Novel Framework for Automated Security Abstraction, Modelling, Implementation and Verification

Saliou L, Dr Buchanan WJ, Graves J, Dr Munoz J
School of Computing, Napier University, Edinburgh, Scotland
l.saliou@napier.ac.uk

Abstract: This paper presents a novel framework for network security, and provides a complete solution to integrated security policies, which meets the objectives of an organisation, and also an automated verification process. The framework uses a security compiler, which converts high-level abstract definitions of the objectives of an organisation, and its security requirements. The output of this is then converted into an XML abstraction of security requirements, which can then be modelled, and converted into an implementable form, such as using firewall and IDS rules. Once it has been implemented, network agents are then used to generate and gather data allowing the security policy to be verified against the requirements.

The main areas of the framework are:

- **Formal definition and abstraction**. This involves the application of formal abstract security languages, such as an ontology mark-up language, and the novel implementation of integrated social rules, along with some form of definition of the aims and objectives of the organisation.
- **Implementation**. This involves converting the abstraction of the security policy into code and configurations, which can be implemented in the network devices, such as in the implementation of firewall and IDS rules, along with rules for data gathering agents. The paper shows practical implementations of these.
- **Test and verification**. This involves using data gathering and test generation agents to test and verify that the security system meets its initial objectives. This is obviously a key element in the system, as it provides automated feedback, and refinement.

The paper also provides novel results, which show how network agents can detect threats, and how the network can reconfigure itself, and limit its damage. It also shows typical delays for well-known worm threats and concludes with a novel method of detection and proposes methods on how the network could automate its configuration to overcome typical network threats, such as worms and viruses.

Keywords: Network agents, security abstraction, formal definition, reconfiguration, automated verification

1. Introduction

A key objective of enhancing computer network security is to promote it as an integrated process, rather than as an addition to the overall operation of the organisational network. It is thus necessary to create a framework that fully integrates the security requirements of an organisation, such as integrity, accessibility, and availability, in order to facilitate the creation of a consistent and configurable set of parameters for the assets. This will enable automatic deployment of a security policy that can be implemented on a live system, and evaluated in a real-time manner, to show the compliance of the system with the organisational objectives. This paper outlines a novel framework, which shows a logical flow of a security policy from its definition to its verfication.

Organisations have a range of techniques available to keep their assets secured, such as smart cards, firewalls, intrusion detection systems (IDS), software patches, usernames, and passwords. These may be efficient from a technical point-of-view, but mostly rely on human intervention for configuration, programming, installation, and maintenance. They also often suffer from the fact that many security implementations do not actually reflect the operational and/or hierarchical structure of the organisation (Danchev 2003, p4). Thus security is often viewed as a limitation to functionality (Smetters, et al. 2002, p83), or an obstacle to usability (Viega, et al. 2004, p62), and it becomes challenging to make choices or implement changes. A critical factor in network security, though, is time (Zou, et al. 2003, p199) and the weakness of most of the security solutions is that they require substantial efforts in upkeep (Rosamond 2004, p25). Hence, computer networks do not adapt very well to new threats or new organisational requirements (Corbitt 2002, p21). Organisations should thus have policies (Rosamond 2004, Corbitt 2002, Timms, et al. 2004, Fraser 1997) that dictate how mechanisms and procedures must be coordinated. This paper highlights essential features of a framework and outlines a prototype of the system to prove the principle with automated rules

generation, and a mitigation process to thwart a particular threat. The paper shows an example of a scenario which circumvexes a File Transport Protocol (FTP) acceptable usage policy. It uses a mitigation architecture with software agents and an intrusion detection system, and implements the mitigation using Access Control Lists (ACLs).

2. Theory

Security is a multi-dimensional concept, including: privacy, physical access restriction, application availability, network confidentiality, content integrity, and access policy (Bashir, et al. 2001, p29). All these parameters can figure in a security policy and be enforced by various mechanisms, and be audited by various software applications. This section outlines some of the key concepts involved in the model. Furthermore, key aspects that call for a novel framework for computer networks security will be underlined.

