Implementation and Comparison of a Rules-Based Approach and a Statistical Approach Intrusion Detection Systems

Khaled M. Elleithy, Shawn Bell, Brenda Plaag, and Darren Stone

Abstract-- **This paper presents an analysis of a rules-based approach and a statistical anomaly approach to Intrusion Detection Systems (IDS). Two IDS systems are implemented. Analysis and comparisons of the systems are presented, as well as conclusions regarding the two approaches.**

Index Terms—**Communications systems security, Intrusion Detection Systems (IDS), Rule-based approaches, Statistical approaches, Ping of Death.**

I. Introduction

INCREASES in the interconnectivity of computers and computer networks as well as the rising level of sophistication and automation of intrusive attacks make IDS a tool of great importance to the network administrator. However, a poor choice in IDS systems can do more harm than good by providing a false sense of security. Research in IDS systems has flourished in the last decade and has attracted extensive efforts in the recent years [1-10].

IDS systems attempt to provide a safety net to other network security systems. An effective IDS implementation makes no assumptions about the effectiveness of other network security services. All activity is suspect and monitored to assess the perceived threat of the actions. Threatening behavior is responded to via logging, reporting or some action on the part of the IDS or related systems.

The value of an IDS system lies in its ability to accurately determine if an action is a result of an intrusion or normal behavior.

Intrusion detection systems fall into two broad categories. The first approach is Statistical Anomaly Detection. The underlying premise of this approach is that an intruder's activity will differ from that of a legitimate user. Legitimate user behavior is derived by analyzing past activity. Aberrations that are outside expected deviations are reported as the act of an intruder. The strength of statistical approaches is that attacks are defined as any non-normal activity, so it can theoretically guard against any type of attack.

The second approach is Rules-Based Detection. A rules-based detection system defines specific activity as being an intrusion. The system looks for known attacks or defined behaviors and reports when these are observed.

This paper presents an examination of both approaches to IDS. We have developed and implemented a rules-based and a statistical anomaly IDS system. Both IDS systems monitored a Web-based wire transfer application of similar architecture. Analysis of the two systems reveals their respective strengths and weaknesses.

II. Techniques and Implementation

We have developed two IDS implementations. One was a rules-based approach, the other a statistical anomaly approach. The rules-based system evaluates actions on the rule-set alone and the statistical approach evaluated actions only against statistical models of historical data.

As mentioned, both systems monitored a Web-based wire transfer system. Users interact with the database via a Web front-end. All activity is monitored surreptitiously by a second IDS database. The respective IDS algorithms are implemented within the interaction between the two databases, easily hidden to the front-end user.

The system administrator can interact with the IDS database via a separate Web site or an isolated section of the production Web site. There the administrator can view logging and report data, or in the case of the rules-based implementation, they can define and modify the rule set.

Both systems use Microsoft IIS Server to host the Web front-end applications (user and administrator components). ASP code using VBScript was used to program the applications and both rely on Microsoft databases as a back-end. The rules-based system uses MS Access, while the statistical anomaly approach uses MS SQL Server. Figure 1 shows the system architecture and Figure 2 shows the used system tools in development.

Khaled M. Elleithy is with the Computer Science and Engineering Department, University of Bridgeport, Bridgeport, CT 06601, elleithy@bridgeport.edu

Shawn Bell, Brenda Plaag, and Darren Stone are with Computer Science and Information Technology, Sacred Heart University, Fairfield, CT 06825

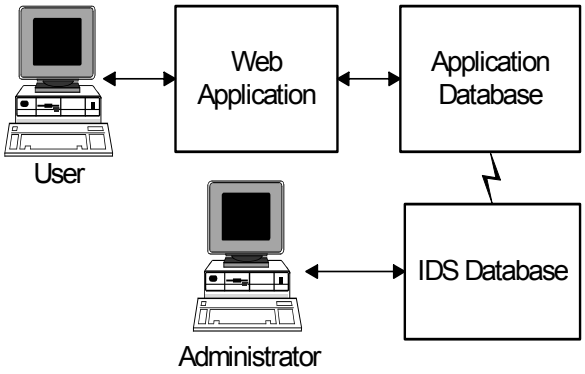

Fig. 1. System Architecture

Technology	Rules-based	Statistical Anomaly
Web Server	MS IIS Server	MS IIS Server
Application	ASP w/ VBScript	ASP w/ VBScript
Database	MS Access	MS SQL Server

Fig. 2. System Tools.

III. Details of Rules-based Implementation

The front-end comprises a variety of different screens, each performing functions that are checked by the IDS system. The login screen provides a security service by authenticating users. Additionally, it also determines the user's level of access based on the type of user (administrator versus standard user).

Once an administrator logs in successfully the application points to the administration screen where intrusion attempts are reported. These reports indicate log-in failures as well as any other unauthorized activity that falls within the scope of the rule set. This reporting module allows the administrator quick access to data regarding possible intrusions.

