

An Architecture for Intrusion Detection Modeled After the Human Immune System

John M. HALL

Computer Science, University of Idaho
Moscow, Idaho 83843, USA

and

Dr. Deborah A. FRINCKE

Computer Science, University of Idaho
Moscow, Idaho 83843, USA

ABSTRACT

We propose a novel architecture for an immunological network intrusion detection system, Immune System Network Intrusion Detection System (ISNIDS), suitable for inclusion in a broader-based multi-enterprise misuse management system. This paper will discuss the architecture, prototype, testing, and lessons learned from ISNIDS, as well as outlining the strategy for integration with a distributed/collaborative misuse management system.

This paper compares the prototype with a similar rule based system in both live and isolated conditions. The live testing was geared toward evaluating the number of false alarms generated under normal conditions. The isolated testing was geared toward evaluating the number of attacks missed under attack conditions. Each detection scheme detected six of the eight implemented attacks. ISNIDS missed one of two masquerading attacks and one password guessing attack. The rule-based system missed both masquerading attacks. As expected, this indicates that the two types of systems could effectively augment each other. The immune-based IDS offers considerable promise as traditional detection methods also have difficulty recognizing masquerading type attacks.

Keywords: Intrusion Detection System, Immunology, Anomaly Detection, Hummer and ISNIDS.

1. INTRODUCTION

Immunology based IDS apply human immune system concepts to network security [6]. The human immune system is very good at repelling a wide range of attacks in a dynamic hostile environment. Immunology is based on determining whether events belong to self or to non-self. This makes immunology approaches anomaly based.

This means that immunological approaches try to look for events that demonstrate the system is operating unusually.

ISNIDS implements immune detection in a hierarchical manner such as used in Hummer [8]. This hierarchy is formed by assigning each group of IDS a manager. In this system, the manager is responsible for creating detectors for the subordinates. The immune detectors used in ISNIDS are formed primarily through negative selection. Negative selection is the process in which random detectors are compared to normal events. In the typical negative selection process, the detectors which match a normal event are destroyed. The detectors that remain after the negative selection process will only match events that are not normal. Unfortunately, this process makes creating detectors a computationally infusible operation. ISNIDS improves this process by dynamically reducing the sensitivity of detectors to guarantee that the detectors do not match the normal traffic. We call this process *dynamic approximate binding*. ISNIDS separates the collection of events, the grouping of events, the analysis of events, and the response to events.

In this preliminary stage, we wished to determine whether this is a feasible prototype with four attacks in mind. The goal was to detect and respond to these attacks automatically, effectively, and quickly using an immunological approach. These general attacks were doorknob rattling, password guessing, masquerading, and illegal data access attempts. Our results indicated that this approach shows considerable promise. The prototype detected 6 of 8 implemented attacks.

Because of the preliminary nature of this prototype, several issues were ignored to simplify development. These limitations would seriously hinder the use of this intrusion detection system in a production environment.

The prototype was not designed for efficiency, robustness, security, or configuration.

2. ARCHITECTURE

ISNIDS was designed as two systems, a primary IDS and a secondary IDS. These components communicate across the network. The primary IDS is centralized and is responsible for creating detectors. This is done using negative selection. In order to adapt to new attacks, the primary IDS also evolves its gene library using clonal selection, a process through which components of successful detectors are recombined using the evolutionary process to make new detectors. The secondary IDS is distributed and is responsible for data gathering, data reduction, detection, and response. It also forwards successful detections to the primary IDS. This architecture is similar to the artificial immune model proposed by Kim and Bentley [14]. Their model offered significant promise. This system is intended to show the results and practical concerns of implementing such a system in a large environment.

In terms of the human immune system, the centralized primary IDS represents the thymus and to some degree the bone marrow. The decentralized secondary IDS represents the mobile portions of the immune system. In particular this IDS represents the flow of immune system detectors throughout the body.

