Security Vulnerabilities and System Intrusions – The need for Automatic Response Frameworks

M.Papadaki, G.Magklaras, S.M.Furnell and A.Alayed

Corresponding Author

S.M. Furnell
Research Co-ordinator

Network Research Group
Department of Communication and Electronic Engineering
University of Plymouth
Plymouth
United Kingdom

Tel: +44 1752 233521
Fax: +44 1752 233520

Email : sfurnell@plymouth.ac.uk

Other Authors

Maria Papadaki Research Student Network Research Group Department of Communication and Electronic Engineering University of Plymouth Plymouth United Kingdom Tel: +44 1752 233520 Fax: +44 1752 233520 Email : mpapa@jack.see.plym.ac.uk	George Magklaras Research Student Network Research Group Department of Communication and Electronic Engineering University of Plymouth Plymouth United Kingdom Tel: +44 1752 233520 Fax: +44 1752 233520 Email : semaphore@jack.see.plym.ac.uk
Abdulaziz Alayed Research Student Network Research Group Department of Communication and Electronic Engineering University of Plymouth Plymouth United Kingdom Tel: +44 1752 233520 Fax: +44 1752 233520 Email : aal-ayed@plymouth.ac.uk	

Security Vulnerabilities and System Intrusions – The need for Automatic Response Frameworks

M.Papadaki, G.B.Magklaras, S.M.Furnell and A.Alayed

Network Research Group, Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, United Kingdom

nrg@jack.see.plym.ac.uk

Abstract

Addressing security vulnerabilities and system intrusions can represent a significant administrative overhead in current computer systems. Although technologies exist for both vulnerability scanning and for intrusion detection, the problems typically require some form of human intervention before they can be rectified. Evidence suggests that, in many cases, this can lead to omissions or oversights in terms of protection, as administrators are forced to prioritise their attention to security amongst various other tasks (particularly within smaller organisations, where a dedicated security administration function is unlikely to be found). As a result, mechanisms for automated response to the issues are considered to be advantageous. The paper describes the problems associated with vulnerability analysis and intrusion response, and then proceeds to consider how, at a conceptual level, the issues could be addressed within the framework of a wider architecture for intrusion monitoring.

Keywords: Vulnerability analysis, intrusion detection, intrusion response.

Introduction

The widespread use of Internet systems by organisations of all types means that the problem of IT security has never been more prominent. It would be no exaggeration to say that many organisations and individuals are reliant upon these systems, their correct operation and the data they contain. Despite their critical role, however, evidence has shown that systems are often vulnerable to various forms of abuse – breaching their security and resulting in intrusions. The problem of security breaches has substantially increased in recent years. In the CSI/FBI 2000 Computer Crime and Security Survey, financial losses due to computer security breaches mounted to $377,828,700, while the average annual total over the three years prior to 2000 was $120,240,180 [1].

An *intrusion* is the series of actions taken by an attacker against a target to achieve an unauthorised result. In order to fulfill this objective, the attacker must exploit a computer or network *vulnerability*, which represents the weakness of the system that allows unauthorised action to be taken [2]. For example, a well-known system

vulnerability is the use of weak, default or even blank passwords [3]. These offer the opportunity for effortless access by attackers, who will routinely attempt to gain access to systems by trying default passwords, and then easily guessable ones. Only if these are unsuccessful will they need to resort to more sophisticated methods. Once inside, attackers can exploit other widely known vulnerabilities to increase their access (e.g. to attain root / administrator privileges).

This paper considers the dual problems of addressing security vulnerabilities and responding to intrusions that may result from their exploitation. In current systems, both elements can be seen to represent an administrative burden, with responsibility falling to system administration staff. In many cases, this may lead to omissions and prioritisation problems, as the same staff will often have numerous other responsibilities. It is considered that this issue is likely to be particularly acute within smaller organisations, due to the typical lack of dedicated IT security management staff. The discussion begins with an examination of the administrative problems posed by security vulnerabilities, in terms of the efforts required to identify and resolve an ever-increasing range of known problems. It then proceeds to consider the further considerations involved if it becomes necessary to respond to a suspected intrusion incident – which will often result from the exploitation of a vulnerability. The desirability of automated responses is recognised in both cases, leading to consideration of how an automated framework could be used to reduce the burden upon system administrators.

