Operational Characteristics of an Automated Intrusion Response System

Maria Papadaki[1], Steven Furnell[1], Benn Lines[1], and Paul Reynolds[2]

[1] Network Research Group, University of Plymouth,
Drake Circus, Plymouth, United Kingdom
info@network-research-group.org
[2] Orange Personal Communications Services Ltd,
St James Court, Great Park Road, Bradley Stoke,
Bristol, United Kingdom.

Abstract. Continuing organisational dependence upon computing and networked systems, in conjunction with the mounting problems of security breaches and attacks, has served to make intrusion detection systems an increasingly common, and even essential, security countermeasure. However, whereas detection technologies have received extensive research focus for over fifteen years, the issue of intrusion response has received relatively little attention - particularly in the context of automated and active response systems. This paper considers the importance of intrusion response, and discusses the operational characteristics required of a flexible, automated responder agent within an intrusion monitoring architecture. This discussion is supported by details of a prototype implementation, based on the architecture described, which demonstrates how response policies and alerts can be managed in a practical context.

1 Introduction

Ever since the commercialisation of the Internet, there has been a substantial growth in the problem of intrusions, such as Denial of Service attacks, website defacements and virus infections [1]. Such intrusions cost organisations significant amounts of money each year; for example, the 2003 CSI/FBI Computer Crime and Security Survey [2] reported annual losses of $201,797,340 from 530 companies questioned. Although these results suggest that the cost of attacks has decreased for the first time since 1999, it is still significant amount, representing a 101.55% increase compared to 1997 [3].

As a defence against such attacks, intrusion detection technologies have been employed to monitor events occurring in computer systems and networks. Intrusion detection has been an active research area for more than 15 years [4,5], and merits a wide acceptance within the IT community [6;3]. However, detecting intrusions is only the first step in combating computer attacks. The next step involves the counteraction of an incident and has so far been largely overlooked [7;8]. The CSI/FBI survey suggests a declining trend amongst organisations to address vulnerabilities, or report incidents to law enforcement since 1999 [2]. Although the percentage of respondents,

A. Lioy and D. Mazzocchi (Eds.): CMS 2003, LNCS 2828, pp. 65–75, 2003.

who patched vulnerabilities after an incident, was reasonably high, it was still decreased by 2% when compared to the respective figure of 1999, while about 50% of the respondents chose not to report the incident at all. Even if vulnerability patching and incident reporting are only two aspects of responding to intrusions, the lower percentages suggest a lack of effective response policies and mechanisms within organisations.

A principal reason for this problem is likely to be the administrative overhead posed by response procedures. At the moment, the detection of a suspected intrusion typically triggers a manual intervention by a system administrator, after having received an alert message from the intrusion detection system. The IDS can additionally assist the incident response process, by providing the details of the attack, saved in a log file [9]. However, responding manually to intrusions is not necessarily an easy task, as it may involve dealing with a high number of alerts and notifications from the IDS [10], ensuring awareness of security bulletins and advisories from incident response teams, and taking appropriate actions to resolve each of the alerts reported. From the system administrator's perspective, the main requirement is to ensure that the system remains operational and available. Thus, unless resolving a detected incident is explicitly required to ensure that this is the case, the task of responding is likely to be given a lower priority.

The importance of timely response has been demonstrated by Cohen [11] in his simulation of attacks, defences and their consequences in complex 'cyber' systems. These showed that, if skilled attackers are given 10 hours between being detected, and generating a response, then they have an 80% chance of a successful attack. When that time interval increases to 20 hours, the rate of success rises to 95%. After 30 hours the skill of the system administrator makes no difference, as the attacker will always succeed. However, if the response is instant, the probability of a successful attack against a skilled system administrator becomes almost zero. This shows not only the importance of response, but also the relationship between its effectiveness and the time it is initiated.

