Interested in learning more about security?

SANS Institute InfoSec Reading Room

Algorithm-based Approaches to Intrusion Detection and Response

Computer and network intrusion detection systems were first implemented in the early 90's. Since that time a field of research in intrusion detection has focused on the ability of the IDS to detect intrusion attempts, using statistical and algorithm based approaches, and discern between what is merely anomalous (unknown to the system) and not a risk, and what is potentially harmful to the system and should be prevented. Tools available on the market have incorporated these statistical and algorithm-based models in the ...

Algorithm-based approaches to intrusion detection and response

Alexis Cort
March 16, 2004
GSEC Practical Requirements (v.1.4b) (August 2002)
Option 1

ABSTRACT

Computer and network intrusions have been with us since the introduction of the computer, but intrusion detection systems are still somewhat new to the market (first implementations started in the early 90's). They have grown from systems capable only of passive logging of anomalous activity into copious activity logs to active defense systems, capable of not only detecting intrusion attempts but also of responding to them autonomously, without human input. Given that attacks are short lived[1], and it takes a human system administrator some time just to analyze the data and comprehend that an attack is taking place before beginning to react to it, these systems must be able to function autonomously to a significant degree in order to serve their purpose.

A field of research in intrusion detection has focused on the ability of the IDS to detect intrusion attempts, using statistical and algorithm based approaches, and discern between what is merely anomalous (unknown to the system) and not a risk, and what is potentially harmful to the system and should be prevented. Tools available on the market have incorporated these statistical and algorithm-based models in the design of their detection modules, but have largely left response up to the operator, giving the user the ability to script responses. Since precious time is used in detecting an attack, these systems will need to adopt some autonomous response capability, using not only risk and response categorization but also a response escalation algorithm, similar to biological and immune response systems. Most of these systems also spend time learning about the systems they are protecting and establishing a baseline, before they and are able to function as intended. Since much of this data is available from system vendors, greater cooperation among vendors will obviate much of the need for this learning process and improve intrusion detection systems.

[1] An example is cited in this paper of an attack which took place inside of 16 seconds.

TABLE OF CONTENTS

Background

Anderson introduced the concept of intrusion detection back in 1980[2], and defined an intrusion attempt or a threat to be the potential possibility of a deliberate unauthorized attempt to access information, manipulate information, or render a system unreliable or unusable.

Types of attacks and intrusions

Graham defines three categories of attacks and intrusions[3]:

Reconnaissance attacks include port scans, ping sweeps, email recons, DNS zone transfers, and public web server indexing to find holes in the network through which they can gain access.

Exploits are bugs or hidden features in applications, servers, and operating systems which allow unauthorized access to the system.

Denial of service (DoS) attacks are typically indiscriminate attempts by an attacker to crash systems or overload network connections, memory buffers, and CPU registers with the intent of denying access to your system by everyone else.

While these categories all seem to address attacks from the outside, we must not forget that attacks and intrusions can come from both outside and inside the organization. Hackers, industrial spies, and other people may try to compromise the security of the system from the outside, but disgruntled employees or individuals who have gained physical access to the organization's systems may similarly compromise security.

Intrusion detection systems

An intrusion detection system (IDS) is a system which monitors traffic to detect intrusions and attacks, and in some cases, initiate a series of actions to respond to the intrusion or attack in an attempt to protect systems and data an prevent future attacks.

Graham further breaks IDS down into the following categories[4]:

Network intrusion detection systems (NIDS) monitor network traffic and can protect either a single system or multiple systems and devices on a network.