The administrative problem of security vulnerabilities

It is recognised that responding to both security vulnerabilities and detected intrusions can represent a significant administrative overhead. In the case of vulnerabilities, for example, there are associated overheads at two levels:

(a) ensuring awareness of vulnerability existence;
(b) being able to take appropriate corrective action to resolve them (e.g. installing software upgrades and patches).

Even though many exploits are based upon vulnerabilities that have been known for some time, the problem is a difficult one to keep on top of. Many software developers routinely release patches that enable known bugs and vulnerabilities in their products to be rectified – in some cases this happens before particular weaknesses have become publicly known, whilst in others it is in response to a problem being reported. As a result, the situation in many cases is that simple maintenance activity by system administrators is all that would be required to plug the holes. However, despite this, the problems clearly remain. The SANS Institute has identified several reasons why this may be the case [4]:

- 1.2 million new computers are added to the Internet every month;

- there is a lack of security experts to address the problems;
- the number of vulnerabilities continues to grow and there is no priority list for dealing with them.

From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available – this is what the users expect and complaints will quickly occur if this is not the case. So, unless installing a patch is explicitly required to ensure that this is the case, then the task is likely to be given a lower priority.

Looking at the number of warnings that are issued, it is easy to see how administrators might downgrade the importance of responding to them immediately. This can be illustrated by considering the security bulletins issued by Microsoft Corporation in relation to its product range. When vulnerabilities are identified in Microsoft products, the company works to develop a solution and then issues an advisory bulletin when a software patch or upgrade is available for download. The graph in Figure 1 summarises the number of security bulletins issued per month, between January 1999 and September 2000 (statistics obtained from http://www.microsoft.com/technet/security/current.asp).

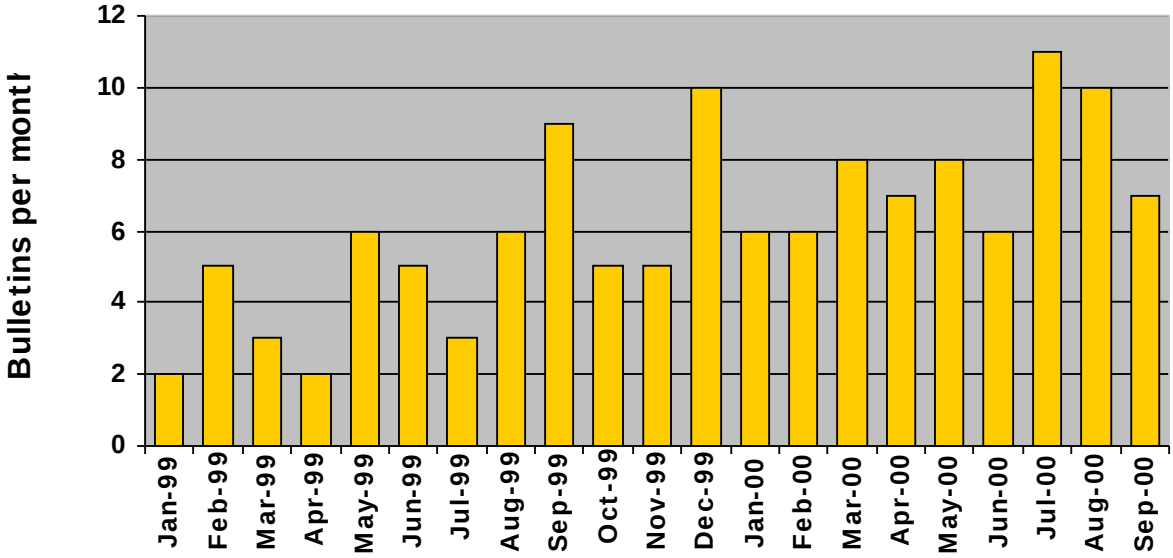

Figure 1 : Microsoft Security Bulletins (January 1999 to September 2000)

