A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis

Ming-Yuh Huang, Thomas M. Wicks
Applied Research and Technology
The Boeing Company
Seattle, WA, U.S.A.
huang@bcstec.ca.boeing.com
thomas.m.wicks@boeing.com

Abstract

To appropriately address the problem of large-scale distributed intrusion assessment/detection, issues such as information exchange, work division and coordination amongst various Intrusion Detection Systems (IDS) must be addressed. An approach based on autonomous local IDS agents performing event processing coupled with cooperative global problem resolution is preferred. However, it is not clear how autonomous the local IDS agents should be and what constitutes the theme that drives multiple IDS to work together.

We believe that focusing on the intruder's intent (attack strategy) provides the theme that drives how various IDS components work together. Analysis on attack strategy also provides an opportunity to perform pro-active look ahead adaptive auditing. This paper presents a high-level conceptual architecture view for such an approach.

The Battleground Management Analogy

Today's large-scale distributed intrusion detection (ID) shares many common traits and challenges with the task of battleground management. Both endeavors face difficult challenges such as:

- widely distributed heterogeneous environment
- voluminous, noisy and volatile data
- incomplete information for decision making
- diverse variety of probes
- difficulty in communication, coordination, command-and-control
- trust between entities
- changing attack patterns

In Intrusion Detection Systems (IDS), *Misuse Detection* performs signature analysis by comparing on-going activities with patterns representing past intrusions in attempt to recognize similar attacks. *Anomaly Detection* works by discovering deviations from normal operation profiles to detect suspicious activities [Sundaram]. Both are effective techniques that are also commonly used in the battleground situation - e.g. investigating enemy’s force layout, monitoring transportation patterns and communication volume. The goal is to look for unusual patterns and/or signs of offensive gathering.

However, these techniques by themselves are more suitable toward local event monitoring and analysis. Globally coordinated dynamic defensive information warfare requires more.

What’s Missing?

What is largely missing in today’s distributed ID theater is the equivalence of strategic decision making and command-and-control that’s common in battleground management. This is the layer where the enemy's strategy is analyzed, own strategy formulated and command-and-control dynamically executed. A platoon in the front line knows where and what to look for only if there is an offensive/defensive strategy devised and orchestrated by the field marshal in the back. It is the strategy that leads to the actions on the battleground.

In addition, battleground command-and-control goes beyond simply exchanging fragmented data. It calls for orders that carry field marshal's intention as well as autonomous execution of orders, reporting back, and perhaps issuance of more orders - based on the collected data.

If multiple tank sightings have been reported, the field marshal is likely to issue an order looking for additional evidences - fuel supplies patterns, force movement, communication volume, etc., to support confirm or deny a possible flank attack. The formation of the suspicion is critical and that is based on the analysis of enemy's intent.

It is also imperative that the orders communicate the field marshal's intention as well as the specifics of what to look for. The ability to interpret an order's intention is also crucial. Understanding the intention behind the order allows the platoon to observe and report back not just the specifics, but all the data pertinent to a flank attack. In the real military scenario, this order is not likely to be in the form of:

> ***“Look for X, look for Y and look for Z.”***

Instead, it is more likely to be -

> ***“Several tanks were sighted at X locations at T time. It is suspected that enemy is mounting an offensive gathering for a flank attack aiming at location L. Observe and report all unusual movements and information along front line F”.***

Figure-1 shows how information flows in a complex and volatile environment - high level, intention driven, dynamic and interactive. The numbers on the figure show the order of how the information flows between the field marshal and the local platoons.