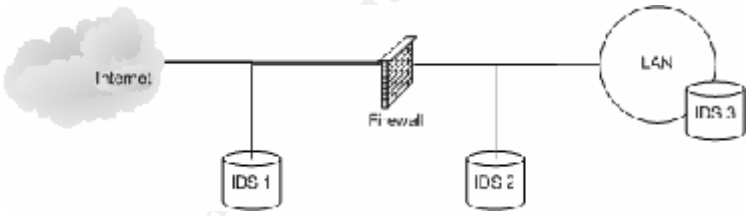
[2] J.P Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James P Anderson Co., Fort Washington, Pennsylvania, April 1980.
[3] Robert Graham, FAQ: Network Intrusion Detection Systems http://www.robertgraham.com/pubs/network-intrusion-detection.html#2.1, accessed on March 10, 2004
[4] Ibid.

System integrity verifiers (SIV) monitor system files to see which have been changed or what files have been added to the system, and send alerts or can take other actions based on security policy. This approach was also introduced in antivirus software packages, which employ heuristic algorithms to monitor applications which write to system files; this approach is however only useful if you know which applications change what system files. While this enabled the antivirus vendors to claim that they detect 'known and unknown' viruses, it left the user with multiple prompts during an application installation: for example, pondering if they should allow the installation to proceed or not. For this reason, many of us have left the heuristics option off in our antivirus program installations, instead relying on a timely update from the vendors and common sense practices such as never opening an email attachment from a suspicious source.

Log file monitors (LFM) monitor log files, looking for patterns matching historical attacks or suspicious activity. In this respect, they monitor the system in the same way an IDS does.

Implementation

In general, IDS can be implemented in the following locations, as shown in this simplified[5] diagram, adapted from Graham[6]:



IDS 1 can detect attacks against the firewall
IDS 2 detects traffic which has penetrated the firewall
IDS 3 represents implementation of one or more IDS at various nodes throughout the network, and can detect attacks by insiders

IDS should be implemented in conjunction with, rather than in replacement of, a firewall, notification systems, and other intrusion countermeasures. Above all, the organization should have a well defined security policy before implementing IDS, which will help configure the IDS detection parameters and determine an appropriate response. For example, the organization should have a policy clearly defining what constitutes an authorized user, what access rules are, and what the consequences for unauthorized access are.

[5] The diagram does not take into account other devices which would affect the IDS installation, like routers, switches, and DMZ (which is beyond the scope of this document); it's intended only to demonstrate the ability of IDS to monitor both local and WAN traffic.
[6] Robert Graham, FAQ: Network Intrusion Detection Systems http://www.robertgraham.com/pubs/network-intrusion-detection.html#2.1, accessed on March 10, 2004

IDS design models

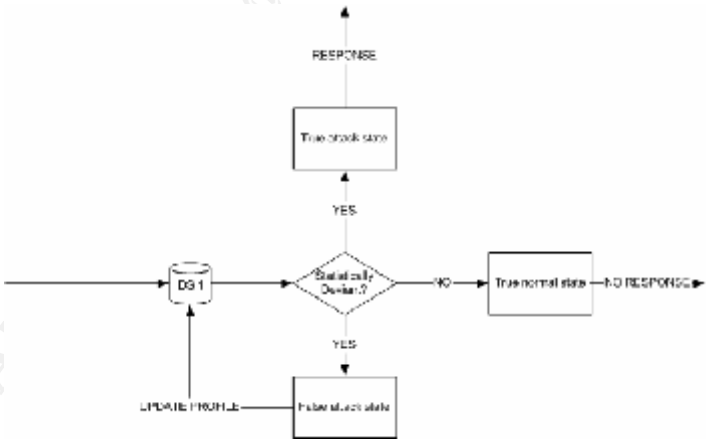
For purposes of discussion in this paper, I will use the broad categories defined in the COAST (Computer Operations, Audit, and Security Technology) project at Purdue University, since renamed CERIAS (Center for Education and Research in Information Assurance and Security). Purdue University has a well established program in intrusion detection research. Price has categorized intrusion detection systems, based on their detection models, into the following[7]:

- *Misuse detection model* detects intrusions by looking for activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities.
- *Anomaly detection model* detects intrusions by looking for activity that is different from a user's or system's normal behavior.