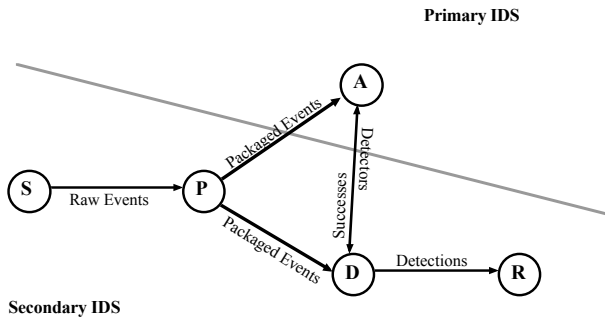


Figure 1. Overview of ISNIDS

The architecture of ISNIDS is shown in figure 1. The secondary IDS consists of four components, the sensors, the packager, the detector, and the response. The primary IDS consists of only an analysis component. The sensors collect audit information and convert it to a common event format. The packager performs data reduction by grouping the events into sessions. The analysis component uses these sessions to create detectors. The detector component matches current sessions to its detectors. Finally, the response component automatically responds to attacks. Ideally, once the secondary IDS had a set of detectors, it could continue to function even if the primary IDS failed.

3. IMPLEMENTATION

Detailed information of the overall ISNIDS system can be found in [9]. However, the implementation of several components warrant additional discussion. These are the packager component, the detection component, and the response component.

The packager component was originally missing from the architecture. Early experiments indicated that raw network events were insufficient for the IDS to effectively distinguish between attacks and normal behavior. The packaging method used should be investigated further. The current implementation groups events based on the associated user, the associated network address, and the event time.

Another interesting implementation detail is the detection method used by detectors to match event sessions. One of the largest problems facing the implementation of an immunological based IDS is negative selection [15]. In the human body, this is a massively distributed process. Kim and Bentley found that generating a detector by applying only negative selection on network data matching four or more consecutive attributes was a computationally infeasible operation [15]. They failed to generate a single detector with four genes that could survive negative selection after running their experiment for 24 hours. ISNIDS solves this by using *dynamic approximate binding*. In this process, immature detectors are created to match all events. Every time the detector matches a valid session, the binding constraints of that detector are tightened until it no longer matches. Using this method, a detector will only be discarded if it exactly matches all the values in a session.

The prototype implementation of the response component is also interesting. The response dispatcher will choose the response with the least impact. The impact of responses will change over time based on prior responses. The firewall response for example has several levels based on the current state of the firewall. The responses will also remember previous actions. For example, if the trace response has recently traced an IP address, then it will not consider itself a valid response if the same address is detected again. The system shutdown response is configured to have the most impact.

4. ASSESSMENT

The prototype was evaluated to determine the viability of this architecture. In order to facilitate this process, a rule-based detection scheme of similar complexity was also built. We tested the prototype and a similar rule-based system under both normal and attack conditions. The normal condition data was generated over a two and a half week

period. Over this period all audit data was saved to files. These files were later sent to both systems. We generated two attacks based on each threat considered. The audit data collected over the duration of these attacks was also saved and later fed into the detection components. Our attacks were, a probe, password guessing, masquerading, and attempts to locate private data. The audit data from each attack was sent to each system independently.

The immune detector detected 400 attacks out of the 2618 identified sessions in this data [9]. At over 20 detections per day, this seems unwieldy. However, almost all of these are doorknob rattling and worm attacks which all fall into our initial definition of attacks. There was one detection of particular interest. The events in this session were packaged incorrectly, merging legitimate activity with a doorknob rattling attack. The detector correctly identified the doorknob rattling portion as an attack. However, this may be a danger if an automated response may affect the legitimate activity. To summarize, we believe that ISNIDS generated only one false positive when run on 17 days of log events.

The rule-based detector detected only 2 attacks in this data. One of these appears to be a legitimate attack, the other appears to be the same incorrectly packaged event that ISNIDS detected. To summarize, the rule-based detector also made only one false positive when run on 17 days of data. The rule-based system did not detect the doorknob-rattling attacks simply because we did not write a rule with these attacks in mind. This was also acceptable behavior.

