Asian Journal of Information Technology 10 (5): 192-200, 2011
ISSN: 1682-3915

Intrusion Response Systems: An Overview

[1]Gbolahan Idowu, [1]Oluwatoyin Enikuomehin and [2]Suzan Olasanoye
[1]Lagos State University, Lagos, Nigeria
[2]Syke Bank Nigeria, Victoria Island, Lagos, Nigeria

Abstract: Intrusion detection and response to attack became a serious concern to researchers and network administrators as network attacks increases by the day. Several propositions and attempts are being developed to detect and respond to these attacks. This study presents an overview of some proposed intrusion response systems. Intrusion response systems were presented based on their methods of decisional analysis of intrusions detected by intrusion detection systems. Also, their methods of response reporting and limitations of each method were discussed.

Key words: Detection, response, threats, security, misuse, snort, access control, generic authorization

INTRODUCTION

Intrusion detection refers to a variety of techniques for detecting attacks in the form of malicious and unauthorized activity while intrusion response is the counter-measure evasive and or and ensure safety of the corrective actions to thwart attacks computing environment. Ignoring security threats can have example, the 2003 CSI/FBI Computer reported that participants in the survey the theft of proprietary information and denial of service attacks. Recently Intrusion Detection Systems (IDS) have been used in monitoring attempts to break security which serious consequences. For crime and security survey lost about $135 million from provides important information for timely and automated countermeasures.

As attacks of computer systems are becoming increasingly numerous and sophisticated, there is a growing need for intrusion detection and response systems to dynamically adapt to better detect and respond to attacks. Annual reports from the Computer Emergency Response Team (CERT) indicate a significant increase in the number of computer security incidents each year. Six incidents were reported in 1988 and over 8,200 were reported in 1999.

These days there are frequent attacks on system resources which often have grave consequences. Recently there was an intrusive attack on the mails of Sarah Palm the republican vice-presidential candidate of the United States election 2008 which had dire consequences. Intrusion detection and response is a proactive OCCSS that requires constant attention of system administrators. In order to remain secure, the network environment must continually be monitored for new attacks and prompt responses must be elicited. Though most intrusions are done over a network, all of these are directed towards a system and its resources.

If an administrator is left to attend to intrusions manually, it would he ineffective because the administrator may not be able to attend to all intrusions at once if multiple intrusion alarms are triggered. Also he may lack the required skill to categorize the intrusions according to the level of damage on the resources (priority) and as a result may attend to intrusions which are less important first and the more important last (misplaced priority).

Intrusion response system is a broad topic which covers a larger scope than is being discussed in this project. The scope of this project is to develop an some response system that responds to any form of intrusion alarm from network Intrusion Detection System (IDS).

The development of an Intrusion response system requires the discussion of a number of different domains including Intrusion detection systems, Intrusion response systems and security taxonomies. Intrusion Response Systems (IRS) are dependent on Intrusion Detection Systems (IDS) in two respects:

- IDS detect the intrusions that IRS must respond to
- IDS are imperfect which requires IRS to adapt the response based on its confidence on the detection capabilities of the IDS

There has been some previous research in IRS as all IDS contain some intrusion response components ranging

Corresponding Author: Gbolahan Idowu, Lagos State University, Lagos, Nigeria

from report generation to automatic defense of the system. Unfortunately intrusion response has rarely been discussed by itself. Instead, most research has focused on the detection of intrusions with intrusion response being left as the responsibility of the system administrator. As a result, the intrusion response mechanisms within these systems arc limited and ineffective.

INTRUSION DETECTION SYSTEMS

Intrusion Detection Systems (IDS) seek to monitor the behavior of users, networks or computer systems and help detect misuse of the systems through identification of anomalous behavior also at times alerting the management to take appropriate corrective action (like a burglar alarm). IDS are the last line of defense against computer attacks behind firewalls, secure architecture design, secure program design, carefully configured network services and penetration audits. In spite of the availability of a large variety of intrusion prevention techniques, the intrusion problem still remains challenging as there is no fool-proof way of reading the attacker's mind and the attackers are still successful in finding system loopholes in order to compromise the system resources. Most computer attacks are made possible due to poorly configured services or bugs in the software.