The following diagrams have been adapted from Sundaram[8] to illustrate the differences in how the models function, and demonstrate the need for these systems to learn network, system and use behavior in order to reduce the likelihood of false positives:

Anomaly Detection Model

A typical anomaly detection model will analyze data, compare to a known profile, run statistical analysis to determine if any deviation is significant, and flag the event(s) as a True Attack State, False Attack State, or Normal State. If it finds a false positive, the profile must be updated to reflect the results[9].



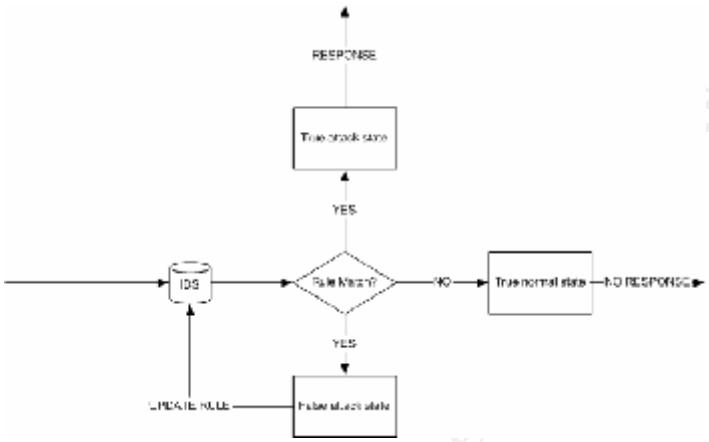
[7] Mark Crosbie, Katherine Price, David A. Curry, Intrusion Detection Systems, COAST Resources. http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/ids.html, accessed on March 8, 2004

[8] Aurobindo Sundaram, An Introduction to Intrusion Detection, http://www.acm.org/crossroads/xrds2-4/intrus.html, accessed on March 5, 2004

[9] The model disregards the possibility of a False Negative, such that the system does not catch an intrusion; the assumption made here is that the IDS ships in an 'all activity is potentially suspicious' state, that the thresholds are set sufficiently low (to allow the triggering of false positives during the learning process), and that the intruder has not compromised passwords or other legitimate means of gaining access.

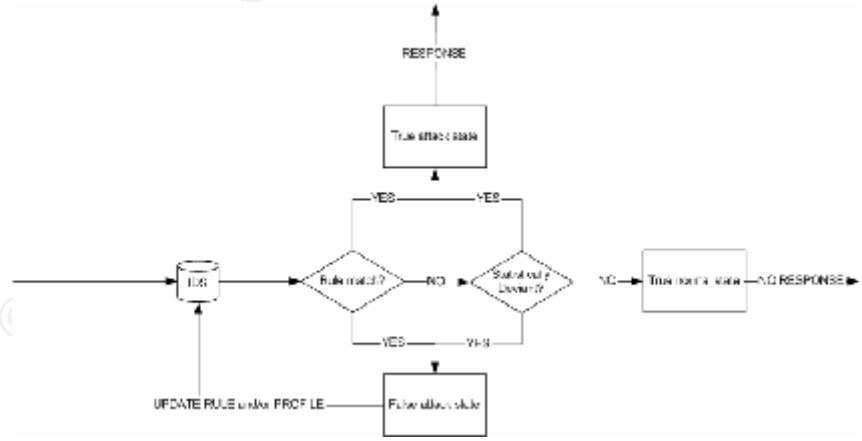
Misuse Detection Model

A typical misuse detection model will similarly analyze data, compare against existing access rules and existing attack profiles or signatures in its database, and flag the event(s) accordingly[10].



Hybrid Anomaly/Misuse Detection Model

Finally, a hybrid anomaly/misuse detection model can analyze data and determine not only if the events are suspicious and if it corresponds to a known attack profile, but also if either the sequence of events or the attack profile are statistically significant. For example, in the case of an attack which has previously not been mapped, the characteristics can be compared against existing profiles in order to determine if it's similar to a known attack profile.



