlike Biological systems which remain fairly stable over the same period.

## 6 Examples

### 6.1 Web page “fitting up”

When web pages are viewed, as mentioned before, the HTML and other files composing the page are loaded into the web browser and also stored in its cache storage somewhere on the viewing computer. The presence of a particular file or files in the cache can have been used as evidence of criminal activity, particularly where both “thumbnail” and full-size images are recovered.

However, it is not a requirement for the victim to actively select to download any image for it to appear on the machine. There are two common ways in which images can be “planted” on a machine during viewing of a web page. These techniques are often used to improve interactive performance of web site, and are widely recognised in the web design community as beneficial [18]. As with all legitimate techniques, they can, of course, be abused.

Although we are specifically considering images here, the same mechanism can be applied to other files (including MP3, HTML, EXE etc.)

#### 6.1.1 Programmatic preloading

Many web page editing tools, by default, include program code (typically in ECMAScript/JavaScript) to preload images (See Fig. 2) which may be used elsewhere on the website, or during specific operations on the page. Such code can easily be adapted to load images which are not actually used at any time, but are stored on the user’s machine through normal operation of the browser’s caching mechanism.

```
<script language='javascript'>
<!--
if (document.images) {
    Image1=new Image();
    Image1.src='http://www.otherserver.com/image1.jpg';
}
-->
```

Figure 2: Javascript function to preload an image into browser cache

#### 6.1.2 Non-programmatic preloading

It is not, however, necessary to use programmatic solutions to preload images for later use during web browsing. Images can, in fact, be preloaded on the front page of a site by judicious use of the `height` and `width` attributes of an `img` tag within standard HTML. By setting both of these to small values, the full file will be downloaded by the browser, as normal, and stored in the browser cache. However, it will be rendered (if possible) as a very small image on the page. If the values are suitably small (e.g. 1 x 1) the image will be invisible to the user. (See Fig. 3)

```
<img src='myfile.jpg' width='1' height='1' >
```

Figure 3: Use of width and height attributes in img to preload images for later use

## 6.2 R. vs. Schofield, 2003

In this case, Schofield was charged with possession of paedophile pornographic images on his computer. The Crown Prosecution Service elected to take the case to court on the basis of evidence obtained by a conventional forensic computing unit.

Subsequent examination of the computer, by a defence expert, revealed the presence of a Trojan Horse program on the machine. It was considered that this program could have given access to the machine to an unknown party, across an internet connection, especially as the software appeared to have been installed before the images were deposited on the machine.

As a result, the presence of the images on the machine could no longer be used as compelling evidence that the accused had chosen to download these images, and he was acquitted [19].

Had a more thorough investigation of the machine been conducted, prior to the decision to prosecute being made, it would have been possible to determine:

- if the Trojan Horse program could really have been involved in deposition of the images
- if there was substantial corroborating evidence, from the computer and other sources (e.g. credit card transactions)

Hence, the element of reasonable doubt introduced by the discovery of the virus could have been considered in advance and properly evaluated.

## 6.3 Unauthorised Proxy Implantation

As computer users and network administrators become wise to the risk of operating insecure systems, the number of exploitable systems may decline. Unfortunately the pool of exploitable systems is being expanded by the affects of certain worms which when they infect a machine, they permit the system to be used to route traffic or carry data for miscreants. The Sobig virus, in particular, when it infects a users machine, can be used as a vector to install a proxy server[14]. Others use the infected computer as a file or web server to further the distribution of improper material[20].

## 6.4 Outside insider dealing

This is a speculative example. The authors are not aware of any real occurrence of this form of activity, at the time of writing. However, readers may know differently.

Consider the situation in a financial institution based in the City of London. As a reputable and responsible organisation, the firm has taken all reasonable

steps to secure its own network and computer from unauthorised use. It has correctly configured firewalls, anti-virus precautions, configuration management and user-education programmes in place.

Consider also, the rate of evolution of computer technology. The trends currently in vogue are for increased mobility and flexibility of staff, enabled by the use of mobile equipment (e.g. laptops, handheld computers, SmartPhones) coupled with widespread creation of pervasive networking using “wire-free” technologies. [21]

It is not difficult to extrapolate a scenario whereby the current vogue for “wardriving” combined with the “sharing culture” described above leads to malicious activity.

One example of this could be the attempted mass implantation of insider trading evidence on multiple, apparently secure, networks by exploitation of the inherent weaknesses in mobile equipment and wireless communications. By inserting a virus which combines the capabilities of a wireless network “sniffer” [22] with the payload capabilities of a worm into a single wireless-enabled laptop, the malicious software can be unknowingly distributed by a single person working in the back of a London Taxi as they move around the square mile. (c.f. the spread of the SARS virus). This scenario is made even more likely by the manner in which wireless network “hotspots” are propagating to offer internet connectivity to patrons of coffee-houses and hotels worldwide. Of necessity, these public access points are less secure than corporate networks and hence reduce the security on individual machines connecting to them for the duration of the connection (and beyond if the user does not restore normal settings immediately after using the public access point).