Host-based IDS versus network based IDS: Some of earliest intrusion detections were performed manually by system administrators who examined the audit logs of user and system events recorded by the computer hosts. Logged events might indicate that activities like a large number of failed login attempts, VIP (File transfer protocol) transfer of sensitive flies or failed file access were potential events for intrusive activity. Later, the manual task was replaced by Host-based IDS that could automatically detect potential attacks by scanning audit logs for signs of any suspicious activity. More recently due to tremendous growth in the field of computer networks, network-based IDS have gained popularity among researchers and even in commercial tools. Network-based IDS typically monitor the network data for intrusive activity and can be placed inside a firewall or outside it or at the system boundary. For example, network-based IDS can detect network probing attacks which map out the network topology of a site by searching for the PINGS (Packet Interface Network Groupers) to the network services across the complete site (Stakhanova *et al*., 2007).

The most difficult choice in the design of a secure system is to decide the location of the IDS in the network. On one hand, directly inspecting the state of the monitored system provides better visibility. Visibility makes detection more effective by increasing the range of analyzable events, decreasing the risk in having an incorrect view of the system and reducing the chances of an unmonitored attack. On the other hand increasing the visibility of the target system usually leads to weaker isolation between the IDS and the attacker thus, increasing the risk of a direct attack on the lOS itself. This leads to a choice between Host-based IDS that has an excellent view of what is happening in the host but is more prone to attack and Network-based IDS that has a poor view of the host's software but offers high attack resistance to it.

Misuse detection vs anomaly detection: Intrusion detection methods are broadly classified into two categories: Misuse detection and anomaly detection. Misuse detection methods also known as signature-based detection, use information about a known security policy known vulnerabilities and known attacks on the systems they monitor. This approach compares network activity or system audited data to a database of known attack signatures or other misuse indicators whereas pattern match produces alarms of various sorts. A lot of work is being done by researchers to find intelligent ways to map the dynamically changing attack patterns to already known attacks. Examples of such signature-based IDS arc Snort (Roesch, 1999) and NetSTAT (Vigna and Kemmerer, 1998). Snort is a popular cross-platform; lightweight network intrusion detection tool configured using a public database of attack signatures. It can be deployed to monitor small TCP/IP networks and detect a wide variety of suspicious network traffic as well as outright attacks. It can provide administrators with enough data to make informed decisions on the proper course of action in the face of suspicious activity. Snort can also be deployed rapidly to fill potential holes in a network's security coverage such as when a new attack emerges and commercial security vendors arc slow to release new attack recognition signatures. NetSTAT is a state based intrusion detection tool aimed for real-time network intrusion detection.

Anomaly detection methods also called behavior based intrusion detection, use information about repetitive and usual behavior on systems and attempt to detect intrusions by noting significant departures from normal behavior. The most significant advantage of anomaly detection techniques is the ability to detect novel attacks against systems. This is possible because anomaly detection techniques do not scan for specific patterns but instead compare current activities against models of past behavior. The biggest disadvantage of

anomaly detection approaches is the high rate of false alarms as compared to misuse detection techniques. Because any significant deviation from the previously learned behavior can be flagged as intrusion, it is highly likely that any non-intrusive behavior that falls outside the normal range will also be flagged as intrusive, resulting in a false positive. Another limitation of the anomaly detection approach is that the training data should be free from any intrusive behavior because if an attack occurs during the training period then this intrusive behavior will become a part of the normal baseline. The most desirable feature of ideal IDS is its ability to think one step ahead of the attacker i.e., its potential to identify novel attacks. Therefore, in spite of its drawbacks, anomaly detection approaches are still a better bet for detecting any future, unknown and novel attacks against computer systems. The researchers leave the discussion on IDS here as the research is mainly focused on what happens after an intrusion has been detected. For the system, the researchers arc using a signature-based network IDS, NetSTAT.

State-based IDS: State-based (Porras, 1993; Postel, 1981) is an approach of representing computer penetrations and can be applied to the development of a real-time intrusion detection tool. The approach, referred to as state transition analysis, views a penetration as a sequence of state changes that lead a computer system from an initial secure state to a target compromised state. State transitions are defined in terms of critical actions and assertions that describe the pre and post-action states of the system. A state transition diagram which is the graphical representation of state transition analysis, identifies precisely the requirements and compromise of a penetration and lists only those critical events that must occur for the successful completion of the penetration.