[10] The same potential to flag false positives exists in this model. This model also disregards the possibility of a False Negative, with the assumption that the rule sets which ship with the products and any modifications made to them during installation are appropriate for the organization.

Other characterizations of intrusion systems exist, such as host/multi-host, network, but for the purposes of this discussion, I will draw the following observations:

1. Both anomaly detection and misuse detection will depend on a learning process or the establishment of a baseline: in the anomaly detection case, logging 'normal' activity, in the misuse detection model, logging or cataloging intrusion signatures and system vulnerabilities. In both cases, therefore, there will be a period of time when the intrusion detection system does not have the baseline data it needs to compare the new activity against, and will therefore be unable to provide information about the attack or respond to the attack.
2. The misuse detection model can further be characterized as backward-looking or reactive, relying on historical data of past attack signatures and system vulnerabilities; the anomaly detection model can be characterized as forward-looking (albeit still reactive[11]), detecting heretofore unknown system behavior.

Farshchi mentions detection methods modeled on the immune system, control-loop measurements, data mining, statistical analysis, and analysis of signatures[12]. Price et al.[13] point to other areas of research in neural networks and machine learning classification techniques, but further in depth discussion of these approaches are outside the scope of this document.

Detection

One of the main tasks of IDS is to help distinguish between malevolent and innocent intrusions. All IDS systems are prone to identify false positive and false negatives[14].

Misuse detection based, or signature-based, systems include NFR[15], Dragon[16], Snort[17], and Cisco Secure[18], and are still the prevalent form of IDS available today. Farshchi points out one of the drawbacks of signature based systems being the multitude of new signatures and exploits coming out weekly and the inability of an already burdened system administrator to keep up with them, in terms of updating the baseline of signature-based IDS to reflect the most up to date signatures[19]. He points out the benefits of a statistical based approach to intrusion detection, in supplementing a

[11] It is reactive because it is still reacting to an event ex post facto.
[12] Jamil Farshchi, Statistical based approach to Intrusion Detection, SANS Intrusion Detection FAQ. http://www.sans.org/resources/idfaq/statistic_ids.php, last accessed on March 8, 2004
[13] Mark Crosbie, Katherine Price, David A. Curry, Intrusion Detection Systems, COAST Resources. http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/ids.html, accessed on March 8, 2004
[14] Dan Hawrylkiw, Network Intrusion and use of automated responses http://www.sans.org/resources/idfaq/auto_res.php, last accessed on March 7, 2004
[15] NFR, http://www.nfr.com/solutions/system.php, last accessed on March 9, 2004
[16] Dragon, http://www.portcullis-security.com/Products/Intrusion-Detection-System/Intrusion-Detection-System.htm, last accessed on March 9, 2004
[17] Snort, http://www.snort.org/, last accessed on March 9, 2004
[18] Cisco Secure, http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/, last accessed on March 9, 2004
[19] Jamil Farshchi, Statistical based approach to Intrusion Detection, SANS Intrusion Detection FAQ. http://www.sans.org/resources/idfaq/statistic_ids.php, last accessed on March 8, 2004

signature based IDS with an anomaly based system which uses statistical analysis to determine a relevance threshold of a new, unknown event or sequence of events and alerts the system administrator that the threshold has been reached or surpassed. He uses a tool called Spade[20] which has the ability to do statistical analysis and threshold calculation. While the tool is still likely to report false positives if the threshold value is set too low, or miss real positives if it's set too high, it does give the user the ability to start baselining normal system activity, with the assumption that given enough data in correlating system behavior to anomaly threshold breaches, the user will be able to arrive at a meaningful threshold value with time. This is a drawback which will be inherent to most algorithm based systems, as they will require time to learn what constitutes normal behavior of the system, and will depend on a historical baseline. While this means that with each new attack or system reconfiguration the learning process will start anew, one could argue that the learning process will get shorter, eventually approaching some asymptotic minimum learning time. His argument is not that statistical based algorithms are superior to signature based systems, but rather that they complement them, and augment the capability of the IDS as a whole.