At the time of writing, the degree of automation in current IDS is very low, offering mostly passive responses (i.e. actions that aim to notify other parties about the occurrence of an incident and relying on them to take further action). In contrast, active responses (actions taken to counter the incident that has occurred) either have to be initiated manually or may not be offered at all. Lee [12] found that even if IDS products offer active responses, they are not trusted by administrators, mainly due to the likely adverse effects in the event of them being falsely initiated. In spite of the potential problems, practical factors suggest that automated response methods will become increasingly important. For example, the widespread use of automated scripts to generate distributed attacks [13] can offer very limited opportunity to respond, and further diminishes the feasibility of doing so manually. Thus, there is a need for the adoption of automated response mechanisms, which will be able to protect system resources in real time and, if possible, without requiring explicit administrator involvement at the time.

As an effort to enhance the effectiveness of automated response and reduce its adverse effects in false rejection scenarios, an automated response framework has been devised. The aim is to enable accurate response decisions to be made autonomously, based on the nature of the attack and the context in which it is occurring (e.g. what applications are running, what account is being used, etc.). The

remainder of this paper describes the concept of the Responder, followed by details of a prototype implementation that demonstrates the approach in practice.

2 The Intrusion Monitoring System (IMS)

IMS has been the focus of research within the authors' research group for several years and is a conceptual 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 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 addresses data collection and response on the client side, and data analysis and recording at the host. The elements of the architecture that are relevant to the discussion presented in this paper are illustrated in Figure 1. The main modules of IMS have already been defined in earlier publications [14], and interested readers are referred to these for associated details. In this paper, specific focus will be given to the modules related to intrusion response.

The Responder is responsible for monitoring the Alerts sent from the Detection Engine (note: this module was referred to as the Anomaly Detector in previous papers) and, after considering them, in conjunction with other contextual factors, taking appropriate actions where necessary. If the actions selected by the Responder need to be performed on the client side, a local Responder Agent is responsible for initiating and managing the process. Without providing an exhaustive list, examples of actions that could be performed at the client side include correcting vulnerabilities, updating software, issuing authentication challenges, limiting access rights and increasing the monitoring level.

The Responder utilises a variety of information in order to make an appropriate decision. This is acquired from several other elements of IMS, including the Detection Engine, the Collector, the Profiles, and the Intrusion Specifications. The possible contributions from each of these sources are described below.

As well as indicating the type of suspected incident, the Detection Engine is also able to directly inform the Responder about the intrusion confidence, the current alert status of the IDS, the source of the alert that triggered the detection, information about the perceived perpetrator(s) and the target involved.

The Collector is able to provide information about current activity on the target system (e.g. applications currently running, network connections currently active, applications installed etc.). This information can be used to minimise the disruption of legitimate activity, by making sure that no important work at the target gets lost, or no important applications are ended unnecessarily, as a result of selected response actions. It can also be used for cases of compromised targets when information about them needs to be reassessed. For example, the determination of whether unauthorised software (sniffing software / malware) has been installed will be vital information for the response decision process. In that way the negative impacts of responses can be minimised and the response capability enhanced as much as possible.

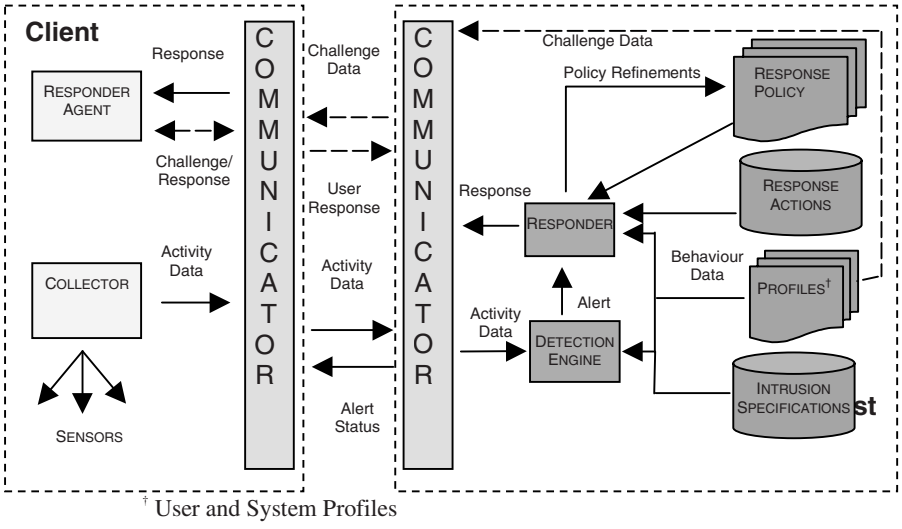


Fig. 1. The Intrusion Monitoring System (IMS)