The evidence itself, once implanted, would appear to have arrived via authorised points on corporate networks and is unlikely to be detected as it does not, in itself, have any affect on the network on which it is implanted. However, at some future point, it may prove to be a destructive payload triggered by an anonymous phone call to the appropriate authorities.

Although this may seem to be a far-fetched example, the same scenario could have a major impact on a forensic computing laboratory which has not take adequate steps to prevent wireless, or any other, intrusion during investigations[23, 24, 25]. This will become a more serious issue as more machines with built-in wireless technology[21] come to market and are widely used by staff, visitors and passers-by alike.

## 7 Conclusion and Recommendations

As the foregoing discussions and examples have shown, there are many and worrying mechanisms which can place data files of a dubious nature onto the disk of a suspect computer system. These could present difficulties for the prosecution if the investigator has not considered many of the points raised in this paper, or conversely, provide many opportunities for the defence.

### 7.1 Application

Normal practice in the conduct of a scientific examination is to propose at least two hypotheses to explain the presence of any evidence discovered. The

likelihood of each hypothesis being true is predicted and a Bayesian approach is used to indicate which is the more likely explanation [26].

The foregoing discussions have highlighted a considerable number of possible mechanisms which could explain the presence of material on a computer system, hence it can act as a preliminary list of hypotheses to be considered during the investigative process.

We have also proposed means by which the investigator can be assisted in the selection of appropriate alternative hypotheses based on the collection and analysis of “genetic” data about computer systems. This data is currently, regrettably, unavailable.

## 7.2 Recommendations

Forensic computing today is still in its infancy, where examiners are, on the whole, expected to be generalists rather than specialists, unlike other forensic sciences. Where specialists exist, they tend to “over-specialise”, concentrating solely on one particular technology (e.g. Unix systems intrusion, mobile phones) with minimal consideration of parallel disciplines which could assist them.

We are, in effect, working in a mode similar to that described by Stuart Kind[27] before the current approaches and acknowledgement of integrated teams of specialists appeared.

However, we are in the fortunate position that we may be able to predict how the science of forensic computing should develop. Some options and recommendations follow.

### 7.2.1 Professionalism

- Complexity of computer systems is increasing rapidly, not least because of the pervasiveness of network availability. To conduct a proper examination of any system now requires a forensic computing scientist who has a thorough understanding of the principles of operation of such systems. This leads to a requirement for a properly structured program of education.
- Because of the increasing complexity of modern computer systems, toolsets to aid the assessment and selection of hypotheses should be developed. Application of such toolsets could aid crime-prevention by identifying “at risk” systems before they are compromised.
- Steps need to be taken to educate the mass population of computer users about the risks inherent in use of technology. This would serve to reduce the levels of “opportunistic” crime.
- Technology providers (including software houses, ISPs etc.) need to take greater responsibility for the production of inherently secure systems.
- Given that many current attacks on computer systems have parallels with biology, the computer scientists should take note of lessons learned in other disciplines and seek to work in tandem with those other disciplines to produce a more holistic investigative approach.

- As new technologies develop and emerge, recognition of new areas of expertise should be promoted and such new expertise should be embraced to retain the holistic approach.

### 7.2.2 Procedural

We propose that the standard examination process be expanded to take account of the complexities we have introduced. Thus a “standard” investigation of any computer system would be:

- Determination of the “normal” configuration of suspect system
- Determination of the possible vulnerabilities in “normal” state
- Determination of any deviation from “normal” state
- Determination of the relative likelihood of exploitation of vulnerabilities (risk assessment)
- Attempt to determine which vulnerabilities were really exploited
- Use of Bayesian approach to compute likelihood of suspect involvement

### 7.2.3 Research & Development

To support the professional & procedural recommendations, considerable background work is required in the following areas:

- Computer Population Sampling & Profiling. To produce the “silicon genome” required for proper vulnerability and misuse assessment.
- Improved Toolsets. To automate processes where possible.
- Enhanced Education Programmes. For investigators and users alike, to ensure that new technologies are not overlooked and that risks are properly understood.
- Vulnerability reference databases. Allied to the “genomic” data, to allow rapid evaluation of suspect systems. Work on this is likely to require significant effort in optimisation due to the opportunities for combinatorial explosion offered by the rate of evolution in technology.
- Since technologies operate beyond local boundaries (especially in the case of networks), there is a need for the establishment of at least a national centre of knowledge and expertise which can work within the jurisdiction of the technology, rather than the geographical boundaries made obsolete by that same technology.

Initiatives such as NCOF<sup>8</sup> and the recent EPSRC<sup>9</sup> Crime Prevention & Detection programme have started to create a more holistic relationship between forensic scientists and those working at the leading edge of modern science.