Anomaly detection systems include EMERALD[21] (Event Monitoring Enabling Responses to Anomalous Live Disturbances), GrIDS[22] (Graph Based Intrusion Detection System), ASAX[23] (Advanced Security audit trail Analysis on UniX), and AAFID2[24] (Autonomous Agents For Intrusion Detection 2). All of these are still mostly research projects, albeit with promise of commercial viability. The anomaly detection model was originally proposed by Denning[25] in 1987, who proposed a series of metrics like CPU load, number of network connections per minute, and number of processes per user to measure what would constitute 'normal' activity for a user. Abnormally high CPU load combined with other anomalous metrics, for example, may indicate a system intrusion in process. Almost by default, anomaly detection systems must use statistical algorithms to analyze data and detect intrusions.

Hawrylkiw argues that no matter what the detection mechanism, IDS which depends on a human response will not be able to respond quickly enough to provide adequate protection[26]. An attack described at the San Diego SANS Network Security 2001 took place inside of 16 seconds, so the fact that the security analyst was not in the office at the time did not matter – he or she would not have been able to react quickly enough to the attack even if they were directly observing the attack. Hawrylkiw points out that autonomous IDS, one that detects the intrusion and automatically sets in motion

[20] Spade, http://ww.silicondefense.com/software/spice/, last accessed on March 5, 2004
[21] EMERALD, http://www.sdl.sri.com/programs/intrusion/, last accessed on March 9, 2004
[22] GrIDS, http://seclab.cs.ucdavis.edu/arpa/grids/welcome.html, last accessed on March 9, 2004
[23] ASAX, http://www.info.fundp.ac.be/~cri/DOCS/asax.html, last accessed on March 9, 2004
[24] AAFID2, http://www.cerias.purdue.edu/about/history/coast/projects/aafid-announce-0999.php, last accessed on March 9, 2004
[25] Dorothy E. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, 13(2):222-232, February 1987.
[26] Dan Hawrylkiw, Network Intrusion and use of automated responses http://www.sans.org/resources/idfaq/auto_res.php, last accessed on March 7, 2004

responses based on a set of predefined parameters, is the only way to provide adequate protection against intrusion attempts.

Wu, Foo, Matheny, Olsen, and Bagchi combine data mining, control loops, statistical analysis, threat and response classification, and an attack graph representation in their design of an autonomous detection system at Purdue[27]. Their ADEPTS (Adaptive Intrusion Tolerant System) system attempts to categorize and subsequently identify the goal of an attack, by taking intrusion alerts from different 'nodes' distributed throughout the system and correlating the alerts. Each node is classified as a child node or a parent node, and represents an intermediate or ultimate goal of an attack; the system calculates a Compromised Confidence Index (CCI) based on the inputs, categorizes each node as a strong, weak, very weak, or non candidate, and then calculates a Response Index to help determine the appropriate response to the attack. The Response Index is based on factors such as confidence (from the CCI), candidate category, likelihood that the response will disrupt users and business processes, and effectiveness of the response from past experience. ADEPTS ultimately provides feedback to aid in choosing a response that is both appropriate to the level of intrusion (for example, the perceived goal of the attack) and most able to contain the attack. An ADEPTS prototype installed in a model e-commerce system was able to provide response feedback in under 4 seconds with as many as 25 concurrent alerts.

Response

It is this final phase of autonomous IDS (automating a response to the intrusion) that is most interesting and fraught with risk. Slow down or shut down the network, and you disrupt business processes and upset users. Counter the intrusion with an attack of your own, and you risk reprisals from innocent parties whose systems were compromised and then used for the attack. Limit yourself to notification, and you risk depending on an overburdened security staff to choose the right response, while confidential company data has already been pilfered.