The Profiles contain information about users and systems, both of which can provide some information in the context of response decisions:

- User profiles: If the incident involves the utilisation of a user account, then the corresponding user profile can indicate aspects such as the privileges and access rights associated with it.
- System profiles: These relate to system characteristics, such as versions of operating systems and installed services, the expected load at given hours/periods, the importance of the system within the organisation (e.g. whether it holds sensitive information or offers critical services), its location on the network etc.

Finally, Intrusion Specifications contain information about specific types of intrusions and their characteristics - such as incident severity rating, ratings of likely impacts (e.g. in terms of confidentiality, integrity and availability), and the speed with which the attack is likely to evolve [15]. Once the Detection Engine has indicated the type of incident that it believes to have occurred, additional information can be retrieved from the specifications to obtain a comprehensive view of the incident (all of which would again influence the response selection).

Having gathered all of the available information, the actions that should be initiated in different contexts are then specified in the Response Policy. In the first instance, the Response Policy would need to be explicitly defined by the system administrator; however, it could also be refined over time to reflect practical experience. For example, if a particular response is found to be ineffective against a particular situation, then the policy could be updated to account for this. It is envisaged that this refinement could be initiated manually by the system administrator, as well as automatically by the system itself. Further information about this process is given in the next section.

3 Operational Characteristics of the Responder

In order to enable increasingly automated responses, and reduce the risks associated with using active response methods, the architecture incorporates techniques to improve the flexibility of the response process when compared to approaches in current IDS. Specifically, the proposed Responder includes the ability to:

- adapt decisions according to the current context; and
- assess the appropriateness of response actions before and after initiating them.

The concept of adaptive decision-making relates to the requirement for flexibility in the response process. A fundamental principle of the proposed approach is that response decisions should vary depending upon the context in which an incident has occurred (i.e. a response that is appropriate to a particular type of incident on one occasion will not necessarily be appropriate if the same incident was to occur again under different circumstances). The previous section described how the Responder draws upon information from a number of other sources within the IMS framework. This enables the system to determine the overall context in which an incident has occurred, including considerations such as:

- the overall alert status of the IDS at the time of the new incident;
- whether the incident is part of an ongoing series of attacks (e.g. how many targets have already been affected? Which responses have already been issued?);
- the perpetrator of the attack (is there enough information to suggest a specific attacker? Is he/she an insider/outsider? Has he/she initiated an attack before? How dangerous is he/she? What attacks is he likely to attempt?);
- the current status of the target (e.g. is it a business critical system? What is its load at the moment? Is there any information or service that needs to be protected? What software/hardware can be used for response?);
- the privileges of the user account involved (e.g. what is the risk of damage to the system?);
- the probability of a false alarm (how reliable has the sensor/source that detected the incident been in the past? What is the level of confidence indicated by the Detection Engine about the occurrence of an intrusion?);
- the probability of a wrong decision (how effective has the Responder been so far? Have these responses been applied before in similar circumstances?).

Having assessed the above factors, response decisions must then be adapted to the context accordingly. For example, if the incident has been detected on a business critical system, and the Detection Engine has indicated a low confidence, then the selection of a response with minimal impact upon the system would represent the most sensible course of action. That decision minimises the chance of critical operations being disrupted in the case of an error alert. However, if the same scenario occurred in conjunction with previous alerts having already been raised (i.e. indicating that the current incident was part of a series of attacks), or if the overall alert status of the IDS was already high, then a more severe response would be

warranted. More comprehensive information about this decision process, and the information that would be assessed, is presented in earlier publications [15; 16].

The other novel feature of the Responder is its ability to assess the appropriateness of response actions. This can be achieved in two ways; firstly by considering the potential side effects of a response action, and secondly by determining its practical effectiveness in containing or combating attacks.

As previously identified in the introduction, the problem of side effects is a particular concern in the context of using active responses, because they have the potential to adversely affect legitimate users of the system. As a result, this needs to be considered before the Responder chooses to initiate a given action. There are a number of characteristics that would be relevant in this context:

- the transparency of the response action. In some cases it might be preferable to issue responses that do not alert the attacker to the fact that he/she has been noticed, whereas in others it could be preferable to issue a response that is very explicit.
- the degree to which the action would disrupt the user to whom it is issued. This is especially relevant in the context of a response action having been mistakenly issued against a legitimate user instead of an attacker. In situations where the Detection Engine has flagged an incident but expressed low confidence, it would be desirable to begin by issuing responses that a legitimate user would be able to overcome easily.
- the degree to which the action would disrupt other users, or the operation of the system in general. Certain types of response (e.g. termination of a process, restriction of network connectivity) would have the potential to affect more than just the perceived attacker, and could cause reduced availability to other people as well. As such, the Response Policy may wish to reserve such responses only for the most extreme conditions.

Each of these factors would need to be rated independently, and the information would be held in the database of available response actions (previously illustrated in Figure 1). The consideration of the ratings could then be incorporated into the response selection process as appropriate, and indeed during the formulation of the Response Policy by the system administrator. In addition to assessing the side effects, each response could also usefully be given an associated rating to indicate its perceived strength (which could inform the Responder and the administrator about its likely 'stopping power' in relation to an attacker).

The second factor that would influence the appropriateness of a response in a particular context would be whether it had been used in the same context before. If the Responder keeps track of its previous response decisions, then they can subsequently be used as the basis for assessing whether the response actions were actually effective or not. This requires some form of feedback mechanism, which can then be used to refine the Response Policy. It is envisaged that feedback could be provided in two ways: explicitly by a system administrator, and implicitly by the Responder itself. In the former case, the administrator would inspect the alert history and manually provide feedback in relation to the responses that had been selected to indicate whether or not they had been effective or appropriate to the incident. By contrast, the latter case would require the Responder itself to infer whether previous

responses had been effective. A simplified example of how it might do this would be to determine whether it had been required to issue repeated responses in relation to the same detected incident. If this was the case, then it could potentially infer that (a) the initial response actions were not effective against that type of incident, and (b) the last response action issued might form a better starting point on future occasions (i.e. upgrading and downgrading the perceived effectiveness of the responses when used in that context).

Having obtained such feedback, it would be desirable for the system to automatically incorporate it into a refined version of the Response Policy. This, however, would be a non-trivial undertaking, and it is anticipated that a full implementation of the system would need to incorporate machine-learning mechanisms to facilitate a fully automated process. An alternative would be to collate the feedback, and present it to the system administrator for later consideration when performing a manual overhaul of the Response Policy.

4 A Prototype Responder System

As an initial step towards the development of the Responder, a prototype system has been implemented that demonstrates the main response features of IMS, including the ability to make decisions based on the information from IDS alerts and other contextual factors.

The first element of the prototype is a console used to simulate intrusion conditions. In the absence of a full Detection Engine, or indeed genuine incidents, this is necessary to enable incident conditions to be configured before generating an alert to trigger the Responder's involvement. The parameters that can be adjusted from the console interface include the ones that are meant to be provided by the Detection Engine in the alert message, and are illustrated in Figure 2. The Responder can form a decision by monitoring (or determining) an additional set of contextual parameters, and then using these in conjunction with the ones included in the alert message.

