V-isoNet: A Virtualised and Isolated Network using Open Source Technologies

Michael Gleeson[1], David Markey[2], Fred Mtenzi[3]

[1]School of Computing
Dublin Institute of Technology
Kevin Street, Dublin
Republic of Ireland
michael.gleeson@comp.dit.ie

[2]School of Computing
Dublin Institute of Technology
Kevin Street, Dublin
Republic of Ireland
david.markey@comp.dit.ie

[3]School of Computing
Dublin Institute of Technology
Kevin Street, Dublin
Republic of Ireland
fred.mtenzi@comp.dit.ie

Journal of Digital Information Management

***ABSTRACT**: Computer Security and Forensics is an emerging academic discipline in which there are many challenges facing educational institutions in teaching, from both an educational and technical perspective. The School of Computing in Dublin Institute of Technology offers two Security and Forensic Computing modules, a 4th year BSc (Hons) module and an MSc module. Currently the BSc module requires a practical section to the course while the MSc module does not require practical work. In order to aid teaching during the next semester, both sets of students will be exposed to a Security and Forensics practical laboratory. This series of research proposes a phased approach to design and establish a suitable overall environment to provide a comprehensive Computer Security and Forensics practical laboratory. The three identifiable components are (1) an isolated networked environment, (2) an exposed repository to be used to demonstrate vulnerabilities and (3) the formation of a Security and Forensics toolkit to be used in conjunction with the isolated network environment and the exposed repository. This first paper, in a series of three, will introduce the reader to the overall research objective. It will then focus on research and development of the first identified component, namely an isolated network environment.*

Categories and Subject Descriptors
D.4.6 Security and Protection]; C.2.1 Network Architecture and Design]; Distributed Networks: **K.3 [Computers and Education]**

General Terms: Computer security and forensics, Computer education

Keywords: Security, Forensics, Isolated, Network, Virtualization, Open source, Education

Received: 2 June 2009; Revised 13 August 2009; Accepted 1 November 2009

1. Introduction

Educators are currently required to cultivate a generation of graduates which are security savvy and to equip these students with the required skills and knowledge to utilise available tools and techniques that assist in the forensic analysis process [1].

Another challenge in introducing students to security and forensic analysis lies in the students varying backgrounds. This is especially more prevalent with MSc students, graduates of Computer Science have a better understanding of the subject matter than those from a business or management background. While lab work is currently not an essential component to the School of Computing MSc module, it is of benefit for the students to complete a practical element to complement traditional teaching lectures. It is also beneficial for the lecturer to gauge the general level of student proficiency in computing and security.

From a technical perspective, a third level institution must prioritise protecting its existing computing environment and ensuring that it is not compromised in any way by the teaching of Computer Security and Forensics. Other technical issues such as hardware, cost, design and the appraisal of current technologies available to facilitate the teaching of Security and Forensics must be addressed.

Section 2 of this paper introduces the reader to the phased approach which will encompass the entire research area. It will also identify the major focus of this paper and describe each subsequent phase. Section 3 will examine an isolated network environment devised as part of this initial research, discuss the thinking behind this environment and why it is required. Section 4 investigates the hardware and analyses the technologies available to implement this environment. Section 5 describes the implementation in detail. Section 6 provides some analysis of the system in practical terms. Section 7 provides critical analysis of issues discovered and explores possible solutions. Finally, Section 8 presents conclusions and a discussion linking the research to subsequent phases of research.

Throughout the entire scope of this research and the teaching module it is necessary to point out the ethical and social responsibility required from Students, Lecturers and Technical staff [2].

2. Phased Research

This research project was conceived out of a practical requirement, therefore requiring a practical solution. The benefit of this practicality is that tangible or actual implementations can

be expected as a result of the research. With this in mind the research clearly identified three distinct deliverables in functional terms, namely, an isolated network, a client repository and a Forensic Analysis Toolkit.

An isolated network is critical to the successful analysis of vulnerabilities and attack tools [3]. This isolated network will allow students and lecturers to interact without fear of compromising a production network. Maintaining an isolated network introduces issues of policy guidelines and implementation while requirements such as the need to transport data from the Internet to the network must also be addressed. This is the focus of this research paper and has been labelled Phase One of the research.