Hawrylkiw describes the following types of responses to an attack[28]:

Session sniping disrupts the communication between the attacker and target, typically by forging RST packets. The source IP, ports, and sequence numbers of the packets must correspond to the traffic that triggered the event; even then, the countermeasure may fail because the attacker's system may handle the forged packets differently than the victim's systems, possibly ignoring the forged packets.

[27] Yu-Sung Wu, Bingrui Foo, Blake Matheny, Tyler Olsen, Saurabh Bagchi, ADEPTS: Adaptive Intrusion Containment and Response using Attack Graphs in an E-Commerce Environment http://www.ece.purdue.edu/~sbagchi/Research/Papers/adepts_dsn04_submit.pdf, last accessed on March 9, 2004

[28] Dan Hawrylkiw, Network Intrusion and use of automated responses http://www.sans.org/resources/idfaq/auto_res.php, last accessed on March 7, 2004

ICMP Messaging gets around the problem of injecting RST packets into a stream using the UDP protocol. Since UDP is a stateless protocol and cannot use RST packets to close connections, an ICMP message can be sent to the attacker instead, informing them that the destination is unreachable. This method also has a low probability of success, as it depends on the attacker system paying attention to the message, which may get ignored or dropped.

Shunning is denial of access to the attacking host, typically by reconfiguring the firewall to block the offending IP address. Frameworks provided by products such as Checkpoint's OPSEC[29] are already integrated with firewalls and allow for this reconfiguration to take place. This is a very effective method, but it can lead to the attacker escalating the attack by forging packets either from other sources or to other ports, and end up blocking access to legitimate users.

Non-blocking responses intervene with, but do not disrupt, the traffic. This has the benefit of not alerting the attacker (possibly triggering a different or escalated attack as discussed previously), while still performing functions which protect the system and users. Post-attack cleanup includes scanning for files placed on the system during the attack and deleting them. Redirection either routes the attacking hosts through additional security controls, or changes destinations, for example by remapping ports. If the IDS installation includes a honeypot, the attacker can be redirected there which has the additional benefit of logging the attacker's activity for further actions to be taken.

Extended notification is an easily scripted response to an attack. While most IDS notify the appropriate personnel inside the organization in response to an attack, extended notification can be used to notify other entities outside the organization, including the attacker's ISP.

Counterattack is also discussed in Hawrylkiw's paper, with some reservations. The problem lies in the inability of IDS to distinguish a real attacker from a compromised innocent system or a spoofed source, possibly resulting in an attack on another innocent victim. The other victim may then decide to take action against you. The possibility of tipping off the attacker and tainting evidence which can be used in a subsequent investigation is also mentioned, even though it can be presumed that the IDS has already logged the attack before initiating this type of response.

Intrusion and response effectiveness

In spite of the options offered by IDS tools in the marketplace, both intrusion and response mechanisms employed by these tools continue to have shortcomings. Detection algorithms can have false positives and false negatives, depending on both the sensitivity and the arbitrary setting of the threshold value. Response systems can adversely affect company operations by disrupting connections, can tip off the attacker, and in most cases depend on human response to respond to an attack which for all

[29] http://www.opsec.com/, last accessed on March 9, 2004

practical purposes has already taken place. Both misuse detection and anomaly detection systems spend time learning about the system they are protecting, and must establish a baseline before reaching a nominal level of effectiveness.

Anomaly detection systems in particular depend on a knowledgeable system administrator or someone who can analyze the data coming out of the system and reclassify false positives, or adjust the multiple parameters in the system. As such, these systems are anything but Plug and Play; with additional development in both algorithms and interface, they will become less resource intensive, at least for the end user. IDS systems can be quite resource intensive in terms of computations, but are still expected to respond in seconds under heavy traffic load. The best solution seems to be multiple nodal IDS installations, with each system capable of receiving data from the other nodes and correlating data to reach a conclusion.