Also, within this section of the site, the administrator can change elements of the rule set thus allowing for a degree of customization in the application. Rules can be changed periodically to reflect changing activity within the system, resulting in few false positive reports of intrusion.

Standard users successfully logging in will be directed to the Banking screen. This screen allows the user to transfer money from one account to another. Users can also add new accounts, and log out of the application.

The Banking screen consists of two drop-down menus determining the account source and destination of the transfer. A text box is provided for entering monetary values.

The system's back-end consists of two separate databases. The first database is called Rules.mdb and is the production database, designed to store the information captured from the front-end.

The second database is called IDS.mdb. IDS.mdb implements the IDS system. This database collects information that has been stored in Rules.mdb and then checks for intrusions based on the rules stored in IDS.mdb. The IDS.mdb will encrypt the data that comes in if time permits, however the assumption is made that security precautions have been made to secure the web server.

The actual rules are stored within tables in IDS.mdb. This facilitates rule changes, as the administrator merely needs to change the values stored in these tables. The following are examples of rules implemented by this system:

Rules 1 through 5 evaluate log in information.

Rule1: Checks for a valid user.
Rule2: Confirms the password, 3 false logins is defined as an intrusion
Rule3: Checks for user being allowed to login at that time
Rule4: Amount of logins in a day
Rule5: Checks to see if user was already logged in and tries to login simultaneously

Rules 6 onwards evaluate the actions of the user after log in.

Rule6: Checks the amount the user is transferring
Rule7: Checks the accounts that the user can transfer to or from
Rule8: Checks to see if user has permission to set up new accounts
Rule9: Number of transactions per user in a day

IV. Details of Statistical Anomaly Implementation

The application monitored by this system is very similar to that of the rules-based system. Users log in and are authenticated from the log in screen and can then process transactions. Log in attempts and all transactions are stored in a production database.

This production database is tied to an IDS database that evaluates the transactions and assigns them an alert level depending on the evaluated level of suspicion. This evaluation occurs as the transaction takes place and could be programmed to send alerts to the administrator in real-time.

The administrator logs in to a separate site that access the IDS database. There the administrator can view reports on the activity in the production database. Reports can be filtered by alert level to ignore innocuous transactions.

Data in the IDS database is encrypted to thwart attackers who gain access to the IDS database. While the encryption

employed in this system is very modest (values are converted to ASCII values, shifted two positions and written back in reverse), it is implemented as a separate 'black-box' function that can easily be swapped out for a more secure encryption scheme.

The anomaly detection engine is implemented as SQL Server stored procedures. These procedures can be encrypted by SQL Server to provide additional security. However, once encrypted they cannot be decrypted even by the developer. To facilitate development the stored procedures were not encrypted in this implementation.

The stored procedure that assesses the alert level does so by performing analysis on the historical data in the system. To do this the data is decrypted and stored in a staging table so that analysis can be performed. Once an alert level has been determined it is encrypted and stored in the transaction table with the corresponding transaction. The decrypted data in the staging table is immediately purged once the alert level is determined.

Alert levels are determined by measuring current activity against previous activity. This is achieved by constructing profiles of normal activity. Actions are decomposed into their component parts and evaluated against the corresponding property of the 'normal' profile.

Profiles model entities in the database such as the user and the bank accounts. The profile consists of a property, the properties average value and a standard deviation for the range of values associated with that property. An example would be the following:

	Bank Account
Property	Transaction Amount
Value	$100
Standard Deviation	22

A new transaction amount is tested to determine if it falls with the standard deviation for the average value, which would be expected. Values outside the standard deviation increment the alert value. The further the new values vary from the standard deviation, the higher the alert value is incremented.

Each property associated with the transaction is measured in a like fashion and a final alert value is determined. This value is then stored in the IDS database with its transaction.

As mentioned previously reporting on this data is provided via a Web page that can be filtered by alert level.

V. Analysis

The strength of statistical approach is the fact that intrusions are defined as any non-normal activity. Rules do not have to be defined to cover all possible intrusions. Any activity that does not relate closely to previous activity is regarded as an intrusion. However, this implementation is far more processor intensive. With each transaction the system must create a profile of all the entities involved in the transaction and measure the profile properties against the corresponding component of the new transaction.

The current implementation is small enough to run without problem. However even given the scope of this approach, each transaction has at least four components (users, account source, account destination and amount). Therefore, each incremental increase of activity in the production database could result in a fourfold increase in activity in the IDS database.

Creating models and storing them in the database could address this issue. The system could then simply retrieve the constructed model and perform its evaluations. However the model would still need to be refreshed periodically, and this refresh would be processor intensive.

This problem is a much smaller factor in the rules-based approach. Each new transaction is evaluated by a simple Boolean evaluation of a pre-defined rule. There is no additional computational overhead in this system.

The rules-based approach suffers from the limitations of its own finite rule set. Transactions that fall outside of the rule set are not evaluated and thus undetected.