Both the immune-based and rule-based detectors performed exceptionally well in this test when evaluated with respect to number of false positives. They both incorrectly identified the same session as an attack.

The next step of the test plan was to feed attack data to both detection systems. For each of the four threats identified, a local and a remote attack was generated. Almost all of the detections occurred in less than a second after the packaging component released the session. The packaging component was designed to take at least 20 seconds. In most cases, this detection time seems reasonable. Both detection schemes detected six attacks and missed two attacks. Based on these tests, this gives each detector a 25% false negative rate. This indicates that each detection scheme offers fairly good protection. It is interesting that the attacks missed by the rule-based system were the masquerading attacks.

5. FUTURE WORK

The response mechanism currently implemented in ISNIDS is not as good an immune response as it could be.

In an immune system context, the current mechanism implements the two standard types of response: weakening the intruder and strengthening self. It also implements global and local responses. However, the system does not classify them by these attributes. Such a classification would aid the system level immune system view. Another weakness of the current implementation is that choosing an appropriate attack does not take into account whether the attack is still in progress. The current mechanism also does not allow cooperative responses between IDS.

The negative selection algorithm used was very effective at creating detectors. With early versions of the gene library, about 7 detectors were destroyed for every 1000 that matured. Later versions of the gene library lowered that to less than 1 for every 4000. That being said, using binary masks with counting values did not seem to work as well as it does for other values. The problem is that most of the sessions contain small values for these. As a result, the mask size tended to be quickly reduced. Perhaps a range type gene would handle these values better.

Building sessions worked well. However, there may be ways to do better. ISNIDS condensed 15,071 log entries to 3394 events to 2618 sessions in the original training data. Increasing the packaging delay did not drastically reduce the number of sessions. Allowing longer delays between events in the same sessions had a much larger effect on the number of sessions. This implies that there are a large number of small sessions generated.

Gene selection is a difficult problem. The current prototype makes it very difficult to add or remove genes. Simplifying this process is necessary before we can easily compare the effectiveness of different genes. A common representation for genes would be desirable. This would also seem to make the system more like the human immune system in that the human immune system represents every detector as a string of the same underlying protein structure.

Training is important. Originally, only about 2000 sessions were used to create the detectors. This was too few. This original detector set matched almost two-thirds of other sessions. Some less than optimal solutions including requiring multiple detector matches were tried. This was not viewed as a good long term solution because it uniformly increased the probability that a malicious event would be missed. The current training set consists of around 4000 sessions. Along with other changes made, the detection rate is much more reasonable. This does have the side effect of training the system to expect more anomalous behavior as normal.

Diversity in the gene library is essential. The original gene

library was built from the training data. This resulted in detectors that matched almost no anomalies. Based on this library, only 2 of the false negative tests above passed. The initial generation of the library was not optimal. We suspect that more randomness is needed to effectively detect novel attacks.

Integrating this tool into the Hummer architecture [8] consists of two steps. First, ISNIDS must have a tool written to integrate its output to the Hummer system. This step may involve porting portions of the system to Java as the current prototype was written in C++. Second, ISNIDS must be integrated with the Hummer communication structure. This will allow the hierarchy needed by ISNIDS to update dynamically as Hummer manager/subordinate relationships change. This step may require extending the existing Hummer architecture to allow tool-initiated communications. To allow efficient early prototyping, we simplified certain aspects of ISNIDS. In particular, we have not included key aspects of production IDS such as bandwidth scalability, speed of processing, intrusion/fault tolerance of the system, and configurability. Some of these may be the subject of future research while others are more appropriate topics of study when ISNIDS is situated in the Hummer architecture.

6. CONCLUSION

We built a prototype of an intrusion detection system that used human immune system concepts to detect attacks. This prototype shows considerable promise, especially at detecting unexpected attacks. The immune system model is often better at noticing patterns than a rule-based system. However, the immune-based system may miss some obvious attacks and raise alerts when exposed to rare but permissible activities. Our tests showed that the immune model performed comparably well to a rule-based prototype system of similar complexity. With this prototype we have shown that reasonably efficient intrusion detection may be performed using human immune system concepts. However, we would further recommend combining both detection methods to maximize the effectiveness of an IDS.