These systems will continue to improve, but they have already achieved a moderate degree of success. The IDS may not be able to stop an intruder, but can be very effective in throwing enough obstacles in the intruder's path, making the cost of the attack to the intruder high enough to all but the most determined. Even a determined intruder will spend time defeating all of the countermeasures, while his or her actions are being logged; and after all, unless the intruder is specifically interested in the target, there are enough other, easier targets out there where the risk of detection and possible consequences are lower.

Escalation

Further automation of the response calculation algorithm can help mitigate some of these shortcomings in IDS systems. Response escalation is a way to initiate a response and selectively escalate to higher level responses, those with a higher probability of deterring an attack or stopping an intrusion in progress, but at the price of affecting company operations. This way the IDS can start responding 'before all the data comes in', while it analyzes data from other IDS nodes, calculates correlation coefficients, and plots its next move.

Taking the response categories described previously and assigning arbitrary effectiveness and business impact values to arrive at a response index, the algorithm could choose from the following table as an example:

Type of response	Effectiveness	Business Impact	Escalation Level	Example
Notification/alarms	1	1	1	Send notification to system administrator, set visual/audio alarm, start logging
Extended notification	1	1	1	Send notification email to offender's ISP informing them of the attack and possible consequences
Session sniping	2	3	2	Send RST packets to offending host to terminate TCP session
ICMP Messaging	2	2	2	Forge ICMP messages to offending host indicating target host is unavailable
Shunning	3	2	2	Reconfigure firewall to block offending IP address
Post-attack cleanup	4	2	3	Detect new/changed files and delete or quarantine them
Redirection	4	3	3	Reconfigure firewall to reassign ports
Counterattack	5	4	4	Initiate counterattack against offender's IP address

Using the data in this table, the IDS could for example start responding at Escalation Level 1, while it continues to gather data. As more data come in or correlation coefficients reach a certain threshold, and the overall confidence in the attack rises, it could then escalate further.

The response algorithm could also interact with the security policies, by for example reassigning or encrypting passwords to sensitive data directories – users would have a 'normal' and 'emergency' password to access.

Escalation would also provide the additional benefit of logging responses to the attack and establishing a response audit trail – establishing a pattern of appropriate response.

The above table shows a gradual escalation of intrusion response, matching the threat with appropriate response. Halme and Bauer propose mention an alternative in a policy of intrusion preemption, where the organization escalates early to provide stern warnings to users to discourage them or a reward system for those spotting unauthorized activity[30]. While some of the other techniques mentioned such as infiltrating hacker lists and spreading disinformation may end up causing the organization more problems than benefits, the organization can still use the set of

[30] Lawrence R. Halme, R. Kenneth Bauer, AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques, SANS Intrusion Detection FAQ. http://www.sans.org/resources/idfaq/aint.php, last accessed on March 9, 2004

options given in the table and decide whether it would be more beneficial to escalate early or slower.

Industry cooperation

One reason why you can safely turn off the heuristics option in antivirus software (other than that the output and prompts you get as a result are of limited use) is that the industry is fairly efficient at releasing virus signatures and software updates. The inability of the antivirus software to identify the hundreds of system file writes that may occur in a single day as either normal or anomalous could be resolved by the operating system and application software vendors providing this data to the makers of antivirus software; since the same problem exists in IDS tools, the same goes for these vendors. This could be accomplished without releasing the closely guarded source code, which should remain proprietary to these vendors; perhaps an API-like document released only to AV and IDS vendors could suffice, and greatly reduce the need for IDS tools to go through a learning phase each time a new OS update or application comes out.

Summary

Intrusion detection systems have grown from a concept to an important component in an organization's security infrastructure. When used in conjunction with firewalls and other access control devices, they can bolster an organization's ability to detect, prevent, and respond to unauthorized access and intrusion attacks.