We have concerns, however, that, in the area, of information and communication technology the rate of change is so fast that there is a risk that criminals are exploiting the opportunities faster than current professionals can adapt.

---

<sup>8</sup>National Crime and Operations Faculty

<sup>9</sup>Engineering and Physical Sciences Research Council

## References

- [1] Marshall A.M. and Tompsett B.C. Spam 'n' Chips - a discussion of internet crime. *Science & Justice*. 2002; 42 : 117-122
- [2] MacFarlane P.S., Reid R and Callander D. *Pathology Illustrated*. London, UK. Churchill Livingstone 2000:ix
- [3] Sammes A.J. and Jenkinson B.L. *Forensic Computing - a Practitioner's Approach*. London, UK. Springer-Verlag 2001.
- [4] Pollitt M.M. *A Five-Step Approach to Forensic Examinations*. Baltimore, MD, USA: FBI: Undated
- [5] Pollitt M.M. *Principles, Practices and Procedures: An Approach to Standards in Computer Forensics*. Second International Conference on Computer Evidence. Baltimore, MD, USA: 1995.
- [6] Microsoft. DirectX. <http://www.microsoft.com/windows/directx/default.aspx> Last modified: 2nd June 2003 Last viewed: 26th June 2003
- [7] Adobe. Acrobat Reader. <http://www.adobe.com/products/acrobat/main.html> Last viewed: 26th June 2003.
- [8] Microsoft. Description of the Automatic Update Features in Windows. <http://support.microsoft.com/?kbid=294871>. Last updated: 7th May 2003. Last viewed: 26th June 2003.
- [9] RedHat. RedHat Network. <http://rhn.redhat.com/> Last viewed: 26th June 2003.
- [10] Microsoft. Patch available for Outlook Express Preview Pane Vulnerability. <http://support.microsoft.com/support/kb/articles/q261/2/55.asp> Last updated: 4th June 2003 Last viewed: 26th June 2003.
- [11] Sharman Networks Ltd. How Peer-to-Peer (P2P) and Kazaa Media Desktop Work. [http://www.kazaa.com/us/help/guide\\_aboutp2p.htm](http://www.kazaa.com/us/help/guide_aboutp2p.htm) Last viewed: 26th June 2003.
- [12] Freenet. <http://freenet.sourceforge.net/index.php?page=whatis> Last viewed: 25th June 2003.
- [13] Menn J. *All the Rave: The Rise and Fall of Shaun Fanning's Napster*. Crown Pub. 2003.
- [14] LURHQ. <http://www.lurhq.com/sobig.htm>, Sobig.a and the Spam You Received Today, Last viewed : 25th June 2003
- [15] National Institute of Science and Technology. National Software Reference Library. <http://www.nsrl.nist.gov/> Last updated: 23rd June 2003 Last viewed: 26th June 2003.
- [16] Tripwire. Tripwire Open Source, Linux Edition FAQ. <http://www.tripwire.org/qanda/index.php> Last viewed: 26th June 2003.

- [17] Moore G. Cramming more components onto integrated circuits. *Electronics*: 38(8) 1965.
- [18] Niederst J. *Web Design in a Nutshell*, 2e. Sebastopol, CA, USA. O'Reilly 2001: 176.
- [19] Reading Evening Post. Program put child porn pics on my PC. <http://www.getreading.co.uk/story.asp?intid=6541> Last modified: 24th April 2003 Last viewed: 25th June 2003.
- [20] BBC. Spam virus 'hijacks' computers. <http://news.bbc.co.uk/1/hi/technology/2987558.stm> Last modified: 13th June 2003 Last viewed: 25th June 2003
- [21] Intel. Intel Pro / Wireless 2100 Network Connection. <http://www.intel.com/products/mobiletechnology/prowireless.htm>. Last viewed 25th June 2003.
- [22] The Shmoo Group. Airsnort. <http://airsnort.shmoo.com/> Last modified: 22nd Feb 2003. Last viewed: 25th June 2003
- [23] Bamford, J. *Body of Secrets: Anatomy of the Ultra-Secret National Security Agency*. Random House: 2001
- [24] Ward G. *Eavesdropping On the Electromagnetic Emanations of Digital Equipment: The Laws of Canada, England and the United States*. <http://www.fas.org/irp/eprint/tempest.htm> 1993. Last viewed: 25th June 2003.
- [25] Faraday M. In: James F ed. *The Correspondence of Michael Faraday (Vol 2.)*. London, UK: IEE 1993.
- [26] Booth G. Johnston F. and Jackson G. Case assement and interpretation -applications to a drugs supply case. *Science & Justice*. 2002; 42: 123-125
- [27] Kind S. *The Sceptical Witness*. Hodology Ltd. 1999