7. REFERENCES

- [1] Bradley, Daryl, Cesar Ortega, Andy Tyrrell, **Embryonics + Immunotronics: A Bio-Inspired Approach to Fault Tolerance**, 2000, <http://citeseer.nj.nec.com/bradley00embryonics.html>.
- [2] Dasgupta, Dipankar, Nii Attoh-Okine, **Immunity-Based Systems: A Survey**, 1997, <http://citeseer.nj.nec.com/dasgupta97immunitybased.html>.
- [3] Forrest, Stephanie, Alan S. Perelson, **Self-Nonsel Self Discrimination in a Computer**, 1994, <http://citeseer.nj.nec.com/forrest94selfnonsel.html>.
- [4] Forrest, Stephanie, Brenda Javornik, Robert E. Smith, Alan S. Perelson, **Using Genetic Algorithms to Explore Pattern Recognition in the Immune System**, 1993, <http://citeseer.nj.nec.com/forrest93using.html>.
- [5] Forrest, Stephanie, Steven A. Hofmeyr, **Engineering an Immune System**, 2001, http://www.cs.unm.edu/~forrest/ism_papers.htm.
- [6] Forrest, Stephanie, Steven A. Hofmeyr, Anil Somayaji, **Computer Immunology**, 1996, <http://citeseer.nj.nec.com/forrest96computer.html>.
- [7] Forrest, Stephanie, Steven A. Hofmeyr, Anil Somayaji, Thomas A. Longstaff, **A Sense of Self for Unix Processes**, 1996, <http://citeseer.nj.nec.com/forrest96sense.html>.
- [8] Frincke, Deborah, Don Tobin, Jesse McConnell, Jamie Marconi, Dean Polla, "A Framework for Cooperative Intrusion Detection", **Proceedings of the 21st NIST-NCSC National Information Systems Security Conference**, 1998, <http://citeseer.nj.nec.com/frincke98framework.html>.
- [9] Hall, John, ISNIDS, "A Network Intrusion Detection System Inspired by the Human Immune System", **CSDS Technical Report, CSDS-DF-TR-03-12**, 2002, <http://www.uidaho.edu/~hall0393/isnids.pdf>.
- [10] Hofmeyr, Steven A., **An Immunological Model of Distributed Detection and Its Application to Computer Security**, 1999, <http://citeseer.nj.nec.com/hofmeyr99immunological.html>.
- [11] Hofmeyr, Steven A., Stephanie Forrest, **Architecture for an Artificial Immune System**, 2000, <http://citeseer.nj.nec.com/374069.html>.
- [12] Kephart, Jeffrey O., **A Biologically Inspired Immune System for Computers**, 1994, <http://citeseer.nj.nec.com/kephart94biologically.html>.
- [13] Kephart, Jeffrey O., Gregory B. Sorkin, Morton Swimmer, and Steve R. White, **Blueprint for a Computer Immune System**, Virus Bulletin International Conference, San Francisco, CA, 1997.
- [14] Kim, Jungwon, Peter Bentley, **An Artificial Immune Model for Network Intrusion Detection**, 1999, <http://citeseer.nj.nec.com/262970.html>.
- [15] Kim, Jungwon, Peter Bentley, **An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection**, 2001, <http://citeseer.nj.nec.com/447684.html>.
- [16] Kim, Jungwon, Peter Bentley, **The Human Immune System and Network Intrusion Detection**, 1999, <http://citeseer.nj.nec.com/kim99human.html>.
- [17] Kim, Jungwon, Peter Bentley, **Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection**, 1999, <http://citeseer.nj.nec.com/kim99negative.html>.
- [18] Kim, Jungwon, Peter Bentley, **Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection**, 2002, <http://citeseer.nj.nec.com/531326.html>.