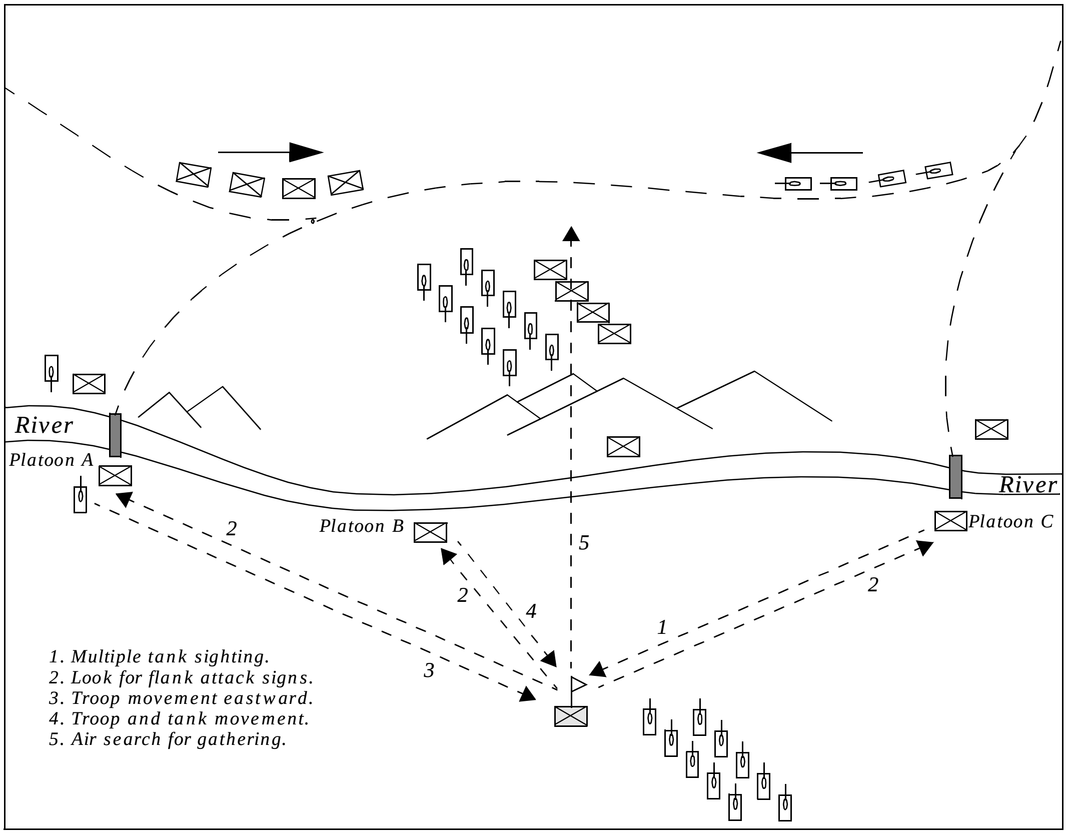
River
Platoon A
River
Platoon B
Platoon C
2
5
2
4
2
1
3
1. Multiple tank sighting.
2. Look for flank attack signs.
3. Troop movement eastward.
4. Troop and tank movement.
5. Air search for gathering.

Figure-1 Battleground intention analysis, command and control

A Centralized, One Way Event Processing Architecture

Today's centralized, one-way data collection and event processing ID architecture is a passive information processing paradigm. This architecture faces considerable limitations when trying to detect increasingly sophisticated attacks. The situation deteriorates as IDS start to deal with large and complex networks.

Large-scale heterogeneous networks generate a tremendous amount of real-time data in very diverse formats. Much of these data are not security related. Most are system management information. Only careful analysis can determine which are security related and which are not. It is a very noisy world. In the process of performing analysis and correlation, it is extremely difficult to accurately interpret the meaning behind these data out of the context.

For example, the significance of a “file-access-violation” cannot be determined without additional context. The file accessed could be a regular user file on a workstation or it could be a critical system file on a gateway machine or a sensitive data server. The severity depends very much on the context.

In addition, contextual information about machine configuration at the time of event is also important. The locations of the files, auditing parameters, setups for applications, users, groups and many other parameters fluctuate. This affects the auditing outcome. In fact, mis-configuration accounts for large percentage of the false-positives. NT, VMS and Unix all have different ways of setting up users and groups. NT allows for ACL (Access Control List) based on user defined “customized” group. As a result, NT File System's group-oriented “file-access-violation” carries a different meaning from UNIX's. To interpret this correctly, IDS actually needs to know the NT user-defined group.

In a centralized event-processing framework, by the time huge amount of data arrives at a centralized location, the contextual information needed to properly analyze the event has already been lost. That information existed only in the original environments. Worse even, the time latency may make it impossible to go back and collect additional data, from the original environment, to confirm or exonerate any suspicion. This one-way communication paradigm cannot support an advanced adaptive look-ahead auditing paradigm where the adaptive capability allows IDS to adjust the system's auditing behavior based on the evidence on hand.

The other almost insurmountable challenge is the difficulty involved in correlating voluminous distributed events. As a result of the lost context and the difficulty in translating/mapping various data formats, the significance of many events is lost. Without that, it is almost impossible to do a quality correlation. Pattern identification thus becomes quite difficult, considering the number of ways one can penetrate a system. All the attack approaches manifest into many forms of data. A centralized location is indeed a very high noise-to-signal ratio and voluminous environment. Detecting intrusion, from both *Misuse Detection* and *Anomaly Detection* perspectives, is extremely difficult.

The Basic Anatomy of an Intrusion

Just like a war, when a computer intrusion happens, the sequence of attack does not take place in a totally random order. Intruders come with a set of tools trying to achieve a specific goal. The selection of the hacking tools and the order of application depends heavily on the situation as well as the responses from the targeted system.

Typically there are multiple ways to invade a system. Nevertheless, it usually requires several actions/tools be applied in a particular logical order to launch a sequence of effective attacks to achieve a particular goal. It is this logical partial order that reveals the short and long-term goals of the invasion.

Figure-2 shows how a typical IP spoofing works. Host Intruder (I) wants to masquerade as A when talking to Victim (V). Host I will first send an initialization request announcing its intent to talk to V while pretending itself as A (1). In response to the request that presumably came from A, V will reply with a acknowledgment to start up a connection with A. The immediate problem here is that when A sees an acknowledgment to a request not generated by itself, it will send a reset (2). This will cause V to drop the connection with I (3).

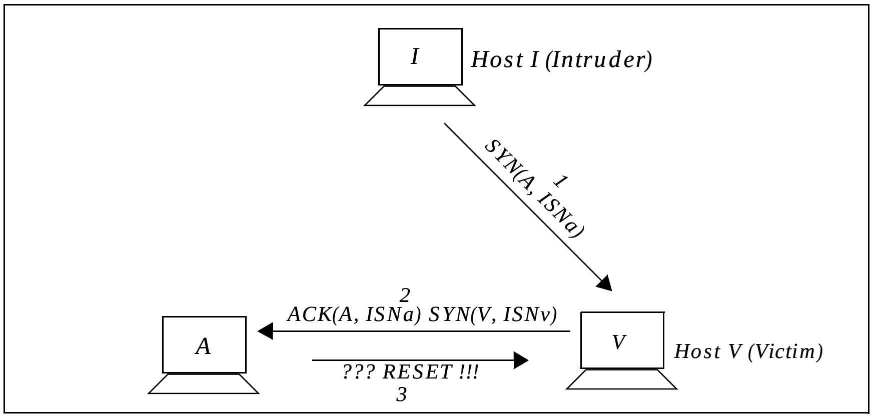

Figure-2 A typical IP spoofing.

So, one way to prevent A from resetting, I will first SYN-Flood. This will keep A so busy that V's acknowledgment (2) is simply dropped and will go unnoticed. As a result, A will not reset and I can spoof A (Figure-3).

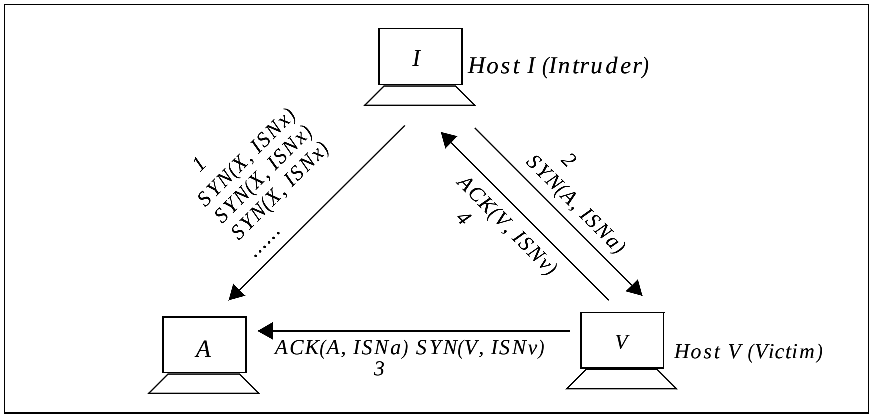

Figure-3 SYN-flood A before IP spoofing.

This simple example illustrates a logical sequence of mounting an attack with two steps, even though SYN-flood by itself can be used for denial-of-service attack.

To extend the scenario further, following is a sample list of logical attack sequences. Each sequence can be performed using different set of tools and that could happen on different systems. Therefore, many varieties of data are generated and even a larger number of permutations. Sequences can be pieced together if that makes sense.

1.Do SYN-flood to do IP spoofing, as in Figure-3.
2.Do IP spoofing to do session highjack, telnet highjack, web spoofing. (Better yet, use tools such as Hijack or TTYWatcher that will do both, with a GUI.)
3.Do Session highjacking to do obtain system data.

4.Obtain encrypted password file to do off-line password cracking (Crack, L0phCrack).
5.After accessing system, hide one's presence (RootKit).
6.After accessing system, spread out to other systems. (sniffing for passwords)
7.Install trojan horse (ifconfig) to hide PROMISC flag. (since sniffing will cause PROMISC flag to be on)
8.Modify file date/checksum (fix) to hide the trojan horse.
9.After accessing system, install backdoor. (RootKit)

Goals can, and often do, change as the intrusion progresses. It is rarely the case that one single tool or action can achieve the purpose of penetration. Attackers often need to collect information and probe around for known or unknown weaknesses. Observation, probing and data-collection are the likely initial steps. Upon any discovery, cautious but logical steps are taken to explore the environment. It is precisely the logic that formulates the attacks that could help in detecting them.

Strategizing Large Scale Distributed Intrusion Detection

To appropriately address the large-scale distributed intrusion assessment/ detection problem, a higher level of information exchange, work division and coordination amongst various IDS must be developed.

An approach leaning towards a higher degree of autonomous local problem resolution coupled with cooperative global problem solving is preferred. However, it is not clear how autonomous the local IDS agents should be and what constitutes the *theme* that drives multiple IDS to work together.

WE BELIEVE THAT FOCUSING ON INTRUDER'S INTENTION (ATTACK STRATEGY) IS THE THEME THAT DRIVES REMOTE IDS TO WORK TOGETHER. WE ALSO BELIEVE THAT ANALYSIS ON ATTACK STRATEGY PROVIDES AN OPPORTUNITY TO PERFORM PRO-ACTIVE LOOKAHEAD ADAPTIVE AUDITING.

A very early sign of intrusion - perhaps a suspicious event, provides a window of opportunity for discovery. But, this is also a prime opportunity for false-positives. Large number of false-positives is a considerable problem in the world IDS deployment. To reduce the false-positives, it makes sense to adjust the focus of attention based on the prediction of the intruder's intent, to zero in the possible attack directions instead of just generating an alarm.

Just like in a criminal case, criminal intent, as well as the evidence of how-to, must both be established. As a matter of fact, intention analysis often leads the way of criminal investigation and evidence is commonly used to establish hypothesis of the intention.

Strategy Analysis Based Architecture

There is a need for a large-scale distributed intrusion assessment and detection framework based on collaborative intrusion intention analysis performed by autonomous local IDS agents.

In addition to the security events, IDS agents communicate in terms of agendas that carry suspected intrusion intentions. Intrusion intentions are high-level platform independent attack strategies that can manifest into large permutation of low level system/network events.

By intention analysis, IDS agents can recognize attacks at the strategic level. At this level, attacks are characterized by a sequence of logically related but not necessarily complete intrusion steps - “sub-goals”. Each represents a state of accomplishment during the formation of an intrusion (the final goal). The task of intrusion detection thus becomes trying to recognize the sub-goal completion and trend development, in addition to the glaring violations.

A sub-goal usually manifests itself in many flavors and permutations of events coming from different platforms. To substantiate the formation of a sub-goal, autonomous local agents, which understand native event format, examine the local context and make the decision. Typically, it takes several events to recognize that a sub-goal is in formation. Under such circumstances, since local IDS agents can make the best judgement and filter out much of noise, high noise-to-signal ration data pose less of a problem.

Representation of Intrusion Intention

The representation format needs to be simple, generic and effective so that sharing is not a problem between various IDS implementations. It also makes sense that the representation can ride on standards in formation, such as USA DARPA's CIDF (Common Intrusion Detection Framework), ANSI/T4 or IETF (Internet Engineering Task Force).

A natural way to represent intrusion intention is the goal-tree. The root node represents the ultimate goal of intrusion. Lower level nodes represent alternatives or ordered sub-goals in achieving the upper node/goal. Leaves (the end nodes) are sub-goals that can be substantiated by event or events generated in different environments.

However, a typical goal-tree representation does not have the notion of temporal sequence or order embedded in it. Augmentation is needed. Figure-4 shows the three basic constructs of such an augmented goal-tree representation.

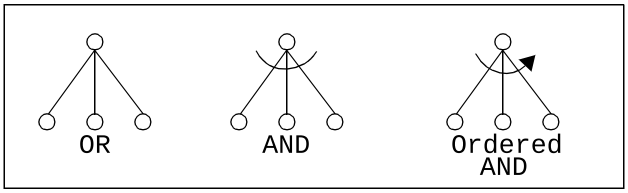

Figure-4 The OR, AND & Ordered-AND construct of an intrusion Intention goal-tree.

With this representation, as in the example in Figure-5, one horizontal thread through the nodes in a tree represents one possible attack scenario. Assuming that the nodes are filled as they are being confirmed, in many cases, the threads are incomplete and have hole(s) along the path. An incomplete thread represents a possibility of an intrusion in development. Past data was either over-looked, not audited or too weak to support it. In figure-5, node X and node Y represent windows of opportunity to re-examine the archived data to either confirm or exonerate the suspicion. Figure-6 shows a possible representation of the previously described flooding/spoofing/sniffing sequences.

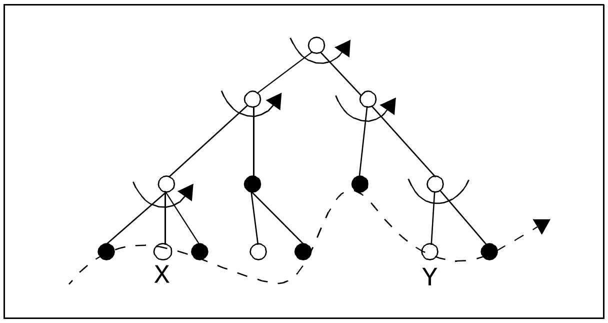

Figure-5 An incomplete suspicion in development.

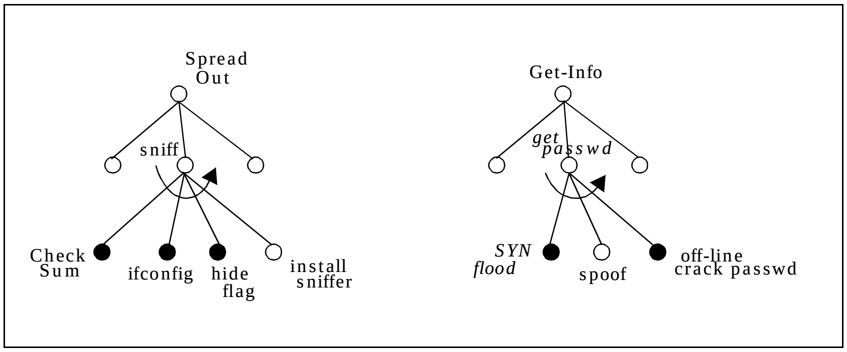

Figure-6 Flooding/spoofing/sniffing and possible directions of intrusion development.

Look Ahead Adaptive Auditing and High Level Recognition

IDS agents communicate in “agendas” that carry suspected intrusion intentions based on the threads given by the global IDS. This allows local agents to predict possible attack directions and fine tune its own system specific auditing utilities to pro-actively look for all relevant data. Later on, when meaningful data is observed, local IDS agent reports back sub-goal confirmation as well as the supporting data. Equipped with the knowledge of this new development, the global IDS agent re-evaluates the attack strategy and issue further commands. Helpful local IDS agents again look for data relevant to the new development.

With an intention representation, possible thread development provides information for look ahead adaptive auditing. In figure-6, it is possible to tell where the possible developments are. When an order is issued at this level, local IDS can start to adjust system's auditing behavior and look for events pertinent to the development.

Since a local IDS reports back to the centralized agent with sub-goal confirmation, the "field marshal" recognizes the attack at the strategic level instead of trying to figure out all the details in the large amount of local data. Therefore, for the global IDS agent, data noise and volume become less problematic with this strategic approach.

This is a delegation of responsibility that achieves a higher level of efficiency but requires an equally higher level of communication and trust. IDS and large-scale intrusion assessment/detection can never replace what human experts can do. We take the approach that IDS should be first line of defense therefore attack intention recognition provides a good definition for alert timing as well as potential coverage of attacks that have never happened before.

Integration of Paradigms, Tools and Technologies

One of the major considerations for taking this approach is to address the integration issue amongst different paradigms, tools and technologies.

Misuse Detection and *Anomaly Detection* are two common paradigms. Each uses different data and performs different processing. Different computing technologies are suited for different types of data and problems. Neural Networks are good at dealing with a large amount of high noise-to-signal ratio data. Therefore, it is perhaps a better approach than say, expert systems, for a particular anomaly detection problem. This leads to all kinds of different systems and tools. On top of that, attack patterns evolve as the new systems emerge. It is resource intensive for any one system trying to keep up with all these new attacks. From the practical perspective, ability to make use of and integrate with a wide variety of IDS efforts makes a lot of sense in the real world application.

Under the theme of intention analysis, IDS implemented using different paradigms and different technologies can all work together and contribute to the confirmation or exoneration of suspected intentions. Differences of data, discrepancy in operating environments and the potential of problem space explosion are reduced. The critical issue lies in defining the high-level communication protocol to help different flavors of IDS to contribute to the verification process. For example, to support a general definition of a "suspicious user", misuse-detection and anomaly-detection IDS may use totally different data, process it differently but contribute to the same result.

IDS Agent Communication

We do not view this as a totally autonomous environment where local IDS agents only maintain and serve their own agendas [Bradshaw]. Instead, we have a semi-autonomous environment where command-and-control provides the baseline for IDS agents to work with each other in pursuit of attack development in a coherent manner. Again, this is just like the battleground management situation.

From the behavior perspective, the global agent performs intention analysis, predicts trends and makes strategic decisions. Local agents execute the orders and look for data to confirm/exonerate sub-goals deemed likely by the global agent. Local agents are given total autonomy in determining the mapping from the local events to the sub-goals formation.

In addition, local agents are also responsible for monitoring the local environment and announce suspicions whenever detected - the traditional role. This is appropriate because they have the visibility of the local context needed to interpret the data. Local agents are "encouraged" to adjust own auditing behavior based on the global agent's predicted threads and sub-goals.

Within this behavior, it is apparent that the communication of "intrusion threads" between IDS agents is one of the major factors that need to be supported. These "threads" represent global IDS agent's attack intention analysis results and its determination in pursue. Figure-7 and figure-8 describe part of a IDS agent communication protocol in support of this high level collaboration.

"Offer" is used for announcing the availability for service and data. A new local IDS can offer its availability and the type of the services. This will obviously start off an authentication process. A local IDS agent can also "offer" to the global agent any suspicious event(s), e.g. multiple tank-sighting, not being tracked by the global IDS agent.

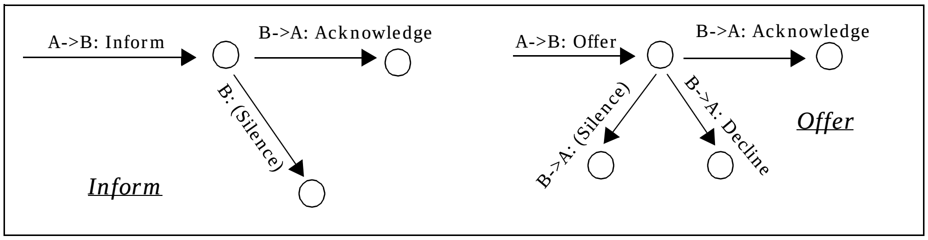

Figure-7 IDS agent communication protocol - Offer and Inform.

"Inform" can be used by a new global IDS to announce its intent to take over the control of the local IDS agents. It is also be used to announce status changes to other IDS agents.

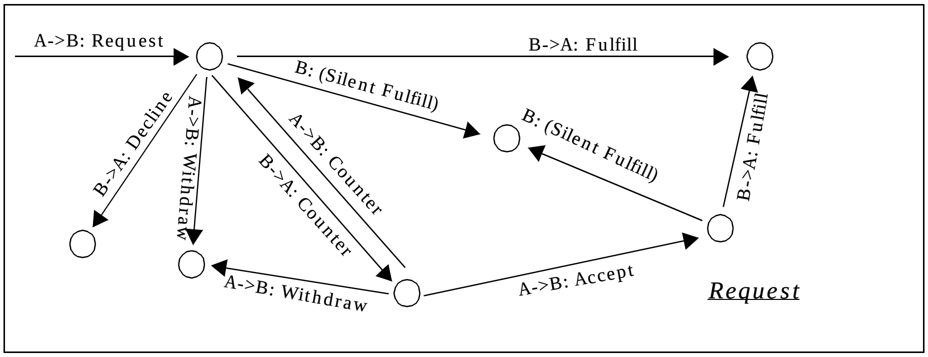

Figure-8 IDS agent communication protocol - Request.

"Request" is almost equivalent to "order". Global IDS agent can request suspicious intrusion threads to be observed/substantiated by local IDS agents. A global IDS agent may also request specific event data from the local agents. Request (of identity) is also used to start up an authentication process after an "offer" of service or an "inform" of taking over the control.

Issues

Both in the real-life and ID situations, intruders may take evasive action to prevent the real attack direction from being discovered. Various decoys are also used to generate large volume of misleading data. It is apparently more difficult to perform intention analysis when one of the opposing strategies is to mislead it. However, there are mathematical modeling and other techniques that can be used to do a best-judgement exercise. It is nevertheless an improvement from today's ID scenario that IDS labors at the noisy event level without any aid from strategic analysis.

Using augmented goal-tree representation for intrusion intention is similar to using the fault-tree model for the diagnosis process. The concept of intrusion intention is somewhat equivalent to failure model in a fault-tree - the logics behind failure. One more augmentation to the goal-tree representation is the build-in probability factor. With that, thread development can be statistically analyzed to help the over-all decision making process.

Related Works

The model-based approach was used by DEC in building the PLOYCENTER (ESSENSE) host-based IDS. POLYCENTER uses s-expressions to represent high-level attack patterns - not sequence of events. It then compares current events with these patterns to adjust auditing and performs limited look-ahead adaptive auditing [HV][VHJ][TC]. This approach greatly reduces the search space in VMS and Ultrix misuse detection. However, the high-level patterns were not modeled after intentions and it does not integrate with *Anomaly Detection* IDS. The model-based approach has also not been generalized to address large-scale network IDS problem.

Kumar and Spafford used Colored Petri Automaton (CPA) condition analysis as a generic pattern matching model for misuse intrusion detection [KS]. The signature patterns represented by the automaton focus on system/network event level. Bishop, Wee and Frank examined various models for representing security policies [BWF]. By working with a system model, the criteria and goals for auditing can be determined.

What the "Intention Analysis" based approach brings is the multi-leveled event abstraction that facilitates high-level intention recognition, a theme for distributed collaboration and look ahead adaptive auditing.

Acknowledgment

Without the supporting environment provided by The Boeing Company, Kjell Carlsen and Ken Neves, the making of this effort would be impossible. Special appreciation to Roberto Altschul, Jeff Bradshaw, Larry Bugbee, Jai Choi, Mark Greaves, Rob Jasper, Anne Kao, Dave Levine, Steve Poteet and Tom Tosch for their technical insights and stimulating idea exchanges that greatly contributed to the formation of the concept behind this architectural paradigm.

References

[BWF] Matt Bishop, Christopher Wee, Jeremy Frank. Goal-Oriented Auditing and Logging

[Bradshaw] Jeffrey M. Bradshaw. KaoS: Toward An Industrial-Strength Open Agent Architecture

[HV] Gary W. Hoglund, Eduardo M. Valcarce. The “ESSENSE” of Intrusion Detection: A Knowledge-Based Approach to Security Monitoring and Control.

[KS] Sandeep Kumar, Eugene H. Spafford. A Pattern Matching Model For Misuse Intrusion Detection, An Application of Pattern Matching in Intrusion Detection.

[Sundaram] Aurobindo Sundaram. An Introduction to Intrusion Detection.

[TC] H. S. Teng, J. Chen. Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns.

[VHJ] Eduardo M. Valcarce, Gary W. Hoglund, L. Jansen, L. Baillie. ESSENSE: An Experiment in Knowledge-Based Security Monitoring and Control.