2.1 Security Policies

There are few standard definitions for security policies, but overall they have been defined as being:

- **An approach**. With this, designers observe the system in a context and from various points of view in order to understand the consequences of a choice in terms of technology (Viega, et al. 2004, Stajano, et al. 2002). The disadvantage with this method is that it is time-consuming, and requires expert knowledge.
- **A script**. This is written in a particular *language* and then interpreted or incorporated into a mechanism, such as a firewall or an IDS. Such method is utilised by Paxson for his intrusion detection system (Paxson 1999) and by Bertino, et al. (2001) in their secured web-document access system. The disadvantage is that the link between the aims and objectives of the organisation are typically not mapped correct to the technical implementation, and the designer often interprets their own understanding of the policy, with a typically related to the technical performance rather than the actual requirements for the organisation. Along with this, the script typically only relates to a specific system; for instance Paxson method only applies to the “Bro” IDS (Paxson 1999).
- **A managerial document**. This is where the objectives, aims, and rules of the organisation are expressed and justified (Corbitt 2002, Danchev 2003, Huseyin, et al. 2004, Fraser 1997). This has the problem that it often does not match of actual technical infrastructure of the organisation, and that it is often difficult to implement in a precise way. Along with this there maybe many additional features, which could be added, but these are not implemented due to a lack of understanding of technical realities at a management level.

However, whatever the method, choice or combination of security policy chosen, the success depends on the way in which the IT personnel in charge understand and interpret these policies. Danchev defines that: “*The secret lies within the individual who configure your security system(s)*” (Danchev 2003, p24). Furthermore, Kamara, et al. (2003, p214) argue that vulnerabilities can be caused by invalid assumption and that is typically a problem related to the involvement of human operators. Each of these methods, though, has their own strengths, but they also have weaknesses, thus this paper tries to integrate the three approaches, to overcome the weaknesses of each to build a complete framework for the generation, modelling, implementation and verification of a security policy.

2.2 Agents-based Systems

Security must be carried-out in a distributed manner, as networks and users, also tend to be distributed. Thus, we require agents to implement security on devices, such as firewall agents, and agents that monitor network traffic for threats, such as with IDS agents. We may also use other agents to verify the operation of the system using SNMP agents and host-based agents. A system as such would require the agents to communicate with one another and respect a hierarchy. Staniford, et al. (2002, p163) stress that agents' activities must be controlled and Santana Torrellas, et al. (2003) introduce a multi-agent system (MAS) where agents are aware of the tasks they must perform on each node, but are also aware of the organisation's global objectives. Thus,

agents would not undertake actions that could compromise the aims' completion (Santana Torrellas, et al. 2003, p368). In Campbell, et al. 1999, it is argued that the deployment or modification of services eventually requires some change within the intermediate networking devices. Hence, the challenge facing the use of MAS to automate security would be the intelligence interfacing between the agent and the hardware/software arrangement of the node to be reconfigured.

2.3 Intrusion Detection Systems

IDSs are passive systems, which inspect the network traffic for violations and report these with alerts or logs (Julisch 2003, p444). They are well suited in the refinement of security policies, espeically ones which cannot be implemented by standard devices, such as with firewalls. They can also be used to uncover malicious activities carried via legitimate means, such as a client executing a remote script on a Web browser, or messages hidden into the headers or data packets. Paxson (1999, p28) reports that many attacks present the same patterns as genuine traffic. Thus with increasing amount of networked applications and increasing network link throughput, the numbers of alarms produced by IDSs has often become unmanageable (Julisch 2003, p444; Yan, et al. 2004, p200). Furthermore, Ning, et al. (2003) report that intruders go through stages while attacking, and the traces of such activities are often scattered within alert logs. Consequently, if the IDS is used to mitigate a threat, it is important that they do not drop packets. Yet, some researchers argue that, in some circumstances, IDSs drop packets (Yan, et al. 2004), thus distributed approach to deploying the IDS's will help in reducing the verification losses. To outline the strength of using IDSs detect threats, Zou, et al. (2003) conducted an experiment where their approach could detect fast spreading worms in a suitable timely manner. Despite the intrinsic benefit of their model, Zou, et al. (2003) argue that it is not sufficient. They suggest that the output should be acted upon with a relevant counter-measure. This paper outlines an example of the detection and mitigation of threats.