The second component of the prototype is the Responder itself, which is responsible for receiving the alerts and making response decisions according to the given context. The Responder largely bases its decision upon the Response Policy, which can be accessed from the Responder module, by selecting the Response Policy Manager tool. A user-friendly interface is provided for the review of Policy rules, which are represented via a hierarchical tree, where the incidents are at the highest level and the response actions lie at the lowest levels. At the most basic level, there will be a one to one correspondence between a type of incident and an associated type of response. However, a more likely situation is that the desired response(s) to an incident will vary, depending upon other contextual factors, and the Policy Manager allows these alternative paths to be specified via intermediate branches in the tree. Between them, these intermediate branches comprise the conditions, under which specific response actions are initiated for particular incidents.

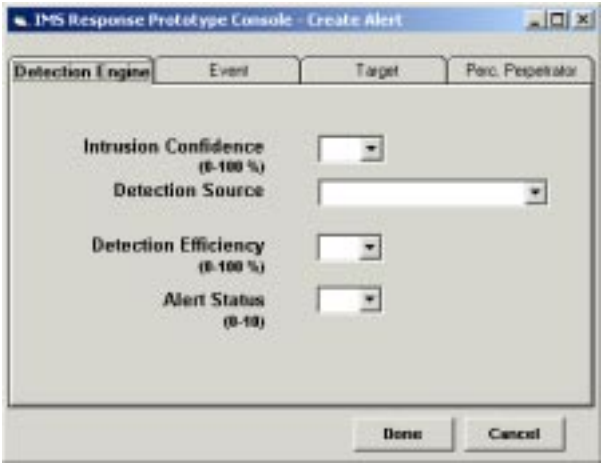


Fig. 2. Prototype Console Interface

The IMS Response Policy Manager is illustrated in Figure 3, with an example of response rules that could be specified in relation to an 'authentication failure' incident. In this case, had there been an alarm from the Detection Engine describing the successful login of a suspected masquerador, the Responder would check for the most recent update of related software to ensure that it is not vulnerable, and initiate keystroke analysis and facial recognition (if available) to authenticate the user in a non-intrusive manner. Of course, the conditions for the latter to happen would not be just the occurrence of the incident. Only the addition of the alarm to a log file would happen in that case. For the previously mentioned responses to be issued, the intrusion confidence would need to be low (hence the responder would need to collect more information about the incident), the overall threat and the importance of the target would need to be at low levels as well, not justifying the issue of more severe responses. Also, the account involved would need to be not privileged, with login time outside the normal pattern, in order to issue non-intrusive authentication.

Had there been a privileged account logged in at an abnormal time, then the urgency to collect more information about the incident would be greater and thus more intrusive countermeasures could be allowed. More authentication challenges like continuous keystroke analysis [17], the use of cognitive questions [18], and fingerprint recognition could also be used. Other methods that could be utilised include session logging (for further future reference or forensic purposes), alerting the user himself/herself about the occurrence of this suspicious behaviour (aiming to provoke a reaction from him/her and possible discourage him/her from any further unauthorised activity). Finally another option would be the redirection to a decoy system, in order to protect the integrity of the original target. Although this option would be more suited in the case of a server being compromised, it could still be an option for very sensitive environments, where a maximum level of security is required and minimum levels of risk are allowed. In any case, Figure 3 depicts an example of a security policy, which may or may not be optimal.

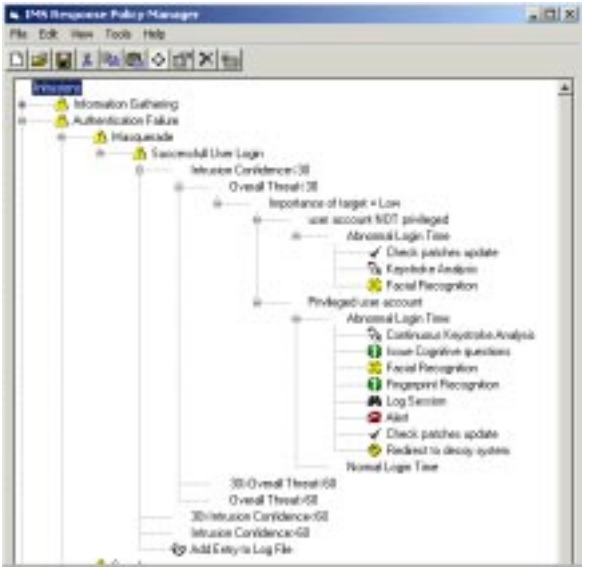
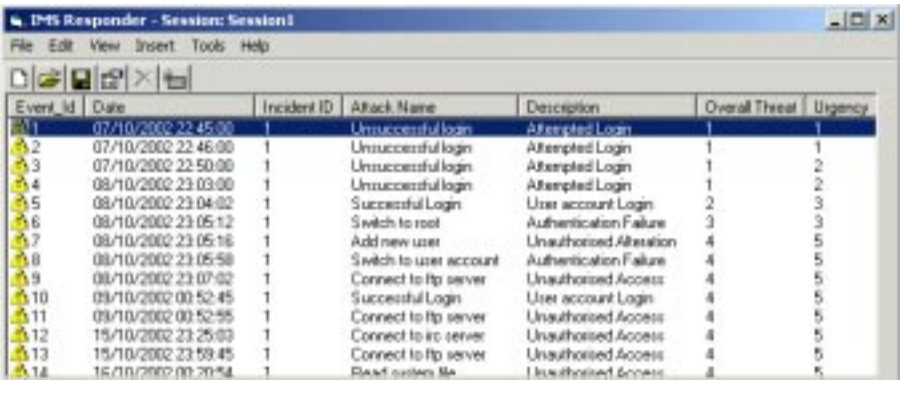


Fig. 3. IMS Response Policy Manager

Having determined the Response Policy, the Responder can make decisions about the alerts it receives. During normal operation, the Responder logs the details of responses that have been issued so that they can be tracked and reviewed by a system administrator. This is achieved via the Alert Manager interface (see Figure 4), which contains a list of suspected incidents, allowing them to be selected and reveal the response action(s) initiated for them. Each alert contains information about the incident itself, and the reasoning for the associated response decision. When viewing the alerts, it is also possible for the administrator to review the response decision that was made by the system, and provide feedback about the effectiveness of the actions selected. A full implementation of the Responder would use this feedback as the basis for automatic refinement of the response policy over time.



Event_Id	Date	Incident ID	Attack Name	Description	Overall Threat	Urgency
1	07/10/2002 22:45:00	1	Unsuccessful login	Attempted Login	1	1
2	07/10/2002 22:46:00	1	Unsuccessful login	Attempted Login	1	1
3	07/10/2002 22:50:00	1	Unsuccessful login	Attempted Login	1	2
4	08/10/2002 23:03:00	1	Unsuccessful login	Attempted Login	1	2
5	08/10/2002 23:04:02	1	Successful Login	User account Login	2	3
6	08/10/2002 23:05:12	1	Switch to root	Authentication Failure	3	3
7	08/10/2002 23:05:16	1	Add new user	Unauthorised Alteration	4	5
8	08/10/2002 23:05:58	1	Switch to user account	Authentication Failure	4	5
9	08/10/2002 23:07:02	1	Connect to ftp server	Unauthorised Access	4	5
10	09/10/2002 00:52:45	1	Successful Login	User account Login	4	5
11	09/10/2002 00:52:55	1	Connect to ftp server	Unauthorised Access	4	5
12	15/10/2002 23:25:03	1	Connect to irc server	Unauthorised Access	4	5
13	15/10/2002 23:59:45	1	Connect to ftp server	Unauthorised Access	4	5
14	16/10/2002 00:20:54	1	Read system file	Unauthorised Access	4	5

Fig. 4. IMS Responder: Alert Manager

5 Conclusions and Future Work

This paper has presented the requirements for enhanced intrusion response and the operational characteristics of an automated response architecture that enables flexible, escalating response strategies. The prototype system developed provides a proof-of-concept, and demonstrates the process of creating and managing a flexible response policy, as well as allowing intrusion scenarios to be simulated in order to test the response actions that would be initiated. Although the IMS approach as a whole would not necessarily be suited to all computing environments it is considered that the automated response concept could still be more generally applicable.

Future work could usefully include the integration of machine learning algorithms into the Responder implementation, in order to enable it to learn from the effectiveness (or otherwise) of previous response decisions and automatically refine the response policy accordingly. Based on the feedback from experience, the ability to learn and to assess its decision-making capability, the Responder could eventually attain a sufficient level of confidence to operate autonomously.

Acknowledgments. The research presented in this paper has been supported by funding from the State Scholarships Foundation (SSF) of Greece.

References

1. CERT Coordination Center: Security of the Internet, Vol. 15, The Froehlich/Kent Encyclopedia of Telecommunications, Marcel Dekker, New York (1997) 231–255
2. Richardson, R.: 2003 CSI/FBI Computer Crime and Security Survey (2003) http://www.gocsi.com/
3. Power, R.: 2002 CSI/FBI Computer Crime and Security Survey, Vol. VIII, No. 1, Computer Security Issues and Trends (2002) 10–11, 20–21
4. Denning, D.E.: An Intrusion-Detection Model**,** Vol. SE-13, No. 2, IEEE Transactions on Software Engineering (1987) 222–232
5. Allen, J., Christie, A., et al.: State of the Practice of Intrusion Detection Technologies, Technical Report CMU/SEI-99-TR-028, Carnegie Mellon University (2000) http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html
6. Mukherjee, B., Heberlein, L.T.; Levitt, K.N.: Network Intrusion Detection, *IEEE Networks 8*, no.3 (1994) 26–41
7. Schneier, B.: Secrets and Lies: Digital Security in a Networked World, John Wiley & Sons (2000)
8. Amoroso, E.: Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response, Second Printing, Intrusion.Net Books, New Jersey (1999)
9. Bace, R., and Mell, P.: NIST Special Publication on Intrusion Detection Systems, National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/drafts/idsdraft.pdf (2001)

10. Newman, D., Snyder, J., and Thayer, R.: Crying Wolf: False Alarms hide attacks, Network World Fusion Magazine, http://www.nwfusion.com/techinsider/2002/0624security1.html/ (2002)
11. Cohen, F.B.: Simulating Cyber Attacks, Defences, and Consequences, The Infosec Technical Baseline studies, http://all.net/journal/ntb/simulate/simulate.html (1999)
12. Lee, S.Y.J.: Methods of response to IT system intrusions, MSc thesis, University of Plymouth, Plymouth (2001)
13. Cheung, S., and Levitt, K.N.: Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection, Proceedings of the New Security Paradigms Workshop, Langdale,Cumbria UK (1997) http://riss.keris.or.kr:8080/pubs/contents/proceedings/commsec/283699/
14. Furnell, S.M., and Dowland, P.S.: A conceptual architecture for real-time intrusion monitoring, Vol. 8, No. 2, Information Management & Computer Security (2000) 65-74
15. Papadaki, M., Furnell, S.M., Lines, B.M., and Reynolds, P.L.: A Response-Oriented Taxonomy of IT System Intrusions, Proceedings of Euromedia 2002, Modena, Italy (2002) 87–95
16. Papadaki, M., Furnell, S.M., Lee, S.J, Lines, B.M., and Reynolds, P.L.: Enhancing response in intrusion detection systems, Vol. 2, No. 1, Journal of Information Warfare (2002) 90–102
17. Dowland, P., Furnell, S., and Papadaki, M.: Keystroke Analysis as a Method of Advanced User Authentication and Response, Proceedings of IFIP/SEC 2002 - 17th International Conference on Information Security, Cairo, Egypt (2002) 215–226
18. Irakleous, I., Furnell, S., Dowland, P., and Papadaki, M.: An experimental comparison of secret-based user authentication technologies, Vol. 10, No. 3, Journal of Information Management & Computer Security (2002) 100–108