According to Ilgun *et al.* (1995), the State Transition Analysis Tool (STAT) is an advanced rule-based expert system that analyzes the audit trails of multi-user computer systems in search of impending security violations. STAT represents state transition diagrams within its rule-base and uses them to seek out those state transitions within the target system that correspond to known penetration scenarios. Unlike comparable analysis tools that pattern match sequences of audit records to the expected audit trails of known penetrations, STAT rules focus on the effects that the individual steps of a penetration have of the state of the computer system.

The resulting rule-base is not only more intuitive to read and update than current penetration rule-bases but also provides greater functionality to detect impending compromises.

STAT was initially developed for host-based IDS but later extended to describe network attacks (Vigna and Kemmerer, 1998). In network-based state transition analysis the state includes the currently active connections (for connection oriented services), the state of interactions (for connectionless services) and the values of the network tables (e.g., routing tables, DNS mappings and ARP caches, etc.). For instance, both an open connection and a mounted file system are part of the state of the network. A pending DNS request that has not yet been answered is also part of the state such as the mapping between an IP address and the machine name.

Suspicion state versus conviction state: A suspicion state is a state in the state transition analysis at which there is an initial hint of an intrusive activity. This state might or might not lead to a final compromised state but it acts as a trigger to start monitoring the system. An example of the initial state is the receipt of port scan packets from a different subnet. In the case of denial-of-service attacks (such as the ping of death) this initial state is the first attack packet. A conviction state is the final compromised state at which the IDS is convinced of the attack. In case of DoS attacks this could be the receipt of a threshold number of packets in a given period of time.

In case of SYN flood attack, the attacker sends packets with SYN flag set as a request to open a new connection to the server. The victim responds to the request then waits for confirmation that never arrives. As a result, the victim's connection table fills up waiting for replies and all new connections are ignored. In case of this attack, the suspicious state could be receipt of a threshold number of requests with no following ACKs and the conviction state is when the connection table is filled up.

Automated response mechanisms: The ideal IDS is the one that responds to intruder action, to stop his/her activity before he/she can do any damage to the system or access sensitive information. Most systems require intensive management which may be a cumbersome task for a human administrator. After the attack has been detected, the responsibility to respond to an attack is left to the system administrator. This process might be slow and prone to human errors. Hence, there is a need to automate the response and management mechanism. An effective defense system should be able to detect the attacks and communicate with other entities in the system to take an effective collaborative action in an automated fashion (Carver and Pooch, 2000). The basic lacking factor is the ability to detect attacks in real time and for that the ability to recognize unknown attacks is necessary.

According to NIST, there are 20-30 new attacks that are posted on the internet every month and keeping track of all new attacks and updating the signatures can be quite expensive and inefficient as 1 week link can leave the system vulnerable.

INTRUSION RESPONSE SYSTEMS

The autonomous Intrusion Response Systems (IRSs) are designed to respond at runtime to the attack in progress. The goals of an IRS may be a combination of the following: to contain the effect of the current attack if the underlying model is a multi-stage attack, to recover the affected services and to take longer term actions of reconfiguration of the system to make future attacks of a similar kind less likely to succeed. There are several challenges in the design of an IRS:

- First, the attacks through automated scripts are fast moving through the different services in the system
- Second, the nature of the distributed applications enables the spread of the attack, since under normal behavior the services have interactions among them and a compromised service can infect another
- Third, the owner of the distributed system does not have knowledge of or access to the internals of the different services. For example, the source code may not be available or even if available; the expertise to understand the internals may not be available
- Hence, an IRS should ideally work at the interfaces rather than in the internals
- Fourth, it may not be possible to deploy detectors at each service for performance reasons (say, the performance overhead imposed by the packet matching at a network-based detector is excessive for a host) or deployment conditions (say, no host-based detector is available for the particular platform)

Additionally, the detectors if installed may be faulty and produce false alarms or missed alarms. The IRS therefore has to suppress inaccurate detections and extrapolate from the available detectors to determine the appropriate services at which to take the response action.