2.4 Peripheral Defence Mechanism (Firewall)

Typically, organisations only have one gateway to the Internet. A common sentiment is to consider the network under the control of the organisation as trustworthy and the outside world untrusted. Thus, organisations with Internet access would rely on a perimeter defence system to keep unwanted traffic and malware at bay. This defence system will reside between the two networks and is materialised by the use of firewalls. Since this configuration is a centralised setup it offers a single point of control and allows services used to be audited (Avolio 1998, p17).

With respect to current network design and capabilities, Ioannidis, et al. (2000, p191) outline that wireless and dial-up access are examples of connection means to the corporate network that avoid the firewall's scrutiny. For the time of this connection, nodes using these means are part of the network but not totally under the control of the organisation. Hence, the probability of hosts being compromised increases and malware may exploit this type of connection to penetrate the network. Consequently, the concept of outside/inside network or trusted/non-trusted network is no longer relevant.

Glenn (2003, p23) reports that deploying access control lists (ACLs) onto the network is less complicated than removing them or perhaps modifying them. While logs will present traces of malicious events, traditional firewalls are not designed to adapt their behaviour based on these evidences. Once a weakness within the firewall is found and exploited, the presence of the firewall can be rendered useless (Kamara, et al. 2003).

Some vendors such as Symentec™ (2005) or Sygate® (2005) combine intrusion detection and firewall capabilities within their products, but these tend to be limited in the number and range of instrusions that they can detect and overcome. Also in selecting an integrated technique may impose ties on a given technology, whereas a distributed approach for firewalling such as one proposed by Ioannidis, et al. (2000, p191) has advantages in robustness, and delegation.

2.5 Network Reconfiguration

An automated approach to security deployment has many advantages over the current manual system. This is highlighed by the fact that many organisations choose the same type of equipment for their security implement not because it has the best specification, but because it causes fewer problems in supporting the system. The proposed model in this paper outlines the automated deployment of security implementations, where firewall or Snort rules can be deployed to networked device. Key factors for this include:

- **Heterogeneity**. While there are many advantages of using a certain type of hardware or software to implement security and network monitoring, it can lead to an overall dependence on a specific vendor (Campbell, et al. 1999). As much as possible device reconfiguration must have some knowledge of the actual implement on devices, and should warn the designer as to the limits of the system.
- **Remote configuration**. Most network devices allow some form of remote configuration, such as through a TELNET session, thus this is a key factor in any automated approach. This, though, could cause problems as a strong degree of authentication and authorisation must be built into the system to avoid remote configuration by intruders.
- **Integration of devices**. Many security breaches are caused by the intercommunication between networked devices, thus a key factor is to support in any security system is to allow devices to integrate together without having any security problems in their interfaces.
- **Topology**. A key factor in the reconfiguration is that the system must have a complete picture of the topology of the system, in order to allow reconfiguration of the system to thrawt and, possibly, isolate threats.
- **Mitigation**. Many systems have triggers for possible intrusions, such as triggering an alarm on a ping sweep, and rely on human operators to thawt the threat. Unfortunately, intrusions often happen at times when there is reduced support, such as in the evening and at the weekends. Hence an automated mitigation system will aid is the confinment of these threats, especially ones which have definate patterns and that can be easily implemented without disruption to other network traffic.

3. Novel Framework

This section outlines the novel framework for the complete integration of secuity from its initial specification from the user to its verification. It is focused on the concept of areas of expertise interacting with one another, thus promoting security as a process.