Both implementations are very modest in scope. As a result, the statistical implementation was not noticeably effected by the extra computational activity and the rules-based approach was not compromised by a lack of rules.

However, there was one significant aspect of the two systems that we were able to examine and measure. The statistical implementation reported far greater false-positive alerts than the rules-based implementation.

Two theoretical users were proposed to research this issue. One user, Operator A, performed very regular activity in the database and consistently logged in successfully with one attempt. The other user, Operator B, executed far more dynamic transactions within the database and frequently had multiple failed log-in attempts.

The statistical anomaly approach successfully modeled Operator A's activity after three transactions. The three transactions that incurred an alert value greater than zero were still low, the highest alert value being three. This transaction was rated highest due to the fact that it was the first time it occurred, which systematically generates a level three alert

value. This transaction acts as a seed value for future evaluation.

The statistical system could not effectively model Operator B's dynamic activity. Only one transaction rated zero, while most transactions had alert levels greater than five. Operators B's activity was too dynamic to lend itself to statistical modeling.

This issue cannot be resolved in a pure statistical approach. If the statistical engine is altered it becomes a rules-based system by the nature of the modification. One could develop multiple statistical engines and implement them on a user-by-user case, but once again rules are introduced into the system:

If User A: Use Engine A

The rules-based system could successfully determined that Operator A's were normal from the start, there was no 'seeding' time as in the statistical implementation as the rules are pre-defined. It could also successfully evaluate Operator B's activity because the rules could be altered as they applied to B, giving this user more flexibility and eliminating false-positive reports.

VI. Conclusion

IDS systems can be categorized by their method of intrusion detection. These methods can be placed into two broad categories: rules-based and statistical anomaly intrusion detection. The rules-based system implements a rule set to detect intrusive behavior, while the statistical anomaly approach uses mathematically-determined models to detect intrusion.

Both approaches have their strengths and weaknesses. The statistical anomaly system can detect intrusive behavior that has not been pre-defined, but it is processor intensive. The rules-based system is far less taxing on the processor, but activity outside of its rule set is undetected.

We found that the statistical approach has significant limitations in a very dynamic environment. Such an environment produces a rash of false-positive intrusion reports. These dynamic environments do not lend themselves to statistical modeling and must be approached carefully to avoid both false-positive reporting as well as false-negative reporting (rule sets that are too broad could fail to detect actual intrusions).

A more effective approach to intrusion detection would be one that combines elements of the two systems.

The ability of a statistical approach to detect unknown intrusions is far too valuable to be discarded in an IDS system. However, the ability of rules-based system to limit false-positive alarms is also critically important to a reliable IDS system. Developers need to determine which activities are best suited to rules-based monitoring, and which lend themselves to a statistical approach.

VII. References

[1] S. Northcutt, Network Intrusion Detection, An Analyst's Handbook. New York: New Riders, 2001.

[2] W. Stallings, Network Security Essentials, 2nd edition. Prentice Hall, 2003.

[3] J. Beale et al., Snort 2.0 Intrusion Detection. Rockland: Syngress Publishing, Inc, 2003.

[4] T. Garrison, Todd Garrison Practical Examination for GCIA. Global Information Assurance Certification, 2003.

[5] The Internet Engineering Task Force, Intrusion Detection Exchange Format (IDWG) Charter, IETF, 2003.

[6] T. Buchheim, G. Matthews, et al. "Implementing the Intrusion Detection Exchange Protocol." Annual Computer Security Applications Conference, 2002.

[7] M Reis, F. Paula, et. al. "A Hybrid IDS Architecture Based on the Immune System." WSEG2002: Workshop on Security of Computer Systems, 2002.

[8] J. Kim and P. Bentley. "The Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with a Negative Selection Operator." University College London, 2002.

[9] L. de Castro and F. Von Zuben. "Artificial Immune Systems – A Survey of Applications." State University of Campinas, SP, Brazil, 2000.

[10] D. Dasgupta and F Gonzalez "An Immunity-Based Technique to Characterize Intrusions in Computer Networks." IEEE Transactions on Evolutionary Computation, 2002.

VIII. Biographies

Khaled M. Elleithy (M'1988) received the B.Sc. degree in computer science and automatic control from Alexandria University in 1983, the M.Sc. Degree in computer networks from the same university in 1986, and the M.Sc. and Ph.D. degrees in computer science from The Center for Advanced Computer Studies at the University of Southwestern Louisiana in 1988 and 1990, respectively. From 1983 to 1986 he was with the Computer Science Department, Alexandria University, Egypt, as a lecturer.

From September 1990 to May 1995 he worked as an assistant professor at the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. From May 1995 to December 2000 he has worked as an Associate Professor in the same department. In January 2000 Professor Elleithy joined the Department of Computer Science and Engineering in University of Bridgeport as an associate professor. In May 2003 he was promoted to full professor.

Professor Elleithy published more than sixty research papers in international journals and conferences. He has research interests in the areas of network security, mobile / wireless communications, computer arithmetic and formal approaches for design and verification.