Finally, the distributed systems are often complex enough that the universe of attacks possible against such systems is not enumerable and therefore, the IRS has to work with possibly unanticipated attacks.

The current IRSs meet only a subset of the above challenges and none that we are aware of addresses all of them. The general principles followed in the development of the IRS naturally classify them into four categories.

Static decision making: This class of IRS provides a static mapping of the alert from the detector to the response that is to be deployed. The IRS includes basically a look-up table where the administrator has anticipated all alerts possible in the system and an expert indicated responses to take for each. In some cases, the response site is the same as the site from which the alarm was flagged as with the responses often bundled with anti-virus products (disallow access to the file that was detected to be infected) or network-based IDS (terminate a network connection which matched a signature for anomalous behavior). Snort-inline and Norton Antivirus fall in this category.

Dynamic decision making: This class of IRS reasons about an ongoing attack based on the observed alerts and determines an appropriate response to take. The first step in the reasoning process is to determine which services in the system are likely affected, taking I to account the characteristics of the detector, the network topology, etc. The actual choice of the response is then taken dependent on a host of factors such as the amount of evidence about the attack, the severity of the response, etc. The third step is to determine the effectiveness of the deployed response to decide if further responses are required for the current attack or to modify the measure of effectiveness of the deployed response to guide future choices. Not all IRSs in this class include all the three steps. A wide variety is discernible in this class based on the sophistication of the algorithms (Stakhanova *et al*., 2007).

Intrusion tolerance through diverse replicas: This class of IRS implicitly provides the response to an attack by masking the effect of the response and allowing the computer system to continue uninterrupted operation. The basic approach is to employ a diverse set of replicas to implement any given service. The fault model is the replicas are unlikely to share the same vulnerabilities and therefore not all will be compromised by any given attack (Wu *et al*., 2007). A voting process on the outputs or the state of the replicas can mask the compromised replicas provided less than half are compromised. An advantage of this approach is the system can continue operation without a disruption. This approach is reminiscent of active replication in the fault tolerance field.

Responses to specific kinds of attacks: This class of IRS is customized to respond to specific kinds of attacks, most commonly, Distributed Denial of Service (DDoS) attacks. The approach is to trace back as close to the source of the

attack as possible and then limit the amount of resources available to the potentially adversarial network flows. A characteristic of this approach is cooperation is required from entities outside the computer system being protected for an accurate trace back. In this study, the researchers will describe the primary IRSs that have been reported in the literature and label each in one of these four categories.

Static decision making systems: The characteristic that defines this class of IRSs is that they respond to attacks defined exactly, prior to deployment and using responses that arc enumerated and completely configured. They are in general simple to understand and deploy and work well for a large class of systems that have determinism in the kinds of workload and where the attack modes are enumerable a priori. However, they are not very effective for dynamic systems with changing workloads, new kinds of services installed and new vulnerabilities introduced due to hardware or software changes.

Generic Authorization and Access control API (GAA-API): The Generic Authorization and Access Control-Application Programming Interface (GAA-API) (Ryutov *et al*., 2003) is a signature-based intrusion detection and response system that provides a dynamic authorization mechanism at the application layer of a computer system. The basic idea is integrate access control policy with intrusion detection and some countermeasure according to policy such as generating audit records.

GAA-API, developed by the Information Sciences Institute supports access control policies and conditions defined by a BNF-syntax language. It is a generic tool that has been integrated with many applications including Apache, SSH, SOCKS5 and FreeS/WAN (IPSec VPN), running on Linux and Sun Solaris platforms. It is designed as a generic interface based on standard C language APIs SC) it can be easily ported to other platforms and applications.

Snort inline: Snort inline is a mode of operation for Snort, the popular open-source intrusion detection system. Originally developed as an independent, modified version of Snort, it was rated in version 2.3.0 RCI of the Snort project to provide intrusion prevention capabilities. It requires the Net filters/IPtables software developed by the same project. Snort Inline provides detection at the application layer to the IPtables firewall so it can respond dynamically to real time attacks that take advantage of vulnerabilities at the application level (Roesch, 1999).

Dynamic decision making systems: Dynamic decision making based IRS involves the process of reasoning about an ongoing attack based on the observed alerts and determining an appropriate response to take. There have been various designs and architectures proposed for this kind of dynamic decision making based IRS systems. However, the core issue underlying all these systems is how the decision making should be achieved. Many factors can contribute to and complicate the decision making process.

For instance, a response can come with a certain cost such as the computation resource required for executing the response and the negative impact on the system after the execution of this response. Also a response can fail with some probability. So, at the highest level of abstraction for each applicable response option, an IRS has to consider both the outcome from deploying the specific response and not deploying it and makes a decision between these two choices based on some metric. From this point, the researchers can see three potential research issues regarding dynamic decision making based IRSs:

- One is modeling the effect of an attack on the system and this is directly related to the outcome from a decision on not using any response
- The second issue is modeling the effect of the responses and this is related to the outcome from a decision on using responses
- Finally, there's the issue of how to decide the set of responses for deployment for a given attack, considering that responses are deployed on different hosts or services in a distributed environment and that they are not all independent. Now we provide the details of some representative dynamic IRSs

ADEPTS

ADEPTS Foo *et al*. (2005) makes use of the characteristics of a distributed appl ication in guiding its response choices. It considers the interaction effects among the multiple services both to accurately identify patterns of the intrusions relevant to the response process (e.g., cascading failures due to service interactions) and to identify the effectiveness of the deployed response mechanism.

In designing an IRS, a possible approach is to consider different attacks and provide customized sequence of response actions for each step in an attack. A second approach, subtly yet significantly different is to consider the constituent services in the system and the different levels of degradation of each individual k service due to a successful attack. For easier understanding, one

may visualize a malicious adversary who is trying to impact the constituent services (the sub-goals) with the overall goal of either degrading some system functionality (e.g., no new orders may be placed to the e-store) or violating some system guarantee (e.g., credit card records of the c-store customers will be made public). In ADEPTS, the researchers take the latter approach. This is motivated by the fact that the set of services and their service levels are finite and reasonably well understood while the possible universe of attack sequences is potentially unbounded.

They focus on the manifestations of the different attacks as they pertain to the services rather than the attack sequence itself. This leads them to use a representation called an Intrusion Graph (I-GRAPH) where the nodes represent sub-goals for the intrusion and the edges represent pre-conditions/post conditions between the goals. Thus, an edge may be OR/AND/Quorum indicating any all or a subset of the goals of the nodes at the head of the edge need to be achieved before the goal at the tail can be achieved.

In ADEPTS, the response choice is determined by a combination of three factors-static information about the response such as how disruptive the response is to normal users; dynamic information, essentially history of how effective the response has been for a specific class of intrusion and out-of-band parameters of the response such as expert system knowledge of an effective response for a specific intrusion or policy determined response when a specific manifestation occurs. Importantly and distinct from other work, ADEPTS points out the need for the IRS to provide its service in the face of unanticipated attacks (Foo *et al*., 2005). Thus, it neither assumes that the I-GRAPH is complete nor that there is a detector to flag whenever an I-GRAPH node is achieved. However, it assumes that the intrusion will ultimately have a manifested goal which is detectable ADEPTS also considers the imperfections of the detection system that inputs alerts to it. The detectors would have both type I and type 11 errors i.e., false alarms and missed alarms. If false alarms are not handled this can cause the IRS to take unnecessary responses, potentially degrading the system functionality below that of an unsecured system. If missed alarms (or delayed alarms) are not compensated for, the system functionality may be severely degraded despite the IRS. ADEPTS can co-exist with off the shelf detectors and estimates the likelihood that an alarm from the detection system is false or there is a missing alarm. The algorithm is based on following the pattern of nodes being achieved in the I-GRAPH with the intuition that a lower level sub-goal is achieved with the intention of achieving a higher level sub-goal.

The design of ADEPTS is realized in an implementation which provides intrusion response service to a distributed c-commerce system. The e-commerce system mimics an online book store system and two auxiliary systems for the warehouse and the bank. Real attack scenarios are injected into the system with each scenario being realized through a sequence of steps. The sequence may be nonlinear and have control flow such as trying out a different step if one fails. The responses of ADEPTS are deployed for different runs of the attack scenarios with different speeds of propagation which brings out the latency of the response action and its adaptive nature (Foo *et al*., 2005).

The survivability of the system is shown to improve over a baseline system with a larger number of runs leading to greater improvement.

Intrusion tolerance through diverse replicas: The use of diverse replicas in IRS borrows ideas from the field of natural fault tolerance and from observations of biological systems. By introducing artificial diversity, a common phenomenon in biological systems, an attack specific to a vulnerability in a system cannot affect another system that lacks that vulnerability. Coupled with redundancy, the effect of an attack ban be masked, allowing the system to provide continued service in the presence of disruptions. The basic approach is to employ a diverse set of replicas for a given service such that they provide the same high level functionality with respect to other services but their internal designs and implementations differ. The fault masking techniques used are similar to methods in natural fault tolerance such as voting and agreement protocols. The use of diverse replicas is attractive because provable theoretical improvements to the survivability or security of the system can be obtained, compared to other techniques that are more suitably classified as heuristics.

Evaluation techniques from the mature field of natural fault tolerance arc more readily adapted to this class of IRS's. A common assumption is to assume at most a fraction of the servers in a network may fail. This assumption is strengthened through the use of active and periodic recovery. Another common assumption is that failures in the system are independent which motivates the use of diversity. Extending this argument to vulnerabilities, the assumption states that vulnerabilities do not occur across different operating systems and applications.

THOUGHTS ON EVOLUTION OF IRS TECHNOLOGY

The researchers anticipate that for IRSs to be widely deployed they will have to evolve in several directions over coming years. These include the following:

Ability to withstand unpredictable attack scenarios: It is inconceivable that all attack scenarios would be programmed in the IRS. The IRS should therefore be able to extrapolate strategies available in its knowledge base and take responses to hitherto unseen attacks. This will be an important requirement since polymorphic worms, viruses and other forms of attacks are rampant in today's security landscape. In this matter, there is a delicate balancing game between learning from the past and being agile to respond to future attacks. It is possible to build up large knowledge bases and do exact matches with them to choose appropriate response from the history. However, this may affect the ability of the system to respond quickly. Also in taking lessons from the past, the IRS should take into account the fact that the impact of the attack may be different even though the attack steps may be the same. Thus a more drastic or quicker response may be called for.

Dynamic responses with changing network configurations: The IRS will have to deal with topology and configuration changes in the distributed system. It may take inputs from change notification software systems, such as Tripwire and modify its response strategies accordingly (Ragsdale *et al*., 2000). In any medium to large sized distributed system, there are multiple administrators responsible for maintaining the system. The tools are often not standardized or uniform across different administrators. Thus, modifying the tools to send notification to the IRS seems daunting. A more feasible approach appears to be software to observe the resultant changes and notify the IRS. A change in the configuration may render some responses unnecessary (such as a critical service being made accessible from only inside the corporate network) or some responses more critical (such as a service being made web accessible).

Interaction with other components of the security framework: The response strategy decided on by the IRS is predicated on confidence placed on other components of the security framework such as IDS, change notification software and firewalls, etc. The confidence placed on these components should not be pre-defined constant values. The confidence should change as new software is installed, rules update or configurations change. This also indicates why a probabilistic framework for the IRS seems the promising avenue, rather than deterministic response decisions. On another point, the IRS may depend on various basic functionalities in the system such as firewalls or access control system to deploy the computed responses (Toth and Kruegel, 2002).

Separation of policy and mechanism: It is important for the IRS to provide mechanisms for determining the appropriate response based on security policy settings. As far as practicable the two aspects should be clearly delineated. This will enable a system administrator to set the policy which can be at various levels of abstraction such as a paranoid versus laissez faire policy at the system-wide level, to policy levels for individual services. In the absence of this, an IRS will not have buy-in for production systems.

User interface design: Visualizing the different effects of an attack and its responses in a distributed environment is inherently challenging. The speed of the processes (attacks as well as responses) makes this a particularly daunting task. However, for critical functions, all the stake holders (system administrators to CIOs of the organization) will like to have a human digestible form of the information available to them. This should include online tools which lets them visualize the network while an attack or its responses are being deployed as well as offline tools which will aid in forensics action.

OVERVIEW OF REPORTING AND RESPONSE

Many attempts have been made to improve on intrusion detection and response system. These efforts have only made ways into the effectiveness of intrusion detection and response system. These efforts have only made ways into the effectiveness of intrusion detection and have not eliminated the requirement for an automated response system.

A classification of response functions in other response systems is given in Carver and Pooch (2000). The response function in detection systems can be categorized as a notification system, manual response system or automatic response system. According to the researchers, most systems today are notification systems.

In Carver (2001), an Adaptive Agent-Based Intrusion Response System (AAIRS) was proposed. This was the first response system implementing a notion of learning. In his research, the interface relies on human action to update its intrusion detection systems confidence metric.

An adaptive intrusion detection system is described in Ragsdale *et al*. (2000). This system is used together with AAIRS to provide both adaptive detection and response. The response system is relatively advanced. It keeps track of previous alarms and classifies attacks on the basis of whether they are a continuation of an existing incident or whether it is a new attack. Alarms from different intrusion detection systems in the system have

different confidence metrics according to previous detection results. The confidence in a suspected incident and nature of the incident affects the course of action taken.

A study by Toth and Kruegel (2002) proposed yet another promising model for automating intrusion response. The researchers suggested a way of approaching the problem of response to network intrusions by constructing dependency trees that model configuration of the network.

Another significant research in the area of intrusion response system includes a thorough consideration of some intrusion detection and response cost modeling aspect by Lee *et al.* (2002). They provided a good introduction to modeling costs of an intrusion detection and responses.

Comprehensive and thorough surveys of 56 intrusion detection systems were carried out in Carver (2001). From his findings, there were no deducted solutions for intrusion response. There were however, some responses implemented in a variety of intrusion detection systems. Most of the intrusion detection systems were notification and manual response systems which were not preferable solutions. There were however, some automatic response systems as well but these were rather insignificant. There is this possibility of having a delay between an alert and human reaction when manual system responds to attack.

Reporting and response: Human beings are incapable of dealing with the speed and amount of information which computers generate. They are also prone to error, misinterpretation and it is very difficult to accurately predict their capabilities. Computers on the other hand precisely execute what they have been instructed to, deterministic in that the execution of the same sequence of instructions will always produce the same result and their processing capabilities can be estimated in advance.

Reporting: Reporting is a key phase of intrusion detection as it is the main point of interaction with computers and humans. The immense amount of data gathered, analyzed, sorted and classified by the intrusion detection system must now be presented to the human administrator. The ability of the administrator to react to an intrusion and take appropriate action will depend greatly on his ability to process information reported by the intrusion detection systems.

Limitation

Human factors: The major limitation of the reporting system is the human administrator. If thousands and thousands of events scroll by, it is impossible for a human to understand them and take appropriate action. The impact of over whelming amount of information is delivered to the automated response system with proper analysis and classification. If not handled properly, humans have the tendency to eventually switch off features that are too loud (Dobrucki, 2003).

Response

Manual response: Many experts choose to do away with automated response and concentrate efforts on optimizing manual response instead. The researchers can view manual response as having somewhat different properties and goals than automated one.

The researchers cannot expect the reaction time to be near real-time and we must have an operator who is trained in incident response. To the benefit though, the response which is tailored to the specific incident can be followed by in-depth analysis and recovery and lead to problem eradication.

Whereas the automated response is often aimed at stopping the intrusion in progress, manual response strives to give a balanced methodological approach to solving the intrusion problem. The researchers can see the response as a four step process:

- Containment
- Eradication
- Recovery
- Lesson learned

A requirement for successful execution of these four steps is well-trained incident response personnel aided by proper documentation. The combination of personnel and response time is the main cost factor in manual response. The growing number of site security officer required to minimize time quickly becomes the dominant factor in intrusion detection system maintenance. The procedure for manual response is typically as follows: locate the problems area and systems which have been compromised (Containment). Patch the security Hole which allows the intrusion to proceed, verify that other systems do not have this problem (Eradication). Finally, recover the systems which took part and document the incident (Lesson learned).