The second facet identified was the client repository which is to be introduced to the isolated network to introduce threats and expose vulnerabilities. This client repository is required to fully utilise the benefit of the isolated network [4]. This will be examined in detail in Phase Two of the research and a separate research paper will be presented on the implementation of this client repository.

The final phase of the research, Phase Three will involve the development of a Forensic Analysis Toolkit which is to be utilised in tandem with the isolated network environment and the client repository, to analyse and record forensic evidence for use in tracking security breaches and attacks.

3. Isolated networked environment

An isolated network environment is required due to a number of factors, from both a teaching and a technical aspect. The goal of teaching security and forensics module in the School of Computing is to prepare students for 'real world' practicalities of investigating and analysing security threats. Essentially for students, a practical demonstration of attacks and vulnerabilities is of benefit to help to reinforce and extend concepts conveyed in lectures.

The goal of the technical staff of the School of Computing differs in that the main responsibility lies in protecting the existing network. This must be balanced with the requirement to offer students and lecturers a 'playground' in which students and lecturers can experiment without fear of corrupting or attacking the production network of the School.

To satisfy these teaching and technical requirements a computing environment was devised to a) allow student the freedom to interact within a networked environment and b) to isolate this environment from the rest of the institute's actual network, therefore avoiding any issues regarding the compromising of the institute's back-bone network [5]. The approach decided upon to implement this isolated network was to leverage the utilisation of virtualization technologies to the utmost.

Virtualization technology suits the purposes from two aspects. Firstly, virtualization allows the ad-hoc addition of virtual machines to the isolated network. Secondly and purely from a public relations aspect, having the 'playground' network physically limited to one machine adds to the level of trust. In other words Academic and Technical staff and students will have a greater perceived confidence in the isolated aspect of this security 'playground' as it is only located on one piece of hardware. Figure 1 conceptually represents this 'playground' or isolated networked environment.

4. Hardware and technology

The hardware available to host the isolated network environment, supplied by the School of Computing, DIT is the following. A Sun Microsystems Fire X4150 with the specification as shown in Figure 2. The Sun Fire X4150 offers a high compute density, increased storage capacity and networking connectivity within a single 2U rack unit.

Virtualization technologies are increasingly becoming well-matched to solve the issues relating to the teaching of Security and Forensics. The idea of an isolated network has been achieved before, the differentiation in this study lies in its implementation of an isolated network internally on a hypervisor or virtualized system. The theory behind this can be traced back as far as 1973 in a paper by Madnick and Donovan [6], however the virtualisation technologies available at the time are not comparable to modern hardware or software. This paper proposes to label the implementation of this virtualized isolated network "V-isoNet", standing for Virtual isolated Network. Open source and commercial vendors have a number of options available to implement this, namely VMWare, Xen and User Mode Linux (UML). For the purposes of this research there were two options analyzed and evaluated for the technological implementation of the V-isoNet were to be decided between two distinct virtual machine methodologies.

First option is a Type 1 native hypervisor, namely Xen running on a Linux distribution. A native hypervisor is a software program that runs directly on a physical hardware platform and the guest or virtual machine is run at a second level above the hardware. More specifically Xen is open source software and functions as a virtual machine monitor or VMM, it allows several guest operating systems to be executed concurrently on a single physical system with close-to-native performance [7]. In our V-isoNet implementation, each individual student would have access to a single Virtual Machine (VM) and we could create an internal network of any IP address range. The difficulty with this solution is the isolated aspect, how do we enforce that IP addresses can't change and that students can't take it on