A security policy and its implementation should not only reflect the aims and objectives of an organisation and its hierarchical and structural aspects, but also its social and moral responsibilities. A problems in actually implemented any of these should be highlighted at the initial stage, thus a knowledge of the overall networked system and its limits of operation need to be linked into the definition. A template approach can be used to generate outlines of these policies, which provide the overall definition, and are known to be implementable on the target system. These templates are then tailored for the actual requirements, and to match the aims and objectives of the organisation. This process is defined as the policy generator, and is as near a written English format as possible, so that everyone in the organisation can understand the policy (a good description this document's content is given by Corbitt 2002). This security definition document will then serve as an input for a **Security Compiler**. The output from the compile is defined in a formal language using eXtended Markup Language (XML), which is a more formal model of the security system. This can then be used with a **Security Modeller**, which checks that there are no limitations to functionalities or hindering completion of objectives, and that it still achieves the aims and objectives of the organisation (and, obviously, that it still confirms to the organisation's moral and social responsibilities).

The mechanisms required to enforce the chosen security model would be identified from the combination of the security formal definition and the knowledge base, which includes data on the hardware equipment and software applications available within the organisation along with their corresponding threats. The mechanisms are not limited to firewall rules but could also include

details of the user rights on workstations. Lastly, the technology identifier will provide configuration details and propose test scripts to verify compliance. The configuration data could be identified as similar to the approach, used by Paxson (1999) for his IDS system.

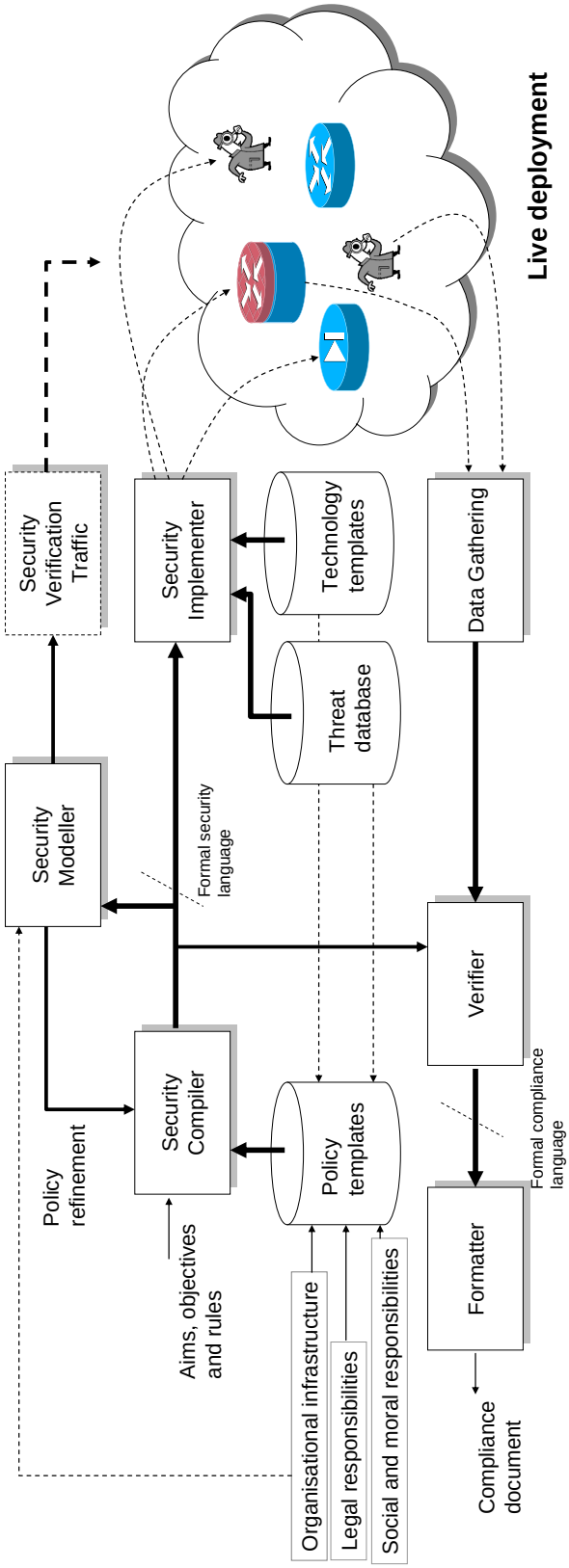

Figure 1: Framework's Abstraction

The **Security Implementer** covers the formal security language definition into a form, which can be deployed to agents, such as SNMP and firewall agents. These agents will then be in charge of implementing the configuration/rules files, as well as carrying out the test procedures while the system is in use. These tests are built with an understanding of how the security policy could be exercised, within its limits of operations, especially in simulating particular traffic patterns, such as following scenarios or scripts often used by intruders or malware. This type of approach offers a scientific approach to network security evaluation (Bajcsy, et al. 2004, p58).

The **Security Implementer** also determines the data gathered by systems such as from firewall or IDS's that posses login capability. From the information, the current security status will be evaluated against the organisation's requirements, using the **Verifier**, and a status report will be created, using the **Formatter**. This status report should offer the possibility to non-technical personnel to build a picture of the current security status of the organisation.

4. Methodology and Experimentations

As stated previously, an affective network security system needs not only to provide effective and efficient sensor and alert capabilities, but also the ability to act upon these alerts in order to reconfigure devices in the face of a threat. Therefore, building on from Zou, et al. (2003) and Moore, et al. (2003), who argue that detection is not enough in a secure system, the first stage of this research was to determine a framework and effectiveness of a dynamically self-reconfiguring system. Thus, this section focuses on two elements:

- Summarise the results from an initial proof-of-concept experiment.
- Presentation of the future experiment's design. This has, as its objective, the mapping of a security policy and a response procedure to existing network devices.

The common feature of both experiments is that the architecture of the reconfiguration system relies on low-cost or off-the-shelf solutions. The chosen system includes (Figure 2):

- **Snort** (Roesch 2005). This is one of the most flexible and powerful IDS's, and is avialable for many platforms.
- **Access Control Lists and Context Based Access Control Lists**. These are a feature in Cisco routers, and allow filtering of the data packets.
- **MySQL database**. This is used to quarantine in the database (MySQL-AB 2005)
- **ActivePerl**. This allows communications and actions between agents (ActiveState 2005).

4.1 Preliminary Experiment

Our first design was an attempt to identify the technologies required to build a system using networking elements to thwart network-based threats. The concept of merging capabilities is not new, especially in software applications. However, we argue that the benefits are much higher if the approach is successfully applied in a computer network; not only in terms of number of nodes protected at once, but also in terms of time and funding saved.

To evaluate our system, we assumed a scenario where a malware of some sort requires a given time to propagate through a network using multiple instances of a specific signature. We further assumed that the system could not block the threats in advance, as this might prevent legitimate activities. Taking action immediately allowed the system to stop the threat in its tracks.

From the logging database, the threat agent then determines that malicious activity was under way and extracts from the database information such as source IP address, destination IP, port numbers, and protocol in use. It then passes this information to the reconfiguration agent, which will create Access Control List statements to stop the stream of data from the offending node.

We tested this scenario with a mixture of legitimate and malicious traffic when all system elements on a single node as well as all elements dispatched on several nodes. In either situation, the legitimate activities were not pertubated. However, some instances of the signature were let

through due to communication latency within the mitigation system. Hence, logs must be audited with regards to the vulnerability exploited and services or software provided on the targeted node in order to determine if the threat has had consequences.

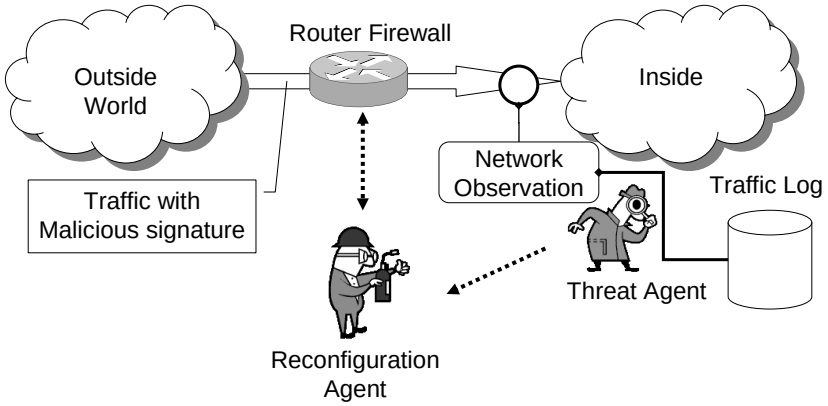

Figure 2: Preliminary Experiment's Abstract

4.2 Future Experiment Design

4.2.1 Motivations

Our first experiment used a network setup similar to the DARPA 99 project (Lippmann , et al. 2000), with an open network with no pre-established limitations or firewall for instance. The aim of the experiment, is to map a security policy and response procedure to existing networking elements. Furthermore, it evaluates the limit of a system relying only on an IDS for its intelligence gathering.

With respect to the framework presented in this paper, the ficticious organisation possesses English written polices. However, since security compiler and technology identifier are still under development, researchers will be in charge of creating the formal definitions and networking devices configurations.

4.2.2 Organisation's policies

The policy is based on FTP traffic, of which more information can be found in Appendix A. Samples of the written policy and detection and migation procedures for the FlashHosting Corporation (our fictious company) are:

A) **Ennouncing the acceptable usage policy**: FlashHosting Corporation is providing high bandwidth FTP hosting for its register clients only at the address ***%IP***. To ensure best quality of service FlashHosting Corporation will only accept to host a controlled list of file extensions. In addition, files should be at least 1MB but not larger than 125MB in size.

B) **Attack summary:** Intruders often target poorly secured FTP servers that they identify by scanning range of IP addresses. Once an address with FTP capability is found, they attempt to gain privileged access by trying multiple combinations of username and password. When access is granted, they will download files of interest and/or upload files to share them with others.

C) **Desired Response:** While scanning IP addresses or attempting to login respectively might be considered as suspicious activities, these are harmless on their own. However, strong evidences of their combination provide sufficient ground for a mitigation system to stop file transfers in progress and account owner being notified.

4.2.3 Parameters of the system

Figure 3 details the technical arrangements of the experiment, based on the formal definition of the policies and constructed according to the principles set in the framework.

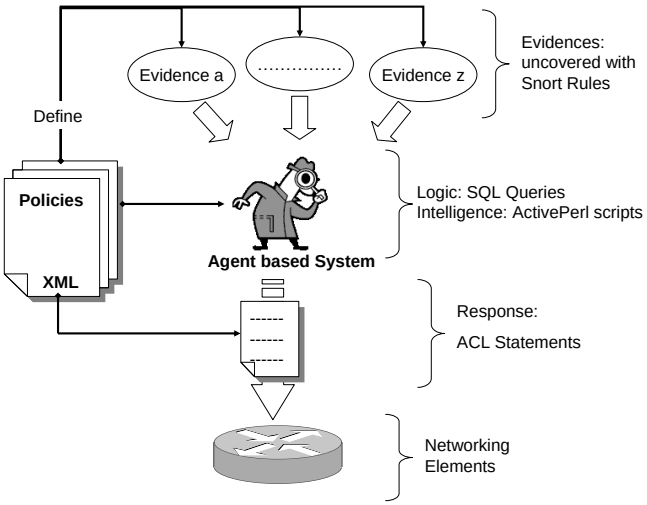

Figure 3: Network Setup

Thus, the formal language will define which evidences should be collected. The agent-based system will interpret these and determine if there is a threat. Whenever the system uncovers strong correlation between evidences, it will construct ACL statements and apply them to networking devices hence, reducing the damages caused to the system. Since this experiment is focused on protecting FTP service, the resulting ACL could be similar to the following snipet:

```
! Stop offending host with IP 172.16.20.12 to access FTP service on Server 192.168.10.11
! Allow any other host to access the server
Access-list 101 deny tcp host 172.16.30.12 192.168.10.11 0.0.0.0 eq FTP
Access-list 101 permit ip any any
```

Figure 4: ACL code Snipet

5. Conclusion and Furture Work

In this paper, we present a novel framework for computer networks security. This framework promotes security as a process. To achieve that aim, it incorporates the organisation's attributes such as legal liabilities, social and moral responsibilities, and organisational structure to build up a security definition that is tailored to the organisaiton's aims and objectivies. Furthermore, rather than leaving the technical personel solely in charge of interpreting the policy and implementing it on the various systems and assets of the organisation, the framework models the security requirements so they can be refined, identifies the technologies involved, and offer procedures that can be used to assess how well the live implementation matches the initial requirements.

Moreover, we reported the findings of an early experiment that indicates that rapid mitigation can be achieved without hindering legitimate activities. We also introduced the design of further experiment which challenge is to map policies to existing networking elements in addition to respond to threats in a timely manner.

Finally, we will provide our framework with the capability to compile English written documents into a formal language and identify the technologies suitable to enforce the organisation chosen policies.

6. Appendix: File Transport Protocol

Described in the request for comments number 959 (Postel, et al. 1985), FTP is a protocol that aims to promote sharing of data and or programmes, encourage the usage of remote computers, ensure transparency of the underlying file system, and ensure reliable transmission of data. Whether the transactions are carried out using a client graphical user interface or command lines, the exchange of information (i.e. username, password, etc.) or commands (“mode”, “pathname”, etc.) between the communicants appear in clear text in the network traffic payload. Evidently, these facts might cause privacy concerns; Nevertheless, FTP remains a very popular way to distribute files. Sun Microsystems and the Mozilla foundation, for instance, rely on this application to keep updated or support their various products. Moreover, hosting companies might offer FTP facility along with their web pages hosting packages. Thus, FTP servers are assets that ought to be carefully looked after.

In order to ease the serving of files, administrator might enable the server to accept anonymous login. In turn, this type of accounts will be restricted to downloading files. Predictably, this is not enough to stop intruders from trying to obtain privileged access or subverting the server. In order to do so intruders do not have to be technical experts. They can design their attacks by combining scripts or ready-made piece of software geared for that purpose. Glenn (2003, p1) argues that the availability of such code over the Internet explains the rapid decreasing time-window between public announcements of vulnerabilities and exploits which have led to the increase of the numbers of intrusions. Publicly available FTP services reside on high bandwidth link and thus offer a great medium for intruders to distribute their loot at someone else’s expenses (Chuvakin 2004, p1). As always, fixes are applied after problems have been noticed and dully identified. Unfortunately, these actions might not happen soon enough to preserve assets such as corporate image Chuvakin 2004, p24).

7. Acknowledgements

The authors would like to thank Professor Rao Bhamidimarri, Dean, to make founding available for research as well as Professor Jon Kerridge and Dr Bob Rankin, former and current Head of School respectively, for assigning grants to this research program.

References

ActiveState (2005) "The industry-standard Perl distribution for Linux, Solaris, and Windows", [online], http://www.activestate.com/Products/ActivePerl/

F. M. Avolio (1998) "Putting it together a multi-dimensional approach to Internet security", netWorker, 2, 2, 15-22

R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, K. Levitt, B. Lindell, P. Liu, D. Miller, R. Mundy, C. Neuman, R. Ostrenga, V. Paxson, P. Porras, C. Rosenberg, J. D. Tygar, S. Sastry, D. Sterne and S. F. Wu (2004) "Cyber defense technology networking and evaluation", Communications of the ACM, 47, 3, 58-61

I. Bashir, E. Serafini and K. Wall (2001) "Securing network software applications: introduction", Communications of the ACM, 44, 2, 28 - 30

E. Bertino, S. Castano and E. Ferrari (2001) On specifying security policies for web documents with an XML-based language,

A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente and D. Villela (1999) "A survey of programmable networks", ACM SIGCOMM Computer Communication Review, 29, 2, 7 - 23

A. Chuvakin (2004) "Issues Discovering Compromised machines", [online], SecurityFocus, http://www.securityfocus.com/printable/infocus/1808