LIMITATION

The main limitation of manual response is inevitable delay between an alert and human reaction. An automated

exploit script the tasks in about 30 sec It is simply impossible to achieve this kind of response time from humans regardless of available resources.

CONCLUSION

In this study, the researchers have presented the motivation for designing Intrusion Response Systems (IRSs) for distributed systems. The researchers lay out the design challenges in designing and implementing IRSs. Then, the researchers present existing research in the field, classified into four classes. The first category of IRSs called static decision making provides a static mapping of the alert from the detector to the response that is to be deployed. The second class called dynamic decision making reasons about an ongoing attack based on the observed alerts and determines an appropriate response to take. The third class called intrusion tolerance through diverse replicas provides masking of security failures through the use of diverse replicas concurrently for performing security critical functions. The fourth class includes IRSs meant to target specific kinds of attacks with the focus being Denial of Service (DOS) attacks. Finally, the researchers presented five key areas in which IRSs need to evolve for a widespread adoption. In conclusion, the researchers find that the design and development of IRS has been gaining in research attention and the researchers expect that they will become main stream in the computer security landscape in the near future.

REFERENCES

Carver, C.A. and U.W. Pooch, 2000. An intrusion response taxonomy and its role in automatic intrusion response. Proceedings of the IEEE Systems, Man and Cybernetics Information Assurance and Security Workshop, June 6-7, New York, USA., pp: 129-135.

Carver, C.A., 2001. Adaptive agent-based intrusion response. Ph.D. Thesis, Department of Computer Science, Texas A and M University, College Station, Texas.

Dobrucki, M., 2003. Priority in the deployment of network intrusiondetection systems. Master's Thesis, Telecommunications Software and Multimedia Laboratory, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland.

Foo, B., Y.S. Wu, Y.C. Mao, S. Bagchi and E.H. Spafford, 2005. ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. Proceedings of the International Conference on Dependable Systems and Networks, June 28-July 1, Yokohama, Japan, pp: 508-517.

Ilgun, K., R.A. Kemmerer and P.A. Porras, 1995. State transition analysis: A rule-based intrusion detection system. IEEE Trans. Software Eng., 21: 181-199.

Lee, W., W. Fan, M. Miller, S. Stolfo and E. Zadok, 2002. Toward cost-sensitive modeling for intrusion detection and response. J. Comput. Secur., 10: 5-22.

Porras, P., 1993. STAT-a state transition analysis tool for intrusion detection. Technical Report, University of California at Santa Barbara, Santa Barbara, CA., USA.

Postel, J., 1981. RFC 792: Internet control message protocol. IETF RFC Publication. http://www.rfc-ref.org/RFC-TEXTS/792/index.html.

Ragsdale, D.J., C.A. Carver, J.W. Humphries and U.W. Pooch, 2000. Adaptation techniques for intrusion detection and response systems. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 8-11, Nashville, TN., USA., pp: 2344-2349.

Roesch, M., 1999. Snort-lightweight intrusion detection for networks. Proceedings of the 13th LISA Conference on System Administration, Nov. 7-12, Seattle, Washington, pp: 229-238.

Ryutov, T., C. Neuman, K. Dongho and Z. Li, 2003. Integrated access control and intrusion detection for web servers. Proceedings of the 23rd International Conference on Distributed Computing Systems, May 19-22, California, USA., pp: 394-401.

Stakhanova, N., S. Basu and J. Wong, 2007. A taxonomy of intrusion response systems. Int. J. Inform. Comput. Secur., 1: 169-184.

Toth, T. and C. Kruegel, 2002. Evaluating the impact of automated intrusion response mechanisms. Proceedings of the 18th Annual Computer Security Applications Conference, (ACSAC'02), IEEE Computer Society Washington, DC., USA., pp: 301-310.

Vigna, G. and R. Kemmerer, 1998. NetSTAT: A network-based intrusion detection approach. Proceedings of the 14th Annual Computer Security Application Conference, Dec. 7-11, Phoenix, AZ., USA., pp: 25-34.

Wu, Y.S., B. Foo, Y.C. Mao, S. Bagchi and E. Spafford, 2007. Automated adaptive intrusion containment in systems of interacting services. Comput. Networks, 51: 1334-1360.