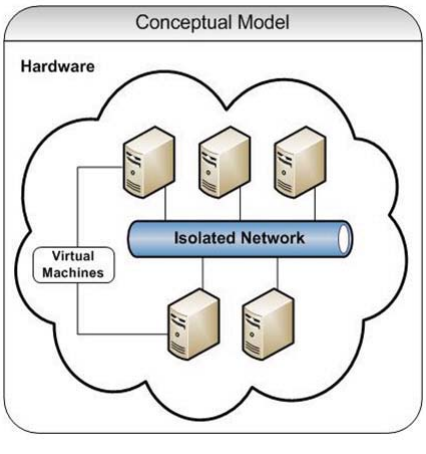


Figure 1. Conceptual Model

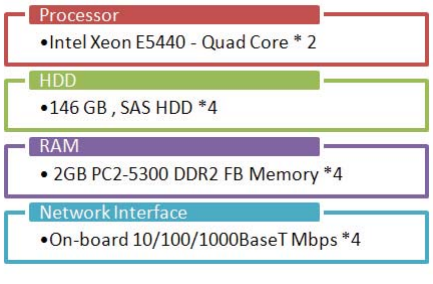


Figure 2. Sun Fire X4150 Hardware Specification

themselves to statically set their own IP addresses.

The second option is a Type 2 hosted hypervisor, namely Sun's VirtualBox running on a Linux distribution. A hosted hypervisor is a software program that runs within an operating system environment and the guest or virtual machine is run at a third level above the hardware [8]. This configuration would allow us to create an entire individual and separate network on a Remote Desktop Protocol (RDP) session for each student. The scalability of this is in question however, as within this idea a student would remotely access a VM acting as a Terminal server by RDP instead of remotely accessing the individual VM. This would allow students to run not just their own VM but a number of VM's to create their own individual network. This would allow the students to see interactions on both sides (attack PC and compromised PC). Network and bandwidth capabilities would be an issue in this configuration.

5. Implementation

Implementation of the V-isoNet took the form of the first option. After initial testing and analyzing compatibility issues the native hypervisor operating

system chosen was Debian. Debian is a freely-available operating system that is based on UNIX, its development is over seen by Debian Project which is an independent, not for profit and decentralized organization. To install Debian OS, the most recent distribution (5.0 or 'lenny') was downloaded in .iso format and burned to DVD to be installed on the Sun X4150. A basic installation was run, any required updates applied and the server was named 'v-isonet'. Next it was necessary to install the Xen hypervisor software. This can be done through a terminal with root privileges using Aptitude package manager, the following command will install Xen and required dependant packages on the Debian system.

```
[root@v-isonet ~]# aptitude install xen-hypervisor-3.2-1-i386
```

These instructions provide a vanilla configuration of Debian OS with Xen virtualization installed. This is the core of the V-isoNet. Once this setup is complete the VMM is ready to be populated with guest machines or VMs. The default representation for each guest OS is dom*U*, where *U* reflects a numerical instance of a VM. In this schema therefore the initial host or privileged system is referred to as dom0, dom0 is the domain you log into to control the hypervisor and this is shown in Figure 3.

There are two modes of virtualization available to guest OS's in Xen, namely para-virtualised and HVM (Hardware Virtual Machine). Para-virtualised domains perform better but require a modified guest OS while HVM domains are fully virtualized.

The School of Computing use Ubuntu, Debian and CentOS as para-virtualised guests and use HVM to virtualize Windows guests. This setup provides the School of Computing with virtual machines that students can take ownership of and utilize to run practical security and forensic laboratories as part of their studies. Xen uses configuration file to create domains, the main command for controlling the hypervisor is <xm> and its usages is displayed in the following terminal excerpt. The command to create a guest OS <xm create> is also displayed.

```
[root@v-isonet ~]# xm
Usage: xm <subcommand> [args]

Control, list, and manipulate Xen guest instances

[root@v-isonet ~]# xm create
Usage: xm create <ConfigFile> [options] [vars]
Create a domain based on <ConfigFile>.
```

The configuration file itself contains parameters for the guest OS, such as kernel details, memory allocation, disk location, boot loader, CPU allocation and virtual ethernet interface (vif). It also contains attributes such as name, access methods, available devices (USB, CDROM) etc. Below is an excerpt from a config file for a Windows XP VM. This can be repeated to create as many VMs as required; the VMs can be of varying types of operating systems.

```
kernel = "/usr/lib/xen/boot/hvmloader"
builder = "hvm"
vcpus = 2
memory = 3000
name = "windowsxp"
vif = [ 'bridge=fypbr0' ]
disk = [ 'phy:/dev/xen1/xpvolume,hda,w' ,
         'tap:aio:/xen/ISOs/WinXPSP3.iso,hdc:cdrom,r' ]
```

What has been achieved so far is simply a vanilla configuration of a native hypervisor and populated it with basic virtual machines for use by students. There are a number of further steps required to be implemented in order to achieve goals set out at the beginning of this program of research. These are to (i) implement a mechanism for routing traffic, thus allowing external network access from the VMs (ii) to enable a mechanism whereby access inward to the VMs is possible from various remote clients and (iii) to filter traffic, ensuring that no VMs can access the production network.

For these purposes a VM is implemented to act as a 'service provider' for the V-isoNet. This 'services' server labelled 'v-services' for V-isoNet Services will provide the services for part (i) routing and external traffic and part (ii) remote access to any VM.

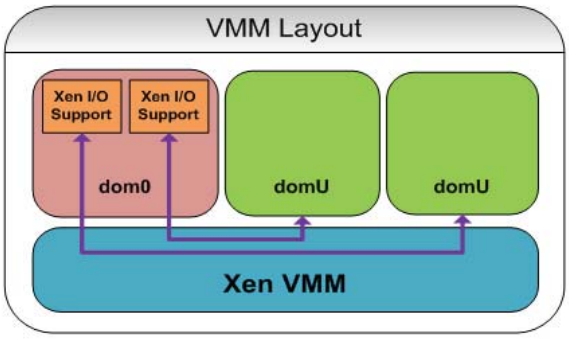


Figure 3. Domains and VMM layout

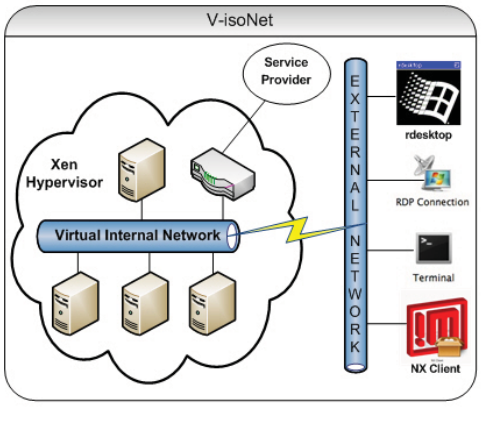


Figure 4. V-isoNet

For technical reasons traffic filtering between the production network and V-isoNet must be implemented by the VMM therefore this service will be provided by dom0, namely the 'v-isonet' server. Figure 4 graphically displays the model for V-isoNet, including all components.

What follows is a technical over view of how these three aspects were implemented, the technologies that were utilised and why these technologies were chosen. The v-services server was created as a VM and as with dom0 the latest Debian linux distribution installed.

5.1. Routing and external traffic

The first step at this stage of the setup is to enable a virtual internal LAN with is treated by each VM as a DMZ (de-militarized-zone). To accomplish this and to have the VM's hosted on this isolated network masqueraded to the rest of the external network, a form of routing is required. A router offers a single point for forwarding of information. This is achieved by installing and enabling IPTABLES on the v-services server. IPTABLES is a mechanism that uses Network Address Translation (NAT) to share one external IP address between VMs on a private internal network. As before with the Xen installation, Aptitude package manager is used. The following command will install IPTABLES on our v-services system.

```
[root@v-services ~]# aptitude install iptables
```

Traffic on v-services is enforced through IPTABLES policies accepting traffic from the private internal network gateway and routing to an external access interface. This interface is attached to dom0 which in turn, the guest VMs are also connected to. In the School of Computing VMs are assigned a 192.168.50.x internal or isolated IP address schema and then uses NAT to gain access to DITs production network, whose schema is in the form 147.252.x.x.

5.2. Remote access to Virtual Machines

The next step in setting up V-isoNet involved allowing a student to access their individual VMs. To implement this, a number of factors needed to be considered. These factors included, what type of access was required, what level of access and where would access be allowed from. Another complication was in relation to how these factors had differing requirements based on the VM, for example a Linux/UNIX VM would simply need terminal session (ssh) access to a command interpreter whereas a Windows based VM would preferably require RDP graphical access.

To enable this variance of access OpenVPN was chosen to offer an SSL based solution to both offer and restrict access to the VMs. Firstly OpenVPN is required to be installed on v-services. To install the package it is required to run the following command in a terminal window.

```
[root@v-services ~]# aptitude install openvpn
```

Once this is installed and configured, the OpenVPN client is required to be installed on each physical machine that will be used to gain remote access to the VMs. Setup and management is simplified by the utilisation of a management port over https (in this case port 1195). In the School of Computing the OpenVPN client was installed on the base PC image for all labs, thus enabling student to access VMs from any of the School of Computing labs. Once a VPN connection is made students can utilise their own preferred access client, for example ssh, RDP, NX etc, all that is required is student authentication on the VM. The added benefit of this is that this implementation method circumvents all the previous factors in relation to access levels, rights and clients that were previously stated.

5.3. Filtering traffic from V-isoNet

The final step in the configuration process is to employ additional functionality to restrict VM access internally within an isolated network. This is achieved by utilizing the functionality offered by a utility called Ethernet bridge frame tables or simply 'ebtables'. Ebtables is a firewalling tool to transparently filter network traffic passing a bridge. To install this is the same mechanism as before by running the following command in a terminal window.

```
[root@v-isonet ~]# aptitiude install ebtables
```

As ebtables is filtering traffic to and from the production network over the physical ethernet adapter it must be installed on the VMM that has access to the physical hardware and not a virtual interface.

To fully understand the functionality offered by ebtables it is necessary to be aware of how Xen handles IP addressing for each of the VM's. Xen will setup clients IP address during machine startup by means of a vif ip statement in the vm_xen.conf file. Xen supports IP declaration in a vif statement within a VM config file. The configuration excerpt below shows an example of such a declaration (you can declare multiple IPs by simply putting a space between them).

```
vif = [ip=192.168.50.1 192.168.50.2 ...]
```

Ebtables allows us to bind IP address(es) from virtual interface list to MAC addresses from a vif list. So this way an un-trusted user of a VM is limited only to the IP addresses defined in Xen conf file and trying to change existing IP address to another IP on the production network will drop any data packets and therefore render that VM unresponsive. Figure 5 represents the interaction of interfaces in V-isoNet.

This section of the research paper has described how all varying aspects of the V-isoNet were achieved. When implemented as specified, this will provide any educational institution a core and extensible infrastructure to implement Security and Forensics practical sessions.

6. Practical use of V-isoNet

The School of Computing, Dublin Institute of Technology currently has this V-isoNet implemented as a prototype for use by its students. It is populated with 20+ Virtual Machines, with varying Operating Systems installed. The infrastructure was utilised to teach a 4th year BSc in Computer Science

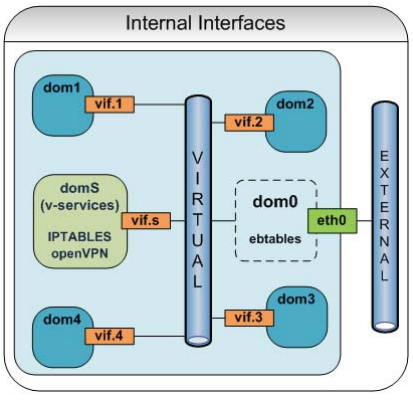


Figure 5. Internal Interfaces Interaction

Security module and the following was noted by lecturer, Dr. Fred Mtenzi.

The VMs provided a platform for students to install and experiment with relevant security software. Each student was given an account and used it for the duration of the module. Among other things, students installed security tools such as Nmap, Cain & Abel, Network Miner, WinPCap etc. Students used some of these tools to monitor network traffic, capture and analyze the network traffic for information. This practically demonstrated to students how, that if captured by an unauthorized intruder, it may lead to the machine being compromised or personally identifiable information being compromised.

Because students could also access the web while still being isolated, it was possible to utilize Cain & Abel and other traffic monitoring tools all at once. In most cases Cain & Abel attempted to get users password and if it was encrypted, tried to decrypt it.

Comments from Dr Fred Mtenzi "As lecturer, V- isoNet helps a lot as it allows me to concentrate on module preparation and delivery rather than tweaking OS and hardware. Students had a firsthand experience of how much information about the machine or the user identity can be collected by authorized and unauthorized personnel."

7. Critical Analysis of V-isoNet

As V-isoNet is a prototype system its performance was generally acceptable, however one of the problems was that the access to the VM account and even using the applications was noted to be slow at times. This bottleneck is issue with the current incarnation of the system and must be addressed whether it is networking, bandwidth or configuration. For example, students had to install Microsoft .NET in order to be able to install security tools which was network/bandwidth intensive. This section of the research investigates and explains possible reasoning for the apparent bottleneck.

To address these issues it is necessary to look at how V-isoNet is setup and what steps are taken to resolve source and destination addresses both internally and externally. This can possibly provide reasoning behind why latency issues are experienced, by analysing the sequence an internal VM takes to access any external location. Firstly an isolated VM, with a 192.168.50.x IP address requires two steps to gain access. The initial step of routing, utilising IPTABLES software residing on the 'v-services' server and passing through a virtual interface. Once this step is completed a process of filtering at the MAC level occurs. This is achieved by utilising ebtables software residing on the 'v-isonet' server and passing through a physical interface. This two step process is graphically represented in Figure 6.

The addition of this two step layer of address translation and filtering can go some way to explaining any latency issues

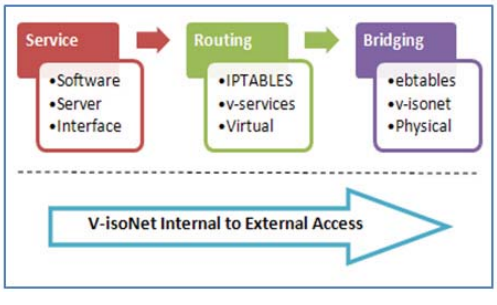


Figure 6. Internal to External Access Sequence

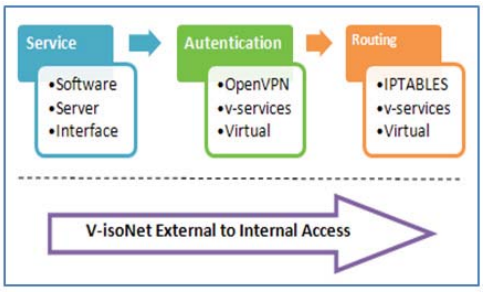


Figure 7. Internal to External Access Sequence

experienced when a VM is trying to access an external destination. However this filtering is required to enable the isolation of the environment. In this way the trade off is made between performance versus requirements.

There is a similar sequence experienced when an external client attempts to access an isolated VM. Firstly upon seeking access to an internal VM, the client must authenticate. This invokes the OpenVPN service which is provided by the 'v-services' server and accessed through a virtual interface. Upon authentication, the client must then route to the specific VM by utilizing IPTABLES which is also provided by the 'v-services' server using a virtual interface. Again here there is a two step process to gain access to the VM's which can introduce some latency issues. This sequence is graphically displayed in Figure 7.

While both Internal and External access have two step sequence, it was noted that internal to external access was perceived to be slower than external to internal access. This can be explained due to the functions that are being carried out depending on the direction of the access. On external to internal access the services are simply authentication a client and then routing to the correct VM. This is a straightforward process requiring minimal computing power and bandwidth. However routing from internal to external destinations the services invoked involve routing and also MAC level filtering. This service can involve larger than normal computing and may then take up excessive bandwidth while also all operating over one physical network interface.

A possible solution to this would be to introduce a second network card operating uni-directionally, this would then double the bandwidth available to internal and external traffic. However this would not address the issues of source and destination address resolution and of filtering.

8. Conclusion

This program of research has enabled the School of Computing to utilise a prototype system as outlined to aid the practical element of a Security and Forensics module in Dublin Institute of Technology. As this is a practical program of research, it is ongoing and evolving. While benefits to the School have been acquired lessons have also been learned. In its current incarnation V-isoNet is a prototype, it is also not comparable to any other system in DIT therefore the precise benefits are hard to quantify.

Benefits are derived in an intangible means, or are difficult to quantify, for example the protection of the Schools backbone LAN from inadvertent compromising of production systems. V-isoNet has enabled School of Computing academic staff to encourage student's innovation by giving the student the ability to in effect do or try 'anything' without consequence. Students have provided positive feedback in relation to the system, especially so for those who utilise their own personal laptops. The V- isoNet allows these students to use their own laptops by

installing OpenVPN client, while also protection their laptops from themselves during the course of any practical demonstration.

Critical analysis of the system was provided by the identification of latency issue throughout the use of the prototype system. These latency issues can be explained due to the configuration of V-isoNet and shows a performance trade off against the requirements of the system. It is also proposed to have a dedicated physical network interface for traffic in each direction, thus expanding the bandwidth capabilities of V-isoNet.

Overall this is a three phased approach to creating a complete Security and Forensics practical for a specific module. This paper addresses phase one, an isolated network. As this is a prototype a more functional and redundant system is planned for next academic term based on findings from this program of research. It is planned to implement this in tandem with the next phase of research, an exposed repository to be used to demonstrate vulnerabilities.

References

[1] Yang, T. A. (2001). Computer Security and Impact on Computer Science Education, *JCSC*, 16, 4, CCSC, USA, May, pp. 223 - 246.

[2] Logan, P. Y., Clarkson, A. (2005). Teaching students to hack: *curriculum issues in information security*, *SIGCSE Bulletin,* 37, 1, ACM, New York, Feb, pp. 157 - 161.

[3] Bishop, M., Heberlein, L. T. (1996). An Isolated Network for Research, *In:* Proceedings of the 19th National Information Systems Security Conference, Baltimore, Maryland, Oct.

[4] Mattord, H. J., Whitman, M. E. (2004). Planning, building and operating the information security and assurance laboratory, *In:* Proceedings of the 1st Annual Conference on Information Security Curriculum Development, Oct, Kennesaw, Georgia.

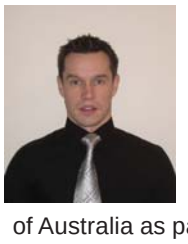
[5] Hill, J. M. D., Carver, C. A Jr., Humphries, J, W., Pooch, U. W. (2001). Using an isolated network laboratory to teach advanced networks and security, *In:* Proceedings of the Thirty-second SIGCSE technical symposium on Computer Science Education, Feb, Charlotte, North Carolina, pp. 36-40.

[6] Madnick, S. E., Donovan, J. J. (1973). Application and analysis of the virtual machine approach to information system security and isolation, *In:* Proceedings of the Workshop on Virtual Computer Systems, March, Cambridge, Massachusetts.

[7] Huang, W., Liu, J., Abali, B., Panda, D. K. (2006). A case for high performance computing with virtual machines, *In:* Proceedings of the 20th Annual International Conference on Supercomputing, June, Cairns, Australia.

[8] Vallee, G., Naughton, T., Scott, S. L. (2007). System management software for virtual environments, *In:* Proceedings of the 4th International Conference on Computing Frontiers, May, Ischia, Italy.

Author Biography



Michael Gleeson, NDip. BSc. MSc. is a Technical Officer and Lecturer in School of Computing in Dublin Institute of Technology. He has been conferred with a Bachelor of Science in Business Computing from Athlone Institute of Technology in October 2002 and a Master of Science Computer Science (1.1) from DIT in November 2007. He is currently working towards an MPhil by undertaking the role of lead researcher in Security, as part of the Ubiquitous Computing Research Group in DIT. Having previously been employed by University College Dublin, Grafton Group and Atkins Global his primary experience lies in system and network administration. He was contracted as a technical consultant for EDS and Commonwealth Bank of Australia as part of the banking groups overall IT infrastructure upgrade project in 2005. His academic pursuits involve researching as part of the UCRG in the area of academic teaching environments, security and virtualization. He has also design and delivered an undergraduate course in Operating Systems and undertakes the supervision of undergraduate final year projects and also postgraduate projects in the area of Security, Smart Spaces and Web Services.

UCRG

Ubiquitous Computing Research Group Logo