T. Corbitt (2002) "Protect you computer system with a security policy", Management Services, 46, 5, 20-21

D. Danchev (2003) "Building and Implementing a Successful Information Security Policy", [online], http://www.windowsecurity.com/pages/article_p.asp?id=1218

B. Fraser (1997) "Site Security Handbook", [online], Internet Engineering Taskforce, http://www.faqs.org/rfcs/rfc2196.html

M. Glenn. (2003). A summary of DoS/DDoS Prevention, monitoring and Mitigation Techniques in a Service Provider Environment (White Paper): SysAdmin, Audit, Network, Security Institute
C. Huseyin, B. Mishra and S. Raghunathan (2004) "A model for evaluating IT security investments", Communications of the ACM, 47, 7, 87 - 92
S. Ioannidis, A. D. Keromytis, S. M. Bellovin and J. M. Smith (2000) Implementing a distributed firewall, 7th ACM conference on Computer and communications security, Athens, Greece
K. Julisch (2003) "Clustering intrusion detection alarms to support root cause analysis", ACM Transactions on Information and System Security (TISSEC), 6, 4, 443 - 471
S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum and M. Frantzen (2003) "Analysis of Vulnerabilities in Internet Firewalls", Computers & Security, 22, 3, 214 - 232
R. Lippmann , J. W. Haines, D. J. Fried, J. Korba and K. Das (2000) "The 1999 DARPA Off-Line Intrusion Detection Evaluation", Computer Networks, 34, 4, 579-595
D. Moore, C. Shannon, G. M. Voelker and S. Savage (2003) Internet Quarantine: Requirements for Containing Self-Propagating Code, IEEE INFOCOM 2003, San Francisco, California, USA
MySQL-AB (2005) "The world's most popular open source database", [online], http://www.mysql.com/
P. Ning and D. Xu (2003) Learning attack strategies from intrusion alerts, Conference on Computer and Communications Security, Washington D.C., USA
V. Paxson (1999) "Bro: A System for Detecting Network Intruders in Real-Time", Computer Networks, 31, 23-24, 2435-2463
J. Postel and J. Reynolds (1985) "Request for Comment File Transfer Protocol 959", [online], Network Working Group, http://www.ietf.org/rfc/rfc959.txt
M. Roesch (2005) "Snort - the de facto standard for Intrusion detection / prevention", [online], http://www.snort.org/
G. Rosamond. (2004). Building a more Secure Network: SANS
G. A. Santana Torrellas and L. A. Villa Vargas (2003) Modelling a flexible network security systems using multi-agents systems: security assessment considerations, 1st international symposium on Information and communication technologies, Dublin, Ireland
D. K. Smetters and R. E. Grinter (2002) Moving from the design of usable security technologies to the design of useful secure applications, 2002 workshop on New security paradigms, Virginia Beach, Virginia, USA
F. Stajano and R. Anderson (2002) "The Resurrecting Duckling: Security Issues for Ubiquitous Computing", [online], IEEE, http://www.computer.org/security/supplement1/sta/print.htm
S. Staniford, V. Paxson and N. Weaver (2002) How to Own the Internet in Your Spare Time,
Sygate (2005) "Sygate Personal Firewall Pro", [online],
http://smb.sygate.com/products/pspf/pspf_ov.htm
Symentec (2005) "Symentec Client Security 2.0", [online],
http://www.symantec.com/smallbiz/scs_sbe/index.html
S. Timms, C. Potter and A. Beard. (2004). Information Security Breaches Survey 2004 (Technical): UK Government
J. Viega and M. Messier (2004) "Security is Harder than You Think", Queue, 2, 5, 60-65
H. Yan and R. Frenz (2004) "An adaptive and scalable architecture for rule-based real-time analysis of network traffic", [online], www.ece.cmu.edu/~rfrenz/papers/yan_frenz2003.pdf
C. C. Zou, L. Gao, W. Gong and D. Towsley (2003) Monitoring and early warning for internet worms, 10th ACM Conference on Computer and Communications Security, Washington D.C., USA