It can be seen from the graph that the number of security bulletins issued ranges from two per month up to eleven per month (the average was 6 per month over the 21 month period). This might not be so bad if the associated patch was being installed on just a single system, but in some cases an organisation's IT and network configuration may dictate that the administrator must go around and update a number of individual systems in turn (which could obviously become quite time consuming). In some cases, the number of systems may run into the thousands, whereas the administration team may number less than ten. Relating this to the number of patches released per month, this could lead to each administrator having to patch about 20 machines per day (assuming

the average of 6 patches per month and that all systems required them). It should also be remembered that these bulletins are only those related to Microsoft products. Where an organisation's IT set up is based upon a heterogeneous, multi-vendor configuration, security advisories from other sources would also have to be taken into consideration.

So, in view of all this, it can be appreciated that administrators might start out with good intentions, responding to each advisory as it arrives. However, this could quickly become burdensome and so the decision may be taken to batch them up and respond to them on a less frequent basis. Whilst this makes good administrative sense, it is less sensible from a security perspective. Once an advisory has been issued, the information about the associated vulnerability is available to anyone – and any hackers who were not aware of it before will certainly have access to it from then on. As such, any systems in which the weakness has not been addressed are exposed to a greater level of risk than before the advisory was made.

So what is the effect of not installing the available fixes? According to Attrition.org, 99% of the 5,823 web site defacements that occurred during 2000 were as a result of failure to patch known vulnerabilities for which the fixes were already available [5].

Intrusion response

If a vulnerability is successfully exploited, a system intrusion is likely to result – which will require some form of consequent response. From this perspective, the issues of vulnerability analysis and intrusion response are related areas, separated only by the occurrence of an incident.

Intrusion response can be specified as the process of counteracting the effects of an intrusion. It includes the series of actions taken by an Intrusion Detection System, which follow the detection of a security-related event. It is important to note that consideration is not only given to taking action after an intrusion has been detected, but also when events of interest take place and raise the alert level of the system. That is the early stages of an attack, when the system is suspecting the occurrence of an intrusion, but is not yet confident enough.

It is possible to distinguish two main approaches to intrusion response, namely human/organisational approaches and technical methods. The former are those that involve human processes and organisational structures, and may include actions such as reporting an incident to the police or invoking disciplinary procedures (e.g. in cases where internal personnel are responsible). By contrast, technical responses involve the use of functional techniques and software-based methods. These technical actions can themselves be further sub-classified, into either passive or active forms of response [6]:

- **Passive responses**: aim to notify other parties (administrators - users) about the occurrence of an incident, relying on them to take further actions about it. Alarms,

notifications and SNMP Traps are the most common passive responses. Passive actions are the most common response options in commercial IDS systems.

- **Active responses**: are the actions taken by a process or system to encounter the incident that has occurred. Those actions might include collecting more information about the incident, limiting permitted user behaviour, or blocking IP traffic through firewalls and routers.

Within these categories there are myriad individual response actions that could be pursued and some decision making ability is required when a suspected incident presents itself. However, although the type of incident will suggest a range of possible responses, the classification of incident alone does not provide enough information to determine which one(s) are actually appropriate. The *specific* response(s) to initiate will depend upon a number of factors, which collectively identify the context in which the incident has occurred. This idea is illustrated in Figure 2.

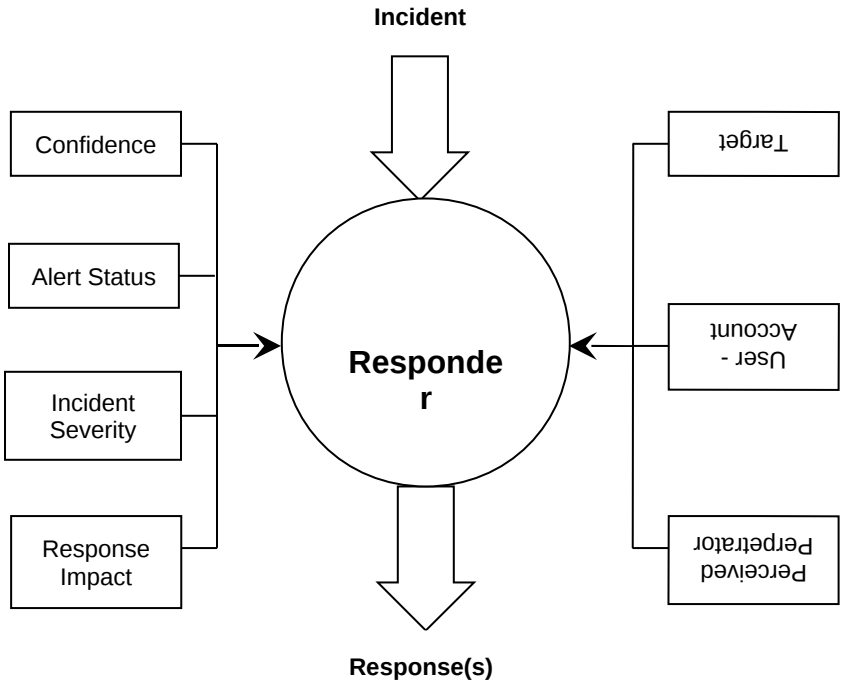

Figure 2 : Factors influencing intrusion response

As the diagram shows, the *incident* is the trigger for the response and still represents the principal influence over what should be done. However, the other influencing factors that also need to be considered are as follows:

- **Confidence:** how many monitored characteristics within the system are suggestive of an intrusion having occurred?

- **Alert status:** what is the current status of the monitoring system, both on the suspect account / process and in the system overall?

- **Incident severity:** what impact has the incident already had upon the confidentiality, integrity or availability of the system and its data? How strong a response is required at this stage?

- **Response impact:** what would be the impact of initiating a particular form of response? How would it affect a legitimate user if the suspected intrusion was, in fact a false alarm? Would there be any adverse impacts upon other system users if a particular response action were taken?

- **Target:** what system, resource or data appears to be the focus of the attack. What assets are at risk if the incident continues or is able to be repeated?

- **User account:** if the attack is being conducted through the suspected compromise of a user account, what privileges are associated with that account?

- **Perceived perpetrator:** does the evidence collected suggest that the perpetrator is an external party or an insider?

At the heart of Figure 2 was an entity referred to as the *responder*. This is the element that will assess the various factors in order to select and invoke the required response(s). Although a great deal of work has been done in the area of automated intrusion detection, current systems are able to do very little in terms of automated response when they suspect a problem. So, in current systems, the responder role is likely to be taken by a system administrator. However, there are practical limits to the effectiveness of this approach. Firstly, the administration of increasingly large and complicated IT infrastructures becomes correspondingly more cumbersome. Secondly, the widespread use of automated scripts to generate attacks of a distributed nature [7] can render the speed of traditional response methods inadequate. As with vulnerability analysis and resolution, therefore, the administrative burden may again mean that the handling of intrusion response becomes sidelined - although, of course, there may be more incentive to respond to an intrusion because it represents a vulnerability that has already been exploited.

Automated response frameworks

In order to assist in resolving the problem of administrative overhead, some form of automated response framework is desirable. For vulnerabilities, it can be observed that there are already numerous tools available to assist in the task of scanning systems to identify potential holes. However, this only goes part of the way to addressing the problem. It relieves the administrators of having to have the detailed knowledge of system security necessary to identify weaknesses, but it still requires their attention to both run an analysis and take consequent corrective actions. Although some scanning

software includes functionality for fixing problems identified, the current approaches are limited - minor system configuration weaknesses can be rectified, but many vulnerabilities require more substantial action than this. Given that vulnerabilities and intrusions are linked issues, it makes sense for vulnerability analysis and resolution to form part of an overall intrusion monitoring approach.

Figure 3 illustrates the conceptual architecture of the Intrusion Monitoring System (IMS), a research prototype that the authors are currently developing. IMS is an architecture for intrusion monitoring and activity supervision, based around the concept of a centralised host handling the monitoring of a number of networked client systems. Intrusion detection in the system is based upon the comparison of current user activity against both historical profiles of 'normal' behaviour for legitimate users and intrusion specifications of recognised attack patterns. The architecture is comprised of a number of functional modules, addressing data collection and response on the client side and data analysis and recording at the host.

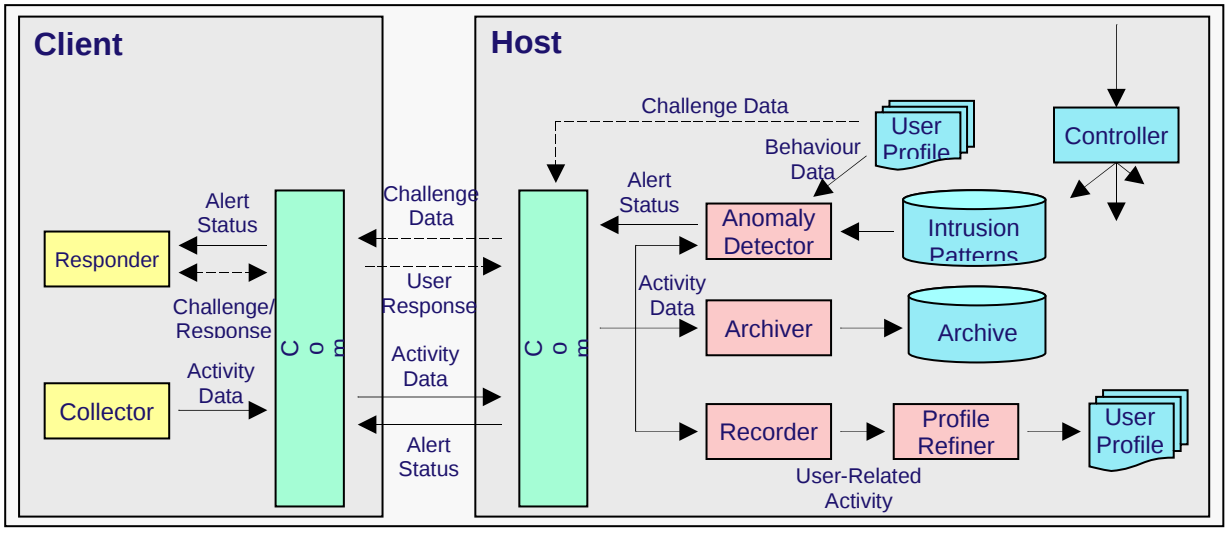

Figure 3 : The Intrusion Monitoring System architecture

The full architecture is described in [8] but, from the perspective of the current discussion, the relevant modules are the *Collector*, *Anomaly Detector* and *Responder* – which can be used to perform activity monitoring (to identify intrusions) and vulnerability scanning, as well as appropriate follow-up actions in the event of problems.

The *Collector* is responsible for obtaining information from individual monitored client systems. In terms of activity monitoring, this information may relate to user data such as applications and files accessed, keystroke data (for biometric analysis) and resource usage statistics. From the perspective of vulnerability scanning, the *Collector* could also take on the role of obtaining system configuration details and the like, which would then be sent for subsequent analysis.

The *Anomaly Detector* resides on the host side and is the main recipient of the *Collector's* data. For user activity, it compares the information against historical profiles of 'normal' behaviour (e.g. frequently used applications, typing style) to identify anomalies that may indicate either an impostor or misuse by a legitimate user. In addition, generic intrusion specifications will be used to compare activities against known patterns of misuse – with a match triggering some form of alert. From a vulnerability analysis perspective, the *Anomaly Detector* will compare the collected scan data against a database of known weaknesses. In the event of problems, the *Anomaly Detector* will increase the alert status of the monitoring system and interact with the *Responder* module.

The *Responder* provides an automated facility for dealing with suspected problems. There are numerous forms of response that it would be possible to allow a system to initiate under automatic control. A small selection of ideas are listed below:

- further investigation of the incident via data collected in audit log files;
- increasing the level monitoring and/or auditing;
- issuing a challenge for further authentication;
- limiting permitted user behaviour;
- delaying (or lowering priority of) intruder's session / process;
- termination (or suspension) of the anomalous session / process.

It is the *Responder* that would be responsible for assessing and weighting the contextual factors that would determine the appropriate response option(s) for a given incident occurrence. As such, the *Responder* (like the *Anomaly Detector*) requires an element of intelligent analysis and decision-making.

In the vulnerability analysis context, the decision about what to do is potentially clear-cut, but the issue remains about when to do it. The *Responder* could conceivably take the role of coordinating and conducting updates on the affected client systems in order to resolve problems identified. A library of fixes, updates and patches would be accumulated and maintained on the host side and then issued to clients as necessary.

The description presented here proposes the solution at a conceptual level only. In practice, of course, the associated mechanisms would be far more involved and elements represented as single boxes or flows within Figure 3 would potentially be realised as a large number of sub-processes. Some issues, such as how the system can maintain awareness of new vulnerabilities and acquire associated patches, remain unresolved and require further investigation. Other aspects, such as the anomaly detection methods and response framework, are already the focus of active research.

Conclusions

Automated response approaches such as those described have the potential to significantly reduce the burden on system administrators. Indeed, within the framework

of an approach such as that proposed with IMS, the whole process of intrusion prevention, detection, response and resolution could be addressed.

Although the proposed approaches have the advantages identified, it is recognised that there is also a risk that any automated action taken could be incorrect. In the case of vulnerabilities, attempts to rectify security weaknesses or install software patches on the fly could adversely affect the operation of the system and/or cause incompatibility with existing elements. In the case of intrusion response, the automatic invocation of an inappropriate method could result in insufficient action being taken or, alternatively, could interrupt or deny service to a legitimate activity. As such, both are aspects that require careful configuration and their degree of permitted autonomy would strongly depend upon the nature of the system they were protecting.

The design of the automated response frameworks is the focus of ongoing research by the authors. Further details of the associated architectural approaches and implementation experiences will be reported in future publications.

References

[1] CSI. 2001. "Financial losses due to Internet intrusions, trade secret theft and other cyber crimes soar", CSI Press Release, 12 March 2001. http://www.gocsi.com/prelea_000321.htm

[2] Howard, J. 1997. "An Analysis of Security Incidents on the Internet 1989 – 1995", PhD thesis. Carnegie Mellon University, April 1997. http://www.cert.org/research/JHThesis

[3] SANS Institute. 2001. "How To Eliminate The Ten Most Critical Internet Security Threats. The Experts' Consensus**",** Version 1.32, 18 January 2001. http://www.sans.org/topten.htm.

[4] Noack, D. 2000. "The Back Door Into Cyber-Terrorism", APBnews.com Report, 2 June 2000.

[5] CNET. 2001. "Patchwork Security - Software "fixes" routinely available but often ignored", CNET News.com report. 24 January 2001. http://news.cnet.com/news/0-1007-201-4578373-0.html

[6] Bace, R. and Mell, P. 2001. "NIST Special Publication on Intrusion Detection Systems", National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/drafts/idsdraft.pdf, February 12 2001.

[7] Cheung, S. and Levitt, K.N. 1997. "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection", Proceedings of the New Security Paradigms Workshop, Langdale,Cumbria UK, September 23 - 26, 1997, http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/

[8] Furnell, S.M. and Dowland, P.S. 2000. "A conceptual architecture for real-time intrusion monitoring", *Information Management & Computer Security*, Vol. 8, No. 2, pp65-74.