The anomaly based IDS, using statistical analysis algorithms, threat and response classification, and response escalation hold the most promise for a system which does not need to rely on historical attack signatures in order to act in time to protect an organization from intrusion.

As intrusion detection systems mature, they must incorporate a certain degree of autonomous response into their detection and response algorithms to solve the problem of time lag due to reliance on human response. Vendors must either tap into existing research projects, or conduct their own research to increase the accuracy of detection, reduce the likelihood of false positives and negatives (by utilizing statistical analysis), and allow the IDS to interact with other devices such as the firewall to enable timely and appropriate response to attacks. The operating system and application software vendors can help improve IDS tools by providing data about how their applications interact with the operating system and vice versa.

References

1. J.P Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James P Anderson Co., Fort Washington, Pennsylvania, April 1980.
2. Dan Hawrylkiw, Network Intrusion and use of automated responses http://www.sans.org/resources/idfaq/auto_res.php, last accessed on March 9, 2004
3. Jamil Farshchi, Statistical based approach to Intrusion Detection, SANS Intrusion Detection FAQ. http://www.sans.org/resources/idfaq/statistic_ids.php, last accessed on March 9, 2004
4. Yu-Sung Wu, Bingrui Foo, Blake Matheny, Tyler Olsen, Saurabh Bagchi, ADEPTS: Adaptive Intrusion Containment and Response using Attack Graphs in an E-Commerce Environment
http://www.ece.purdue.edu/~sbagchi/Research/Papers/adepts_dsn04_submit.pdf, last accessed on March 9, 2004
5. Aurobindo Sundaram, An Introduction to Intrusion Detection,
http://www.acm.org/crossroads/xrds2-4/intrus.html, last accessed on March 9, 2004
6. Lawrence R. Halme, R. Kenneth Bauer, AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques, SANS Intrusion Detection FAQ.
http://www.sans.org/resources/idfaq/aint.php, last accessed on March 9, 2004
7. Robert Graham, FAQ: Network Intrusion Detection Systems
http://www.robertgraham.com/pubs/network-intrusion-detection.html#2.1, last accessed on March 9, 2004
8. Mark Crosbie, Katherine Price, David A. Curry, Intrusion Detection Systems, COAST Resources.
http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/ids.html, last accessed on March 9, 2004
9. Dorothy E. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, 13(2):222-232, February 1987.
10. CERIAS Intrusion Detection at Purdue University.
http://www.cerias.purdue.edu/about/history/coast_resources/intrusion_detection/, last accessed on March 9, 2004
11. CheckPoint's OPSEC, http://www.opsec.com/, last accessed on March 9, 2004
12. Spade, http://ww.silicondefense.com/software/spice/, last accessed on March 5, 2004
13. EMERALD, http://www.sdl.sri.com/programs/intrusion/, last accessed on March 9, 2004
14. GrIDS, http://seclab.cs.ucdavis.edu/arpa/grids/welcome.html, last accessed on March 9, 2004
15. ASAX, http://www.info.fundp.ac.be/~cri/DOCS/asax.html, last accessed on March 9, 2004
16. AAFID2, http://www.cerias.purdue.edu/about/history/coast/projects/aafid-announce-0999.php, last accessed on March 9, 2004
17. NFR, http://www.nfr.com/solutions/system.php, last accessed on March 9, 2004
18. Dragon, http://www.portcullis-security.com/Products/Intrusion-Detection-System/Intrusion-Detection-System.htm, last accessed on March 9, 2004

19. Snort, http://www.snort.org/, last accessed on March 9, 2004
20. Cisco Secure, http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/, last accessed on March 9, 2004

Last Updated: October 19th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
GridEx IV 2017	Online,	Nov 15, 2017 - Nov 16, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, GB	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Berlin 2017	OnlineDE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced